

# Unsupervised Representation Learning for Object-Centric and Neuronal Morphology Modeling

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät  
der Eberhard Karls Universität Tübingen  
zur Erlangung des Grades eines  
Doktors der Naturwissenschaften  
(Dr. rer. nat.)

vorgelegt von  
Marissa Anthea Weis  
aus Berlin

Tübingen  
2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation: 19.12.2023

Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Alexander S. Ecker
2. Berichterstatter:	Prof. Dr. Philipp Berens

# Summary

A key feature of intelligent systems is the ability to generalize beyond one's experiences. The process of deriving general rules from limited observations is the principle of induction. Generalizing based on a finite number of observations to future samples requires a priori assumptions. In machine learning, the set of assumptions made by the learning algorithm are called inductive biases. To design successful learning systems, it is essential to ensure that the inductive biases of the system align well with the structure of the data. Conversely, to understand why learning systems fail in a particular way it is integral to understand their inherent assumptions and biases.

In this dissertation, we study unsupervised representation learning in two different application domains. We look through the lens of evaluation to unmask inductive biases in object-centric models in computer vision as well as show how to successfully employ inductive biases to integrate domain knowledge for modeling neuronal morphologies.

First, we establish a benchmark for object-centric video representations to analyze the strengths and weaknesses of current models. Our results demonstrate that the examined object-centric models encode strong inductive biases such as a tendency to perform mostly color segmentation that work well for synthetic data but fail to generalize to real-world videos.

Second, we propose a self-supervised model that captures the essence of neuronal morphologies. We demonstrate that by encoding domain knowledge about neuronal morphologies in the form of the appropriate inductive biases into our model, it can learn useful representations from limited data and outperform both previous models and expert-defined features on downstream tasks such as cell type classification.

Third, we employ our model for neuronal morphologies to a large-scale dataset of neurons from the mouse visual cortex and prove its utility for analyzing biological data. We demonstrate that our learned representations capture the morphological diversity of cortical pyramidal cells and enable data analysis of neuronal morphologies on an unprecedented scale. We use the learned embeddings to describe the organization of neuronal morphologies in the mouse visual cortex, as well as discover a new cell type and analyze cortical area and layer differences.

Taken together, our findings indicate that identifying the implicit inductive biases in object-centric models is necessary for understanding their failure modes. Conversely, tailored inductive biases, that take the intricacies of the domain into account, enable the successful design of machine learning models for neuronal morphologies.



# Zusammenfassung

Ein wesentliches Merkmal intelligenter Systeme ist die Fähigkeit aus Erfahrungen Schlussfolgerungen zu ziehen und daraus Vorhersagen für die Zukunft zu treffen. Induktion beschreibt die Ableitung allgemeiner gültiger Gesetzmäßigkeiten aus einer begrenzten Anzahl von Beobachtungen. Um allgemeine Aussagen basierend auf einer endlichen Anzahl von Beobachtungen zu treffen, ist es erforderlich a priori Annahmen zu machen. In maschinellem Lernen werden solche Annahmen als induktive Verzerrung des Algorithmus bezeichnet. Um erfolgreiche lernende Systeme zu entwickeln, müssen die induktiven Verzerrungen des Systems mit der Struktur der Daten übereinstimmen. Umgekehrt kann die Analyse inhärenter Annahmen und Verzerrungen von lernenden Systemen dabei helfen zu verstehen, warum Algorithmen in bestimmten Situationen versagen.

In dieser Dissertation untersuchen wir zwei Anwendungsbereiche von unüberwachtem Repräsentationslernen. Im Bereich Computer Vision analysieren wir, wie induktive Verzerrung die Inferenz von objektzentrierten Modellen beeinflusst. Des Weiteren zeigen wir, wie man durch gezieltes Einsetzen induktiver Verzerrung Fachkenntnisse aus den Neurowissenschaften in Algorithmen integrieren und diese zur Modellierung neuronaler Morphologien einsetzen kann.

Zunächst erstellen wir eine Benchmark für objektzentrierte Video-Repräsentationen, um die Stärken und Schwächen aktueller Modelle zu analysieren. Unsere Ergebnisse zeigen, dass die untersuchten objektzentrierten Modelle starke induktive Verzerrungen aufweisen. Sie erkennen zum Beispiel Objekten bevorzugt anhand homogener Farben. Diese induktiven Verzerrungen lassen sich nicht auf reale Videos übertragen und führen deshalb zum Versagen von objektzentrierten Modellen bei solchen.

Im zweiten Teil der Dissertation entwickeln wir ein selbstüberwachtes Modell zur Modellierung neuronaler Morphologien. Wir zeigen, dass unser Modell durch die Integration von Fachwissen in Form von induktiven Verzerrungen nützliche Repräsentationen der neuronalen Morphologien aus begrenzten Daten lernen kann. Die gelernten Repräsentationen können anschließend zur Klassifizierung von Zelltypen verwendet werden und übertreffen dabei sowohl Repräsentationen früherer gelernter Modelle als auch Repräsentationen, die von Experten definiert wurden.

Im letzten Teil der Dissertation wenden wir unser Modell für neuronale Morphologien auf einen großen Datensatz von Neuronen aus dem visuellen Kortex einer Maus an und zeigen den Nutzen der Repräsentationen für die Analyse biologischer Daten. Wir demonstrieren, dass unsere erlernten Repräsentationen die morphologische Vielfalt der kortikalen Pyramidenzellen erfassen können und somit eine Analyse großer Datenmengen ermöglichen. Wir verwenden die erlernten Repräsentationen zur Beschreibung der Organisation der neuronalen Morphologien im visuellen Kortex, wobei wir Gebiets- und Schichtunterschiede des Kortexes analysieren und einen neuen Zelltyp beschreiben.

Zusammenfassend zeigen unsere Ergebnisse, dass die Identifizierung impliziter induktiver Verzerrungen in objektzentrierten Modellen notwendig ist, um ihre Fehlerquellen zu verstehen. Umgekehrt ermöglichen maßgeschneiderte induktive Verzerrungen die erfolgreiche Entwicklung von Modellen für maschinelles Lernen neuronaler Morphologien.



# Contents

<b>Summary</b>	3
<b>Zusammenfassung</b>	5
<b>1 Introduction</b>	9
1.1 Definitions and background . . . . .	10
1.2 Thesis structure . . . . .	14
1.3 Publications . . . . .	15
<b>2 Object-centric learning</b>	16
2.1 Definitions and background . . . . .	16
2.2 Benchmarking unsupervised object representations . . . . .	20
2.3 Advances in object-centric learning . . . . .	22
<b>3 Modeling of neuronal morphologies</b>	25
3.1 Definitions and background . . . . .	26
3.2 Representation learning for neuronal morphologies . . . . .	30
3.3 Discovery of excitatory morphological cell types . . . . .	32
3.4 Limitations and future directions . . . . .	35
<b>4 Discussion</b>	39
<b>5 Outlook</b>	42
<b>Bibliography</b>	45
<b>Appendix</b>	67
Benchmarking unsupervised object representations for video sequences . . . . .	69
Self-supervised graph representation learning for neuronal morphologies . . . . .	133
Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex . . . . .	163
<b>Acknowledgments</b>	185





# 1 Introduction

In the last decade, machine learning has made enormous progress and is nowadays applied to countless applications ranging from natural language processing and computer vision to weather prediction and medical diagnosis (Espeholt et al., 2022; Esteva et al., 2017, 2019). Since the advent of deep learning and the increasing availability of large amounts of data and compute resources, machine learning algorithms exhibit unprecedented performances on diverse tasks across many data domains. However, while machine learning has taken over most of artificial intelligence research, there has been an ongoing debate about how much structure should be built into machine learning systems a priori (LeCun and Manning, 2018; Marcus, 2018; Sutton, 2019; Welling, 2019).<sup>1</sup> Building strong assumptions into models can facilitate learning if the assumptions match the underlying problem well, but restricts the solutions that can be learned. In recent years, the trend in machine learning has shifted to designing general methods that leverage the increasingly large computational resources to learn from experience with few built-in assumptions. However, machine learning models often struggle with systematic generalization and transfer of knowledge under domain shifts.

In contrast, humans and other animals are incredible learners. They are able to continually learn throughout their lifetime, use prior knowledge and generalize it to different contexts as well as learn from limited number of examples (Tenenbaum et al., 2011). To enable fast learning from few examples, prior assumptions are necessary. The Garcia effect nicely illustrates the interplay of inductive biases and learning in biological systems: If animals experience nausea after consuming food, they learn to avoid the respective taste in the future (Garcia et al., 1955). In contrast, experiments have shown that rodents are unable to associate sound or light stimuli with nauseating food intake (Garcia and Koelling, 1966). Hence, while the specific taste which identifies nauseating food is learnt through experience, the animal has an innate bias about which sensory cues can be associated with nausea, namely taste but not audiovisual stimuli. This simplifies the learning problem as it limits the range of possible solutions. To learn such an association only one or a few experiences are necessary. Afterwards, the animal will generalize it to future instances of similar tastes.

The principle of deriving a general rule from a finite set of observations is called induction. It is based on the assumption that the future to some degree resembles the past (Hume, 1739). Conclusions drawn via induction are not guaranteed to be true. Induction rather makes a probabilistic statement about what is likely going to happen (Goodfellow et al., 2016).<sup>2</sup>

Machine learning can be thought of as the automation of inductive reasoning (von Luxburg, 2020). Given a finite set of observations, called the training set, a machine learning algorithm is optimized to make accurate predictions about unseen data points. Generalization to unseen examples is impossible without imposing prior assumptions on the function to learn (Mitchell, 1980). Given a finite amount of data, there are multiple (possibly infinitely many) functions that equally well explain the observed data. Each of them can lead to different predictions for unobserved data. Hence, without imposing any bias, i.e. when treating all hypotheses as equally likely, predictions for unseen data points would be an arbitrary guess. This is independent of the number of available training points. Thus, to choose

---

<sup>1</sup>Similar debates with respect to the origin of knowledge in biological systems arose in cognitive science, where nativists argued for innate structures that underlie cognitive abilities, while empiricists advocated for a *tabula rasa* view, arguing that knowledge originates from experience (Spelke, 1998). Analogously, the “nature versus nurture” debate in biology evolved around the question how much of our behavior is determined by our genetic makeup versus shaped by the environment. Nowadays, evidence suggest that both factors contribute substantially and interact tightly (Robinson, 2004; Traynor and Singleton, 2010).

<sup>2</sup>Induction is usually contrasted with deduction. Deductive reasoning is the process of inferring a logically certain conclusion for a particular data point from one or more general premises.

between these possible hypotheses, additional assumptions need to be made. These assumptions or restrictions on the hypothesis space are called *inductive biases* in machine learning (Hüllermeier et al., 2013).

Wolpert and Macready (1997) formalized the notion that there is no general learner in the “No-free-lunch Theorem”. It states that instilling a model with inductive biases that lead to a better performance for one task is guaranteed to decrease its performance on other tasks. Therefore, no machine learning algorithm is universally better than any other considering all learning problems. However, that does not mean that for a specific task all machine learning algorithms perform equally well. Given a specific task, finding the appropriate inductive biases that align well with the underlying problem structure, will lead to favoring a function or hypothesis in the learning process that reflects prior domain knowledge about the target function and hence will lead to a better performance (Goodfellow et al., 2016). When developing machine learning algorithms, it is therefore integral to ensure that the inductive biases that are explicitly and implicitly encoded in the modeling decisions align well with the structure of the problem at hand in order to achieve good generalization. It is equally important to understand how inductive biases affect the trained models and their generalization capabilities. Analyzing the inductive biases of a model can shed light on its failure cases and identify ways to improve it.

In this dissertation, we take a closer look at two different applications of unsupervised representation learning, namely object-centric learning and modeling of neuronal morphologies. In the first application, we analyze the inductive biases of object-centric video models in order to understand their failure cases and identify the necessary steps to scale them to natural data. In the latter, we show how to incorporate domain knowledge in the form of inductive biases into our model in order to capture the patterns in neuronal morphologies. We then use the proposed model to analyze organization principles of cortical neurons in the mouse visual cortex.

In the following, we first give some necessary background knowledge for representation learning and inductive biases in machine learning, specifically deep learning. Subsequently, we introduce the two fields of application.

## 1.1 Definitions and background

In this section, we briefly introduce background topics and definitions used throughout this dissertation. Additional domain-specific background is provided in Sections 2.1 and 3.1.

### Representation learning

The way in which information is represented can massively simplify or complicate an information processing task (Goodfellow et al., 2016). Representation learning denotes the process of automatically extracting or *learning* discriminative feature representations from raw data (Bengio et al., 2013). The question of what constitutes a “good” representation is not generally answerable. Informally, a “good” representation is one that facilitates the extraction of information for subsequent tasks and is therefore highly dependent on the downstream task. In general, representations should be expressive, meaning that they need to retain all relevant information for the task at hand, while being as simple as possible. Simple here can refer to properties such as being low-dimensional, sparse or disentangled with respect to the underlying factors of variation (Bengio et al., 2013).

Representation learning can be seen as opposed to manually engineering features as it was — and in some domains still is — common. Feature engineering refers to the process of designing preprocessing pipelines and data transformations to obtain a representation of the data that extracts useful features for subsequent machine learning applications (Bengio et al., 2013). It often entails substantial manual labor. Computer vision research has long been focused on engineering the right input features for object recognition models, inventing for example SIFT (Lowe, 1999) and HOG features (Dalal and Triggs, 2005). With the advent of deep learning, these have been rapidly replaced with end-to-end learning of feature representations through deep neural networks. The learned representations often outperform manually-designed features as they are directly optimized for utility for the downstream task (Goodfellow et al., 2016).

Representation learning can be done both in a supervised as well as in an unsupervised setting. In *supervised* learning, given a set of labeled data points  $D = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\} \subset \mathcal{X} \times \mathcal{Y}$  which are sampled independently from a distribution  $P_{X,Y}$ , the model is trained to associate inputs  $\mathbf{x}$  with their respective labels  $\mathbf{y}$ . To train a network in a supervised fashion, a sufficiently large training dataset with ground truth annotations is required. This can result in a prohibitively large manual labelling effort, especially with increasing network sizes that need increasing numbers of training samples to avoid overfitting. Furthermore, the assigned labels impose a strong bias on the features that the network learns as it is only incentivized to retain information that is useful for predicting the labels.

In contrast, in the *unsupervised* training paradigm, the network is only allowed to observe a dataset  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{X}$  consisting of samples  $\mathbf{x}$  without labels attached. Unsupervised learning tries to recover useful structure or patterns from the data without using a supervisory signal in the form of explicit targets associated with the input data. This alleviates the potential heavy annotation costs as well as the bias towards only learning features that are necessary to predict the labels.

*Self-supervised learning* refers to a specific unsupervised learning paradigm in which labels for training are automatically constructed from unlabeled data. The resulting input-label pairs are subsequently used to train models in a supervised way (Jing and Tian, 2019).

## Inductive biases in machine learning

The term *inductive bias* refers to all underlying assumptions of a learning algorithm that are used to make predictions on unseen data points (Mitchell, 1980; Hüllermeier et al., 2013). Models learn a specific solution based on their inductive biases. These inductive biases can stem from the model’s architecture, the optimization procedure used and the training objective, among others.

**Inductive biases in deep learning architectures** The term *deep learning* originates from the concept of successively composing multiple layers of elementary building blocks into a more complex network, hence the characterization as *deep*. The stacking of multiple layers leads to the inductive bias of hierarchical processing: the data is explained by a complex function  $f(\mathbf{x})$  that is a composite of multiple simpler functions  $f_l$  (LeCun et al., 2015; Goyal and Bengio, 2022):  $f(\mathbf{x}) = (f_L \circ \dots \circ f_1)$ , where  $L \in \mathbb{N}$  is the depth of the network. The activations of each layer can be seen as a distributed representation of the input (Hinton et al., 1984) that becomes progressively more abstract with increasing depth.

The *Multi-layer perceptron* (MLP) (Rosenblatt, 1958) is one of the basic deep learning architectures, where  $f_l(\mathbf{x}) = \sigma(\mathbf{W}_l \mathbf{x} + \mathbf{b}_l)$ . Each layer  $f_l$  consists of a learnable weight matrix  $\mathbf{W}_l$  and bias term  $\mathbf{b}_l$ .

$\sigma$  denotes an element-wise non-linearity. In contrast to the architectures described later in this section, MLPs lack weight-sharing as well as a notion of locality. They impose minimal prior assumptions on the structure of the data as each unit can interact with all other units in the previous and subsequent layers and the strength of the interactions is learned.

The *convolutional neuronal network* (CNN) (LeCun et al., 1998) has been very influential in the advent of deep learning, especially in the computer vision domain. CNNs use convolutional layers as their primary building blocks. Convolutional layers apply a smaller kernel repeatedly over all spatial positions of the input, thereby significantly reducing the number of parameters of the network by using weight-sharing and explicitly encoding translational equivariance into the network (LeCun et al., 2015; Battaglia et al., 2018). Additionally, they encode locality as a relational inductive bias, meaning that only entities in close proximity to each other interact directly (LeCun et al., 2015; Battaglia et al., 2018). These proved to be strong inductive biases for processing natural images, as neighboring pixel values are often highly correlated and motifs in images are translation-invariant.

Images and videos are a special data modality in that the information is arranged on a regular grid of pixel values. Deep neural networks (DNNs) are very successful in processing this kind of data as the structure of the data can be directly built into the models, as exemplified by CNNs. However, data is not generally organized in grids. Therefore, geometric deep learning has emerged as a way to generalize DNNs to non-Euclidean data, such as graphs and manifolds (Bronstein et al., 2017). *Graph neural networks* (GNNs) have been developed to cater to data that comes in the form of graphs. GNNs are invariant to permutations over nodes and edges (Battaglia et al., 2018). They can express arbitrary relationships between entities in the graph depending on whether two nodes are connected by an edge and therefore allow for the exchange of information via message passing (Scarselli et al., 2009). This is in contrast to CNNs or MLPs, in which the interaction between inputs is fixed through the connections of units in the architecture and hence identical for all inputs.

Recently, *transformers* that use attention as their primary layers led to a paradigm shift first in natural language processing (Vaswani et al., 2017), which then transferred over to computer vision (Dosovitskiy et al., 2021; Caron et al., 2021) and graph learning (Zhang et al., 2020; Dwivedi and Bresson, 2021). Transformer attention is based on similarity-based information flow between parts of the input. These input parts, for instance words in language modeling or patches of pixels in images, are encoded into so-called tokens. All tokens can exchange information independent of their position in the input, such as the word position in a sentence or the spatial location of an image patch.<sup>3</sup> Thus, transformers exhibit a non-local inductive bias and can model long-range dependencies in the data. Furthermore, the attention mechanism is permutation equivariant over tokens (Goyal and Bengio, 2022). The switch to transformer architectures in computer vision led to more flexible information processing based on similarity of image patches, but meant giving up the well working inductive biases of convolutions for images. With enough training data, it has been shown that the transformer architectures with less specific thus weaker biases for images are still competitive in performance on different computer vision tasks (Dosovitskiy et al., 2021).

In this dissertation, we consider two different forms of architectural biases. As we discuss in more detail in Chapter 2, object-centric models make the notion that complex scenes are composed of simpler building blocks explicit by enforcing a compositional representation through their architecture: Instead of learning one distributed representation of an image or video, they learn a set of multiple representations in a shared format, each representing one object in the scene. Second, in Chapter 3, we draw on domain specific knowledge of neuronal morphologies to design a novel atten-

---

<sup>3</sup>However, by adding positional encodings to the token values, locality information can be taken into account.

tion layer, namely `ADJACENCY-CONDITIONED ATTENTION`, that combines transformer attention and GNN message-passing to optimally process the sparse, long-ranging branches of neuronal skeletons.

**Inductive biases through data augmentations** Recently, models trained using self-supervised or contrastive learning have tremendously improved in their performance (Chen et al., 2020; Caron et al., 2021). One important ingredient to make this learning paradigm work is the definition of appropriate data augmentation strategies. Data augmentations are applied to the input data and encode the desired invariances of the learned representations (Cabannes et al., 2023). In computer vision tasks such as object classification, successful data augmentations include color distortions and random cropping of the input images, reflecting the notion that the underlying semantic object identity does not change with color and size changes or translations of the object in the image (Chen et al., 2020; Bachmann et al., 2023). In Chapter 3.2, we develop appropriate data augmentations for spatially-embedded graphs, and more specifically for neuronal skeletons, to enable self-supervised learning for neuronal morphologies.

**Inductive biases through training objectives** The learning objective including regularization terms plays a significant role in the function a model learns. This can be nicely illustrated by examining the solutions learned for linear regression when either using  $L_1$ - or  $L_2$ -regularization during optimization. The linear regression model assumes a linear relationship between the target value  $y$  and the input values  $x$ :  $\hat{y} = \mathbf{w}^T \mathbf{x} + \epsilon$ , where  $\mathbf{w} \in \mathbb{R}^d$  are the model parameters and  $\epsilon$  is the noise term. To estimate the parameters  $\mathbf{w}$ , commonly the mean squared error (MSE) between the predicted value  $\hat{y}$  and the target value  $y$  is optimized:  $MSE = \frac{1}{n} \sum_{i=1}^N (\hat{y}_i - y_i)^2$ . If additional regularization in the form of the  $L_1$ -norm or  $L_2$ -norm on the parameters  $\mathbf{w}$  is added, the learned solutions differ: While  $L_1$ -regularization encourages sparse solutions, setting some weights to zero,  $L_2$ -regularization encourages all weights to decrease in magnitude (Bishop, 2006).

Similarly, for deep learning architectures the choice of loss function heavily influences the function learned by the network. Training classification with a cross-entropy loss encourages the model to learn the simplest predictor for the class membership, often discarding other possible predictive features in the data (Jacobsen et al., 2019; Malhotra et al., 2020). In unsupervised learning, the reconstruction loss on pixels is commonly used to train autoencoders for images. To satisfy the reconstruction objective, the model needs to reconstruct the pixels of the image including high-frequency details. This is necessary for the generation of images but might not be optimal if the goal is to learn a representation of the semantic content (see Section 2.3). In contrast, placing a reconstruction loss on higher-level features of a deep neural network is hypothesized to rather enforce learning about semantic concepts (Gatys et al., 2016; Seitzer et al., 2023). In self-supervised learning, contrastive objectives are commonly used (Chen et al., 2020). In contrastive learning, a sample is *contrasted* with a positive match, often an augmented version or a different modality of the same sample, and negative counterparts such as random other samples. The contrastive loss encourages positive pairs to have similar representations while negative pairs are separated in latent space. This leads to a representation that retains information that is present in both views or modalities of the positive pair.

**Inductive biases versus data** Strong inductive biases are especially useful in the low data regime. In general, the more constrained the hypothesis space of a model the less it needs to learn from data. Consequently, building the right assumptions into a model enables generalization from fewer training examples. On the other hand, when ample data is available, strong inductive biases can be relaxed while keeping a high performance. For instance, it has been shown that MLPs achieve

competitive performance to CNNs when trained with enough data (Bachmann et al., 2023). Similarly, the rise of vision transformers has shown that given enough data the inductive biases of convolutions are not necessary for good performance on image data and transformers can even outperform CNNs in some settings (Dosovitskiy et al., 2021). Hence a lack of inductive biases can be compensated by sufficient training data. For further discussion of the interaction of inductive biases and availability of data, see Chapter 5.

Every choice — conscious or unconscious — in the process of designing and optimizing a model has underlying assumptions and inductive biases that become part of the model. As discussed in this chapter, this is not only necessary but also desirable in order to achieve good generalization in specific tasks with better sample efficiency. When developing and evaluating unsupervised machine learning algorithms, it is therefore crucial to consider the inductive biases that are explicitly and implicitly encoded in the modeling decisions and how they affect the trained model and its generalization performance.

## 1.2 Thesis structure

This dissertation is partitioned into two main result chapters. Each chapter contains an introduction into the application domain, a summary of the respective publications as well as an additional discussion that transcends the discussion done in the original publications. Chapter 2 describes the computer vision application, namely a study that benchmarks object-centric video models and analyzes their inductive biases. In Chapter 3, we present two publications that are part of the application for neuroscience. The first publication describes the design of a novel self-supervised model for spatially-embedded graphs that learns a low-dimensional representation of the spatial structure of the graph. Here, we deliberately use inductive biases to encode domain knowledge for the design of a model for neuronal morphologies. The second publication builds on this work by applying the proposed model to a large-scale dataset of neuronal morphologies to analyze the cortical organization of excitatory neurons in the mouse visual cortex. The last two chapters of this dissertation give a broader perspective on common themes encountered in both application domains.

## 1.3 Publications

### Publications included in this dissertation

This dissertation is based on two first author, peer-reviewed articles and one first author preprint that is currently under review. The full publications are included in the Appendix. For each article the main motivation, results, and discussion are summarized in the respective chapter.

- Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Benchmarking unsupervised object representations for video sequences. *Journal of Machine Learning Research (JMLR)*, 22(183):1–61, 2021.
- Marissa A. Weis, Laura Hansel, Timo Lüddecke, and Alexander S. Ecker. Self-supervised graph representation learning for neuronal morphologies. *Transactions on Machine Learning Research (TMLR)*, 2023.

An earlier version of this work was additionally presented as a peer-reviewed poster at *COSYNE 2022*.

- Marissa A. Weis, Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G. Fahey, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, R. Clay Reid, Casey M. Schneider-Mizell, H. Sebastian Seung, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Jacob Reimer, Andreas S. Tolias, and Alexander S. Ecker. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex. *bioRxiv*, 2023.

Preprint currently under review.

### Related work not included in this dissertation

The following publication is a peer-reviewed conference paper that is not formally included in this dissertation.

- Elliot Creager, David Madras, Joern-Henrik Jacobsen, Marissa A. Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. *In Proc. of the International Conf. on Machine learning (ICML)*, 2019.

## 2 Object-centric learning

Humans perceive the world in terms of objects (Spelke and Kinzler, 2007). The visual system groups the raw visual information into semantically meaningful entities. Leveraging the compositionality of the world and representing it in terms of objects is thought to be the foundation for higher-level cognition such as language, planning and reasoning (Greff et al., 2020; Peters and Kriegeskorte, 2021).

With the advent of deep learning, the performance of computer vision models has surged. Early approaches focused on solving specific visual tasks such as object classification in images (LeCun et al., 1998; Krizhevsky, 2009; Russakovsky et al., 2015). The prevailing paradigm of the application of deep learning in computer vision was to learn one distributed representation per image (Krizhevsky et al., 2012). This works well in settings in which images contain one dominant object, such as in the omnipresent benchmarks MNIST, CIFAR and ImageNet (LeCun et al., 1998; Krizhevsky, 2009; Russakovsky et al., 2015) but it is less suitable for images that contain multiple objects. Distributed representations do not capture the compositional nature of visual scenes and have been shown to often rely on surface statistics instead of on the underlying semantic concepts as well as to fail to generalize (Brendel and Bethge, 2019; Geirhos et al., 2019; Greff et al., 2020; Geirhos et al., 2020). With increasing success in single-object classification, the field evolved towards more difficult tasks that entail reasoning over more complex scenes featuring multiple objects. Benchmarks such as COCO (Lin et al., 2014) or Cityscapes (Cordts et al., 2016) require localizing and identifying multiple objects in a scene. Considerable progress has been made in training models in a supervised fashion on those tasks. But the supervised learning paradigm comes with several drawbacks. Manual annotation of the necessary labels for training are expensive to generate<sup>4</sup> and annotations can be subjective or ambiguous. In the case of object detection, depending on context and task the delineation of an object can be difficult to assess and different granularity levels are possible.

In contrast, the goal of object-centric models is to learn representations of complex visual scenes by decomposing the scenes into their individual objects without the use of supervision and representing the objects independently. Instead of learning one distributed representation per image or video as formerly done, object-centric models learn a set of representations that each encode different parts of the image. Thus, object-centric models incorporate compositionality as an inductive bias. If models are able to take advantage of the modularity of the world, they do not need to learn about all possible object combinations but can generalize the knowledge about objects to scenes in which objects are placed in novel configurations. This leads to better sample efficiency and generalization (Dittadi et al., 2022); similar to human vision.

### 2.1 Definitions and background

**Definition of “object”** There is no generally agreed upon definition of what constitutes an object (Greff et al., 2020). Objects are entities that exhibit physical cohesion (Peters and Kriegeskorte, 2021) and can be separated from other entities, meaning that they are internally connected and externally bounded (Spelke and Kinzler, 2007). Objects show coherent motion and obey the laws of persistence (Spelke, 1990). Following Greff et al. (2020), we use the term “object” in this dissertation to refer to entities that fulfil the functional role of useful building blocks of visual scenes.

---

<sup>4</sup>However, for the task of instance segmentation recent studies have questioned the validity of this argument. For further discussion, see Sec. 2.3.



**Human perception of objects** In cognitive psychology, object representation is postulated as a core knowledge system in humans (Spelke and Kinzler, 2007). Human perception of objects and their perceived motion is governed by the constraints on objects in the physical world. The representation of objects and their motion is thought to be based on three principles: cohesion, continuity, and contact. First, objects stay internally connected and maintain their boundaries as they move. Second, objects move on connected, unobstructed paths and third, they interact only if they come in contact with each other (Spelke, 2003; Spelke and Kinzler, 2007). Together, these principles enable humans to represent objects even if they move partially or fully out of view, known as object permanence. The ability to represent objects is a prerequisite for reasoning about objects and their future behavior (Spelke, 1990, 1998).

Object representation is a property that arises early on in infancy and might even be partially innate (Spelke, 1990, 1998; Ullman et al., 2012). Infants can follow moving regions and separate them from stationary background (Ullman et al., 2012). Already two months old babies show signs of representing partially occluded objects and approximately six months old babies exhibit object permanence (Baillargeon, 1986; Spelke, 1998). Object perception is refined during infancy and increasingly complex scenes can be processed (Needham, 1998). While object representation based on spatio-temporal continuity is present already in young infants, the perception of objects based on other properties as postulated by Gestalt theory (Wertheimer, 1923) is successively acquired (Spelke, 1990). Gestalt theory states that perception “tends to assume the simplest and most regular organization that is consistent with a given visual array” (Spelke, 1990). It includes the principles of similarity, good form, good continuation and common fate. Adults use Gestalt principles in addition to spatio-temporal continuity for object perception. Only a limited amount of about three to four objects can be represented in the adult working memory simultaneously (Spelke and Kinzler, 2007; Carey, 2009). These objects are primarily tracked based on the principle of spatio-temporal continuity. Other object properties such as texture and color are secondary and might change without the represented object identity changing (Carey, 2009; Peters and Kriegeskorte, 2021).

Object representation is not unique to humans, but has also been found in animals (Spelke, 2003). Newborn chicks exhibit object representation and object permanence (Regolin and Vallortigara, 1995; Lea et al., 1996). Likewise, primates have been found to represent objects (Hauser et al., 1996).

**Compositionality as an inductive bias for object-centric vision** Grouping the raw perceptual input, i.e. pixels or intensity values as detected by photoreceptors in the retina, into a low dimensional and structured representation based on objects is a powerful inductive bias for higher-level cognition and causal understanding (Peters and Kriegeskorte, 2021; Brady et al., 2023).

Unsupervised learning of disentangled representations is impossible without inductive biases (Locatello et al., 2019). To achieve object representation, machine learning models rely on several inductive biases to decompose a scene into objects without supervision. The most important one is compositionality: Representations of individual objects are independent of each other, and objects can be flexibly recombined into new scenes. This notion can be directly built into the architecture of a model or it can be enforced by the training objective.

In object-centric models, an image is no longer represented by one distributed representation  $\mathbf{z} \in \mathbb{R}^d$ , where  $d$  is the dimensionality of the representation vector, but instead by a structured representation that consists of a set of vectors  $\mathcal{Z} = \{\mathbf{z}_0, \dots, \mathbf{z}_K | \mathbf{z}_k \in \mathbb{R}^d\}$  where each vector  $\mathbf{z}_k$  represents one separate entity in the image. These object vectors are referred to as *slots*. Slots share a common format and are ideally independent of each other (Greff et al., 2020). One common strategy to achieve the decomposition into independent objects is to restrict the capacity of the individual slot

representations in order to bias the models to finding simple building blocks of the scenes (Engelcke et al., 2020a). This inductive bias is especially powerful in combination with depth reasoning and amodal completion<sup>5</sup> of objects as occlusions of objects can be more easily resolved.

**Object-centric models** Given that object perception is an integral part of human vision, and it is thought to be a prerequisite for reasoning about the world, it is assumed to also be beneficial for computer vision. Both in terms of sample efficiency and generalization as well as to improve structural understanding of the world, it is desirable to instill machine learning models with object representations. Recently, the field of object-centric representation learning has gained traction in the deep learning and computer vision community. Object-centric models have been proposed both for images (Crawford and Pineau, 2019; Greff et al., 2019; Burgess et al., 2019; Jiang and Ahn, 2020; Lin et al., 2020; Locatello et al., 2020; Sajjadi et al., 2022) and for videos (Kosiorok et al., 2018; Veerapaneni et al., 2020; Jiang et al., 2020; Singh et al., 2022b), where the additional challenge is to keep a stable percept of objects over time.

Object-centric models show a considerable variability in their design choices with respect to how they infer compositional representations from visual scenes, the format of their object representations as well as how they generate visual scenes from object representations.<sup>6</sup> In the following, we highlight some popular and influential design choices.

Object-centric models commonly learn a set of latent vectors, called slots. Each slot represents part of the scene, which can either be an object or (part of) the background. However, at initialization all slots are equal, thus requiring a mechanism to break the symmetry in order to prevent slots from representing the same object (Greff et al., 2020). To bind slots to distinct parts of the image different models use different strategies. One strategy is to *sequentially* bind slots to objects (Eslami et al., 2016; Kosiorok et al., 2018; Crawford and Pineau, 2019; Burgess et al., 2019; Engelcke et al., 2020b; von Kügelgen et al., 2020). This ensures that slots only represent parts of the scene that have not yet been explained by previous slots. However, it introduces an potentially arbitrary order onto the object representations. A different approach is to instead bind slots to spatial locations in the scene by dividing the image into to a regular grid and allowing the discovery of only one object per grid cell (Crawford and Pineau, 2019; Lin et al., 2020; Jiang and Ahn, 2020; Jiang et al., 2020) or by introducing competition across slots to explain pixels (Greff et al., 2017, 2019; Locatello et al., 2019). This approach allows for a *parallel* refinement of the object representations. One successful line of work using parallel refinement is based on SLOTATTENTION (Locatello et al., 2019). SLOTATTENTION achieves symmetry breaking by sampling initial slot representations from a learned distribution and binding them via attention to parts of the input and then iteratively refining them. To prevent slots from representing the same object, slots compete for binding to pixels. SLOTATTENTION inspired many follow-up works that use the SLOTATTENTION module in their architecture due to its flexibility and good performance (Singh et al., 2022a,b; Kipf et al., 2022; Elsayed et al., 2022; Sajjadi et al., 2022; Seitzer et al., 2023; Zadaianchuk et al., 2023).

A different categorization of object-centric models can be based on the *format* of the learned representations. One class of object-centric models learns a continuous representation per object and image-sized segmentation masks (Greff et al., 2017; van Steenkiste et al., 2018; Greff et al., 2019; Veerapaneni et al., 2020; Engelcke et al., 2020b; Weis et al., 2021; Engelcke et al., 2021; Zoran et al., 2021; Kabra et al., 2021; Kipf et al., 2022; Elsayed et al., 2022; Singh et al., 2022b). The other class

---

<sup>5</sup>Amodal completion refers to the ability to represent occluded parts of objects. Even if an object is only partially visible, the full object is perceived (Chen et al., 2016).

<sup>6</sup>See Yuan et al. (2023) for a recent review.

pre-imposes additional structure onto the object representations (Eslami et al., 2016; Kosiorrek et al., 2018; Crawford and Pineau, 2019; Lin et al., 2020; Jiang et al., 2020): Specific dimensions of the object representation are fixed to encode presence, location, size and appearance of the respective object. The latter commonly use `SPATIAL TRANSFORMERS` (Jaderberg et al., 2015) in order to render the objects from the factored latents. This enables them to learn object-sized segmentation masks as opposed to image-sized masks, which is computationally more efficient, especially with increasing numbers of objects per scene.

Despite the fact that most models are phrased such that object latents can be sampled at inference time, they usually do not act as proper generative models for the scene as a whole as only the individual objects can be sampled but not the interplay between them in the scene (von Kügelgen et al., 2020). Thus, some works have focused on developing valid scene generative models using object-centric latents (Jiang and Ahn, 2020; Engelcke et al., 2020b; von Kügelgen et al., 2020; Engelcke et al., 2021; Wang et al., 2023a; Tangemann et al., 2023).

Most works utilize a reconstruction or next-step prediction objective based on the pixel values to train the object-centric models without supervision in an autoencoder setting (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020; Veerapaneni et al., 2020). Here each slot is decoded separately into a reconstructed image and an associated object mask. The final image is then created by summing over the individual images weighted by the decoded masks. Alternatively, in the case of models with factored latents, the objects are rendered and placed on the background using a `SPATIAL TRANSFORMER` to obtain the image reconstruction (Lin et al., 2020; Jiang et al., 2020).

For a long time, object-centric models were only capable of successfully learning to decompose scenes into objects in simple and synthetic toy settings. Recently, some approaches have scaled to more natural data by employing supervisory signals such as location cues, depth or optic flow (Kipf et al., 2022; Elsayed et al., 2022) or by utilizing features of pre-trained networks and a reconstruction objective in latent space (Seitzer et al., 2023; Zadaianchuk et al., 2023). However, there still remains a gap to real-world applications (Yang and Yang, 2022; Zadaianchuk et al., 2023).

## 2.2 Benchmarking unsupervised object representations

*This section summarizes:*

Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Benchmarking unsupervised object representations for video sequences. *Journal of Machine Learning Research (JMLR)*, 22(183):1–61, 2021.

*The full publication can be found in the appendix on page 69.*

### Motivation

Object-centric models learn representations of visual scenes by decomposing them into objects without the use of supervision and representing the objects individually. Determining the delineation of individual objects can be highly ambiguous in still images. In contrast, videos are particularly useful to learn object-centric representation, since common motion gives a strong cue for objectness and information can be accumulated and disambiguated over time. In the emergent field of object-centric learning for video data, several promising models have been proposed (Kosiorsek et al., 2018; He et al., 2019; Veerapaneni et al., 2020; Jiang et al., 2020). However, each study used an individual evaluation protocol to determine the performance of their model, ranging from evaluating tracking to object counting and block stacking tasks in a reinforcement learning setting. Furthermore, the studies used different synthetic datasets for training and evaluation. Therefore, a fair comparison between the models was impossible and their capabilities were difficult to judge. To move forward, a principled analysis of the models and their inductive biases was lacking.

We therefore proposed a benchmark to unify the evaluation of unsupervised object-centric models for videos and to analyze their strengths and failure modes in a principled way. As this was the first comparative study of object-centric video models, we focused on the evaluation of basic perceptual abilities which are the prerequisite for any further object-centric task that reasons over or manipulates objects in visual scenes. Object-centric video models need to fulfil several key criteria: They need to (1) identify the objects in a scene, (2) accurately segment them, as well as (3) maintain a consistent representation of the objects over time. To quantitatively assess these perceptual abilities *multi-object tracking* (MOT) can be used (Bernardin and Stiefelhagen, 2008). MOT metrics are well established in supervised tracking as they give a good indication of basic perceptual abilities including figure-ground segmentation, object detection and object tracking (Milan et al., 2016; Dendorfer et al., 2020; Sun et al., 2022). We therefore used them to systematically evaluate the capabilities of object-centric video models in our benchmark.

### Results

We systematically tested the performance of four state-of-the-art models for object-centric learning from videos and two baselines on their basic perceptual abilities. To this end, we set up a benchmark with four procedurally generated datasets featuring multi-object scenes with increasing visual complexity and different challenging out-of-distribution test sets. The datasets span multiple levels of visual complexity with respect to variability in objects, background and motion patterns. The benchmark includes datasets with both 2D and 3D scenes. Datasets progress from uniformly

colored to textured objects and background. Objects in the scene as well as the background can exhibit no motion, linear motion and non-linear motion depending on the dataset. Additionally, we defined seven out-of-distribution test sets that feature especially challenging scenarios such as heavy occlusion, objects with the same color, objects of certain sizes as well as objects with changing properties over time such as continuous size, color and rotation changes. These test sets permitted us to diagnose the specific scenarios in which the models struggle. Together, this benchmarking suite enabled us to characterize the failure modes of object-centric videos models and their intrinsic inductive biases: All models struggled with occlusion handling. Even those that are equipped with an explicit mechanism to reason over depth of objects were not able to correctly resolve occlusions. The models showed a bias towards relying on color as a cue for objectness, struggling to separate objects with similar colors and completely failing at identifying textured objects. The spatial-transformer based models were additionally prone to failures when objects in the scene had differing sizes and when the object sizes deviated from their size priors. All models completely failed to learn to identify objects in scenes with natural image statistics, even though the scene properties and motion patters were deliberately kept simple. All in all, our study demonstrated that the examined models for object-centric representation learning for videos only worked in limited settings for synthetic scenes with their inductive biases tuned to the specific scene statistics.

## Discussion

Our results demonstrated that none of the examined object-centric models for videos are able to consistently detect and track objects even in relatively simplistic synthetic scenes and they completely fail to detect objects in scenes with more natural image statistics. Their inductive biases are tuned to synthetic datasets and do not generalize to natural scenes.

The models mostly performed color segmentation; merging objects of similar colors into one object slot and failing to detect objects that were similarly colored to the background. This reveals a mismatch between the intuitions behind object-centric learning and what the models learn in practice. Separating objects even of similar color and encoding their simple shapes should be more efficient in terms of representational complexity than encoding the complex shapes of multiple merged objects. But none of the models performed depth reasoning with amodal segmentation masks well in practice. Instead, the models learned to encode the more complex modal object masks and hence rather merged objects of similar colors. Enabling depth reasoning in conjunction with amodal masks would facilitate resolving occlusion, which is a key capability of human object vision and necessary for stable object representations. Color segmentation works fairly well in the synthetic datasets with uniformly colored objects, but fails in scenes with textured objects.<sup>7</sup> The limited capacity of the object slots is not sufficient to represent high-frequency textures. Scaling the capacity per object slot does not solve this problem (Papa et al., 2022) as this weakens the prior to finding “simple” building blocks. In addition, the inductive biases of the reconstruction objective are not well aligned with the task of object-centric learning in datasets with high visual complexity. It requires the reconstruction of pixel-level details such as high-frequency textures. Objects however are rather characterized by their shape than by their texture (Geirhos et al., 2019; Karazija et al., 2021).

To scale object-centric models to natural data it is necessary to move away from these strong inductive biases that were developed based on simple synthetic datasets as they do not generalize to datasets with higher visual complexity. In the following section, we discuss the progress that has been made in object-centric learning since the publication of our study.

---

<sup>7</sup>Similar results were shown for object-centric models of static scenes (Karazija et al., 2021; Yang and Yang, 2022).

## 2.3 Advances in object-centric learning

**Scaling to natural data** Our benchmark has shown that unsupervised object-centric video models are restricted to synthetic datasets and even for those, they struggle to solve more complex situations such as scenes featuring occlusion or textured objects. To move towards solving unsupervised object representation learning for real-world data, better inductive biases or additional supervision are needed. Since our benchmark was published, several studies have been released that address some of the problems we identified. The main objective is to move away from simple synthetic scenes and to scale the approaches to (more) natural data. The proposed solutions can be roughly classified into two approaches: (1) Moving away from unsupervised learning by incorporating supervisory signals or (2) replacing the reconstruction loss on the RGB-pixel values, or a combination of the two.

In the first case, additional supervisory signals are used in order to detect and segment objects in the images or videos. Kipf et al. (2022) proposed to condition the object slots on the object location in the first frame of a video, reducing the problem to tracking the objects through the video without having to discover them first. Furthermore, several works have explored using additional modalities such as depth or optic flow as supervisory signals in the loss (Kipf et al., 2022; Elsayed et al., 2022; Karazija et al., 2022; Bao et al., 2023) as detailed below. Additional supervision in the form of object masks obtained by motion segmentation has been used to guide object discovery (Bao et al., 2022, 2023). Adding weak supervision improves performance of object-centric models for datasets with higher visual complexity, but requires sufficiently large annotated or multi-modal datasets for training, which can be difficult to obtain for real-world datasets.

Second, object-centric models are commonly trained using a reconstruction loss in pixel space. To optimize this loss, models are required to reconstruct high-frequency details such as textures in the images or videos. However, this might not be relevant for representing objects, and competes with the limited capacity of the individual object slots. Therefore, a training objective that is better aligned with the goal of learning about semantically meaningful objects would be desirable. One line of work has replaced the RGB-reconstruction with the reconstruction of optic flow fields (Kipf et al., 2022; Bao et al., 2023) or depth values (Elsayed et al., 2022; Bao et al., 2023). Optic flow is an informative signal for grouping image regions into objects as common fate is a strong cue for objectness that is also heavily used by human vision (Wertheimer, 1923; Spelke, 1990). Both optic flow and depth tend to be fairly uniform within an object; especially for rigid objects which were considered in the respective studies (Kipf et al., 2022; Karazija et al., 2022; Elsayed et al., 2022). Therefore, they provide a good supervisory signal for object detection and segmentation. An alternative approach is to replace the RGB-reconstruction with the reconstruction of higher-level latent features extracted from a deep neural network (DNN). These could either stem from a pre-trained vision model (Seitzer et al., 2023; Zadaianchuk et al., 2023) or from a variational autoencoder with quantized latent space trained in parallel (Bao et al., 2023). High-level features of DNNs are thought to encode semantically more meaningful information compared to pixel intensity values (Gatys et al., 2016; Amir et al., 2022; Seitzer et al., 2023), which makes them suitable for object-centric learning. Other studies introduced auxiliary losses based on cycle-consistency objectives in features space additionally to the reconstruction loss (Didolkar et al., 2023) or as the sole objective on top of pre-trained latent features (Wang et al., 2023b) to guide the segmentation of objects. Bao et al. (2023) compared different reconstruction targets for object-centric learning including RGB-values, depth, optic flow and quantized latent features and concluded that the latter was best suited for object discovery.

While the above listed changes have led to substantial progress in object-centric learning going from sprite-based and synthetic to more realistic datasets, state-of-the-art models are still far away from

solving object representation learning in real-world videos with all its challenges. Recent models show promising performance on complex synthetic datasets that mimic certain aspects of natural videos (Karazija et al., 2021; Greff et al., 2022) or constrained real-world datasets such as YouTube-Aquarium (Singh et al., 2022b) or streets scenes with uniform object types and motion patterns (Geiger et al., 2013; Sun et al., 2020). However, unsupervised object discovery in unconstrained real-world datasets remains an open problem (Yang and Yang, 2022; Zadaianchuk et al., 2023).

**Evaluation** Encouragingly, since the publication of our study and the development of object-centric learning into a more mature field of study, newly published models are now routinely evaluated on commonly used datasets and metrics, facilitating the comparison of different methods. Furthermore, with the development of more potent object-centric models, multiple datasets and benchmarks have been published to test the increasing abilities of the models, as well as closing the gap between the toy datasets that have been previously used in object-centric learning and natural images (Karazija et al., 2021; Greff et al., 2022; Tangemann et al., 2023). These are still synthetic datasets but they exhibit more complex image statistics that form a good trade-off between being controllable with access to ground truth labels for evaluation and approaching the actual goal, namely finding models that work well on natural scenes with all their challenges. Recent publications have also started to evaluate their models on real-world datasets in constrained (Singh et al., 2022b; Elsayed et al., 2022) and unconstrained settings (Zadaianchuk et al., 2023).

Evaluation of object-centric models has mostly been done in terms of their segmentation performance. While localizing and segmenting objects in a scene is a prerequisite for building object representations, it does not evaluate whether the representations can capture any further properties of the objects. A correct segmentation shows that the object representations accurately encode the location and the shape of the object, and in video models potentially also the motion patterns. But so far many other object properties have been disregarded in the evaluation of object-centric representations with the exception of some studies that have assessed the capability of decoding object properties such as material or color from the learned latents (Locatello et al., 2020; Dittadi et al., 2022). Other studies specialized on evaluating how specific properties of the models affect performance such as the role of the bottleneck of the latent embeddings (Engelcke et al., 2020a) or the number of slots (Zimmermann et al., 2023). More work is needed here in the future to get a deeper understanding of the performance of object-centric models that go beyond their segmentation performance.

**Is unsupervised object-centric learning still necessary?** Unsupervised object-centric learning is mostly pitched under the assumption that data annotation is expensive and labeling objects can be ambiguous. Therefore, discovering objects purely from raw pixel data is desirable, however difficult as seen by the limited progress that has been made in the last years. In contrast, supervised learning has generated many models that successfully detect and track objects in real-world datasets (Zhang et al., 2022; Kirillov et al., 2023; Zong et al., 2023). Therefore, one might ask if and why unsupervised learning is still relevant in this domain.

There are several aspects to consider here. First, supervised segmentation is a self-enhancing process that requires increasingly less manual labeling effort. Segmentation models are able to generate data annotations that can be refined by human annotators, requiring less effort than labeling the images from scratch (Kirillov et al., 2023). These annotated datasets can then be used to train better models, which in turn generate better segmentations that require less manual refinement. Recently, Kirillov et al. (2023) demonstrated that iterating over model training and dataset generation with successively less human intervention can yield both a well-performing model and a large-scale labeled dataset for instance segmentation. However, the model proposed by Kirillov et al. (2023) is foremost an

instance segmentation network. Therefore, it does not equip us with a general purpose representation of objects that can be used for downstream tasks out of the box.<sup>8</sup> But arguably once an accurate segmentation of objects has been determined, encoding the object information in a representation is simpler than extracting individual object representations from pixel values of a whole scene. Indeed, it is assumed that human vision follows a segmentation first, representation second approach (Peters and Kriegeskorte, 2021). The ambiguity of how to partition scenes into objects however remains. Kirillov et al. (2023) addressed this by predicting multiple valid segmentations of different granularity levels for a given input at training time. In general, neither supervised nor unsupervised learning on its own can resolve this issue, but rather additional information about the downstream tasks is required to infer the correct granularity level of the representation.

The second claim of object-centric learning is that through the architectural compositionality bias the models exhibit superior robustness and generalization capabilities (for further discussion of compositionality as a useful inductive bias, see Chapter 4). Generalization of object-centric models to different out-of-distribution scenarios has been demonstrated empirically in synthetic settings (Singh et al., 2022a; Dittadi et al., 2022). Therefore, taking inspiration from the structured latent approach for future computer vision models is promising. But compositionality could also be incorporated in supervised models or arise in unsupervised models trained on enough data without explicitly building it into the architecture. Recent studies on large self-supervised vision and vision-language models indicate that these models encode high-level semantic information including information about objects parts that are shared across object classes (Amir et al., 2022; Oquab et al., 2023). However, early studies suggest that language-vision models currently do not exhibit compositionality in multi-object settings (Bogin et al., 2021; Bommasani et al., 2022; Lewis et al., 2023; Ma et al., 2023). But if more powerful vision foundation models are developed in the future, object-centric representations might emerge as a side product or conversely, when managing to train object-centric models on large real-world data corpora, general vision foundation models might emerge.

While computer vision is a domain in which abundant data with associated labels is available, this often does not hold for other domains or even for subdomains of computer vision, for instance medical images or rare object classes. Therefore, object-centric learning is an ideal test bed to discover good inductive biases for unsupervised learning and for compositional generalization. This knowledge can then be transferred to data domains in which we do not have enough annotated data for robust supervised learning without strong inductive biases (see Chapter 5 for further discussion).

Furthermore, unsupervised object-centric learning might be interesting in its own right to study human vision. Object representation is thought to arise unsupervised or weakly-supervised in infant vision and parts of it might even be innate (Spelke, 1990; Carey, 2009; Ullman et al., 2012). Using unsupervised object-centric learning and comparing it to human vision might generate insights into the inductive biases that human vision encodes. Uncovering the inductive biases which lead to a robust visual system with exceptional generalization capabilities would be both highly interesting from a scientific point of view to further our understanding of how the human brain works as well as from an engineering perspective to build better and more robust artificial vision systems.

Taken together, while unsupervised object-centric learning is currently not competitive to supervised learning for tasks such as instance segmentation where sufficiently large labeled training datasets are available, it can still lead to interesting insights both into the inductive biases of biological vision as well as into useful inductive biases for unsupervised learning to enable sample-efficient learning and systematic compositionality.

---

<sup>8</sup>However, naively, the masked image embeddings could be used as an object representation.



### 3 Modeling of neuronal morphologies

The human brain contains billions of neurons with trillions of connections in the form of synapses between them, rendering it a highly complex system that is nearly impossible to grasp in its entirety. To further our understanding of how this system functions, neuroscientists are endeavoring to break it down into cells as its building blocks and categorize these into different cell types. Defining such cell types is a useful abstraction that facilitates the study of neural circuitry, cortical organization and relationships to functional properties (Masland, 2004; Fishell and Heintz, 2013; Zeng and Sanes, 2017; Mukamel and Ngai, 2019; Zeng, 2022). Early descriptions of cell types based on the cells’ shapes, commonly referred to as morphology, date back to Ramón y Cajal in 1911. Since then, numerous criteria for classifying neurons into cell types have been proposed based on their morphological, molecular or functional properties. Despite trying to classify neurons for decades, precisely how to define cortical neuronal cell types remains an open research question (Zeng and Sanes, 2017).

Neurons exhibit intricate three dimensional (3D) shapes that include branching dendrites, tiny protrusions such as spines, but also long-reaching axons that can span the entire brain. Therefore, neuronal morphologies are difficult to summarize into a more tractable representation from which cell types can be easily inferred. Formerly, there were two approaches to morphological cell type classification: First, experts in the field inspect each cell individually to determine its cell type (Ramón y Cajal, 1911; Larkman, 1991; DeFelipe et al., 2013; Markram et al., 2015). This approach suffers from being underpowered (Zeng and Sanes, 2017). It can only take a limited number of neurons into account and the selection of cells is often biased towards neurons that are easy to image and reconstruct. Furthermore, the classification can only be based on human-accessible features (Zeng and Sanes, 2017) and expert annotation can be unreliable. Studies have shown that there can be a high variation between experts’ classification (DeFelipe et al., 2013). Additionally, manually classifying each cell is very time consuming, and thus does not scale to larger datasets. For these reasons, the field has moved towards automatically classifying cells into cell types based on expert-defined features. These features can be calculated from the reconstructions of the neurons, such as the number of branching points or the bifurcation angles, and can subsequently be used as input features to classifiers or clustering methods (Oberlaender et al., 2011; Marx and Feldmeyer, 2012; Narayanan et al., 2017; Gouwens et al., 2019). This approach remedies the scalability problem of manual annotation of the cells, but it retains human biases by only taking a pre-selected subset of morphological features into account. Features that are not obvious to humans or that are difficult to quantify are disregarded, hindering an unbiased census and the discovery of new cell types.

While the ever-increasing dataset sizes are problematic for manual annotation, they provide optimal conditions for unsupervised machine learning techniques, that have shown surging performances over the last decade in numerous domains. These techniques offer the possibility to identify patterns in a data-driven way, dispensing with the need for human annotations and manual feature selection. This makes them especially suitable for the application to neuronal morphology modeling.

This dissertation describes the design of a machine learning method to learn a descriptor of neuronal morphologies purely from data. First, we ask what are the appropriate inductive biases in designing a machine learning model that is capable of encoding complex neuronal morphologies into a low dimensional embedding? In Section 3.2, we answer this question: Building on recent developments in self-supervised learning, we design a novel graph attention model that takes as its input spatially-embedded graphs — such as neuronal skeletons — and embeds them in a low dimensional latent space. By doing so, every neuron is assigned a “bar code” that captures the essence of its 3D shape. This model addresses the shortcomings of the previous methods. First, it is scalable to a large number

of neurons, making it capable of dealing with growing dataset sizes. Second, it does not rely on manually defined features. Instead, the model learns from the data which features are relevant to distinguish individual neurons, thereby reducing human biases in the feature selection. This allows the model to consider features that have not been previously accounted for, opening up the possibility of discovering new cell types. Subsequently, in Section 3.3, we apply the model to a large-scale dataset of the mouse visual cortex containing tens of thousands of cells to perform a comprehensive census of excitatory neurons in the mouse visual cortex.

### 3.1 Definitions and background

**Cortical organization principles** The cerebral cortex forms the outer layer of the brain in humans and other mammals. It can be partitioned into multiple areas that are responsible for processing sensory and motor information as well as performing higher-level associational tasks (Zeng and Sanes, 2017). Sensory information is processed in dedicated regions of the cortex which are associated with individual sensory modalities. As such there is a visual cortex, a somatosensory cortex and an auditory cortex, among others (Kandel et al., 2000).

The *visual cortex* in mammals consists of multiple areas that are interconnected by feedforward and feedback connections. The theory of cortical hierarchy postulates that feedforward connections drive information flow from the primary visual cortex (V1) to increasingly specialized areas, while feedback connections influence neural responses based on past experiences, attention and perceptual tasks (Gilbert and Li, 2013; D’Souza et al., 2022). For primates, such a hierarchical organization of the visual areas was found (Felleman and Van Essen, 1991). Similarly, there exists evidence that rodents, such as rats (Coogan and Burkhalter, 1993) and mice (Glickfeld and Olsen, 2017; D’Souza et al., 2022) likewise exhibit a hierarchical organization of cortical visual areas. However, the hierarchy in mouse visual cortex is deemed to be shallower than in primates (D’Souza et al., 2022).

The mouse visual cortex can be further subdivided into areas, namely the primary visual cortex (V1) and higher visual areas: lateromedial (LM), laterointermediate (LI), anterior (A), anterolateral (AL), rostromedial (RL), anteromedial (AM), posterior (P), postrhinal (POR), and posteromedial area (PM), that are thought to fulfil different functionalities (Wagor et al., 1980; Wang and Burkhalter, 2007; Marshel et al., 2011; D’Souza et al., 2022). LM is commonly believed to be the second processing stage after V1, akin to the secondary visual cortex in primates (Wang and Burkhalter, 2007; D’Souza et al., 2022). In this dissertation, we focus on analyzing the organization of the mouse primary visual cortex (V1) as well as the higher visual areas RL and AL, which are assumed to be part of the second and third processing stage of the dorsal stream, respectively (D’Souza et al., 2022).

**Neuronal morphologies** The cerebral cortex of a mouse consists of approximately 14 million neurons (Herculano-Houzel et al., 2015). Neurons come in a vast variety of shapes and forms. They have three main morphological compartments: soma, dendrites and axon (Kandel et al., 2000). The soma is the cell body that contains the nucleus and the machinery necessary for cell metabolism. Dendrites are highly bifurcated processes branching away from the soma. They receive input from other neurons and propagate it to the soma. In contrast, the axon is a long and thin process that transmits information to other neurons. It can project locally or to long-range targets. In a subtype of neurons called pyramidal cells, dendrites can be classified as apical or basal (Kandel et al., 2000). Apical dendrites originate from the apex of pyramidal cells and usually run up towards the pia mater, or in short pia, a membrane enveloping the brain. Horizontal and inverted neurons form an exception to that rule: Their apical dendrites run horizontally or down towards the white matter, respectively.

Apical dendrites can have multiple oblique dendrites branching off from their main shaft and the top of the apical dendrite can be either tufted or end in a tuft of various degrees (Hattox and Nelson, 2007; Oberlaender et al., 2011; Markram et al., 2015). Dendrites originating from the base of the soma are called basal dendrites (Kandel et al., 2000). To characterize the morphology of a neuron typically the branching patterns of the apical and basal dendrites and targets of the axon are described.

**Neuronal connectivity** Neurons connect and transmit information to each other via synapses. Synaptic connections are directional; the signaling neuron is called the presynaptic neuron, and the receiving neuron is called the postsynaptic neuron. Individual neurons commonly receive inputs from thousands of presynaptic neurons. They integrate this information and forward the aggregated information to postsynaptic neurons. The synapse is located on the axon of the presynaptic neuron. On the postsynaptic neuron, the synapse can either be located on small protrusions of the dendrites, called the spines, or directly on the dendrite (Kandel et al., 2000).

**Neuronal cell types** To understand the organization of the brain, neuroscientists have tried to break down its complexity into basic building blocks, namely into cell types. While there is no general agreed upon definition of neuronal cell types, they can be broadly defined as a group of neurons with homogeneous properties within the group that differ with respect to the properties of other cells (Masland, 2004; Fishell and Heintz, 2013; Zeng and Sanes, 2017; Mukamel and Ngai, 2019; Zeng, 2022).

Cortical neurons can be subdivided into two classes: Excitatory neurons are characterized by spiny dendrites and the use of glutamate as a neurotransmitter. They project to local or long-range targets. Inhibitory interneurons on the other hand primarily form local connections and are characterized by aspiny or sparsely spiny dendrites. Interneurons use  $\gamma$ -amino-butyric acid (GABA) or neuropeptides as neurotransmitters (DeFelipe et al., 2002; Nelson et al., 2006; Zeng and Sanes, 2017; Gouwens et al., 2019). Excitatory neurons typically make up 80% or more of neurons in the cortex (DeFelipe and Fariñas, 1993; DeFelipe et al., 2002; Masland, 2004; Markram et al., 2015). They can be further subdivided into cell types based on their morphological (Ramón y Cajal, 1911; Jiang et al., 2015; Markram et al., 2015; Gouwens et al., 2019), functional (Markram et al., 2015; Gouwens et al., 2019; de Vries et al., 2020) and transcriptomic (Nelson et al., 2006; Zeisel et al., 2015; Tasic et al., 2016; Saunders et al., 2018; Tasic et al., 2018; Yao et al., 2021, 2023) properties or combinations of those (Markram et al., 2015; Cadwell et al., 2016; Fuzik et al., 2015; Gouwens et al., 2019; Scala et al., 2019, 2021; BRAIN Initiative Cell Census Network, 2021). Electrophysiological properties include the resting potential and firing rates of the neurons and their tuning curves. Molecular analyses focus on the genetic makeup of the neurons, their protein composition or transcriptome. Typical morphological properties that have been used to define cell types include the shape and branching patterns of dendrites and axons, as well as the shape of the soma and more fine-grained features such as spine densities (Zeng and Sanes, 2017). We focus on the analysis of excitatory cortical neuron types based on morphology in this dissertation.

**Laminar organization** The cortex of mammals is organized into layers. Depending on the brain area, the number and thickness of layers and their function can vary. Typically, the cortex consists of six layers which are numbered increasingly from pia to white matter. Each layer is defined by its distribution of cell types as well as by the inputs and outputs the cells in it receive (Brodmann, 1909; DeFelipe et al., 2002; Kandel et al., 2000). In Section 3.3, we analyze the organization of the mouse visual cortex which follows the typical six-layer structure (Senzai et al., 2019) and focus on the description of excitatory neurons. Excitatory neurons are found in cortical layers II–VI and

show a specialized distribution over layers (DeFelipe et al., 2002): Layer I (L1) consists of dendrites originating from excitatory neurons based in the layers below and axons projecting to other areas of the cortex. Layers II and III (L2/3) mainly contain small pyramidal cells. Layer IV (L4), as the input layer, primarily receives sensory input from the thalamus (Harris and Shepherd, 2015). Layer V (L5) is known for its large pyramidal neurons that constitute the major output pathways projecting to other cortical and subcortical areas. Layer VI (L6) has a heterogeneous neuronal population and transitions into the white matter within which axons to and from other cortical areas run (Kandel et al., 2000).

**Excitatory morphological cell types** Excitatory neurons can be classified into cell types based on different properties of their morphologies, their projection patterns as well as their layer of origin. The axonal projection patterns define three broad classes: Intratelencephalic (IT) neurons project only within the telencephalon, which includes the cerebral cortex as well as several subcortical regions. IT neurons can be found in layers II–VI. Pyramidal tract (PT) neurons project to subcerebral areas such as the brainstem, the spinal cord and the midbrain. They are only found in layer V. Corticothalamic (CT) neurons project to the thalamus and are based in layer VI (Harris and Shepherd, 2015; Tasic et al., 2018; Scala et al., 2021; Young et al., 2021; Zhang et al., 2021). Some specialized excitatory classes have been consistently described in the literature: L4-IT neurons have a crucial role in the circuit as they receive the main input from the thalamus but few excitatory connections from other layers. They can be subdivided into three classes that show varying prevalence in different species and cortical areas: pyramidal, star-pyramidal and spiny stellate cells (Oberlaender et al., 2011; Harris and Shepherd, 2015). Their projections mostly run locally. L5-PT, also called extra-telencephalic (ET) neurons, are large neurons characterized by thick apical tufts that run into layer I. They receive input from local IT cells, as well as the thalamus and cortical areas (Harris and Shepherd, 2015). They scarcely project to local cells, but mainly have long-range projections to subcortical and subcerebral areas. Another distinct subpopulation of L5 neurons are the so-called near-projecting (NP) neurons (BRAIN Initiative Cell Census Network, 2021; Zhang et al., 2021; Schneider-Mizell et al., 2023) that exhibit sparse and long basal dendrites and are targeted by a specialized group of interneurons (Schneider-Mizell et al., 2023). L6-CT cells differ from L6-IT cells in terms of molecular properties (Greig et al., 2013; Harris and Shepherd, 2015) and receive diverse input from local neurons as well as long-range inputs from cortical areas (Thomson, 2010; Harris and Shepherd, 2015). Besides the typical IT neurons, layer VI also contains horizontal and inverted neurons. Finally, L6 subplate neurons originate in the lower part of layer VI and mostly project locally (Gouwens et al., 2019; Zeng and Sanes, 2017). Depending on the study, further morphological classes of excitatory neurons are distinguished based on their layer of origin and dendritic branching patterns.

**Morphometrics** Morphometrics denote quantitative measurements of the shape or form of an object or organism (Oxford English Dictionary, 2023). In neuroscience, the term is used to describe quantitative summary statistics of neuronal morphologies. Morphometrics have been widely used to automatically extract features that represent neuronal morphologies (Ascoli et al., 2008; Scorcioni et al., 2008; Oberlaender et al., 2011; Marx and Feldmeyer, 2012; Narayanan et al., 2017; Wang et al., 2018; Gouwens et al., 2019; Laturnus et al., 2020; Schneider-Mizell et al., 2023). These can then be used as feature representations for statistical data analysis and can serve as input to machine learning methods such as classifiers or clustering methods for automated cell type classification. To calculate morphometrics, neuronal reconstructions are commonly skeletonized (Celi et al., 2023) and the morphometrics are then computed based on the resulting graph or skeleton (Laturnus et al., 2020). Typical morphometrics include lengths of the different neuronal compartments, diameters, and their bifurcation angles, among others (Scorcioni et al., 2008). As they describe local features of the

morphology, a large set of these morphometrics is usually combined to gain a meaningful description of the overall shape of the neuron. Therefore, significant effort goes into the design and selection of the used morphometrics. However, which features are chosen is subjective and can be inconsistent between studies. Furthermore, the features are often tailored to the studied neuron population and are thus not generally applicable. The use of disparate sets of morphometrics in different studies can lead to inconsistent classification outcomes and hinders comparability.

Morphometrics can be seen akin to manually engineered features in computer vision and as such exhibit similar limitations (see Section 1.1). We therefore chose to learn a morphological descriptor directly from the data using self-supervised learning, avoiding the need for manual feature selection. The next section introduces our proposed model for neuronal morphologies.

## 3.2 Representation learning for neuronal morphologies

*This section summarizes:*

Marissa A. Weis, Laura Hansel, Timo Lüddecke, and Alexander S. Ecker. Self-supervised graph representation learning for neuronal morphologies. *Transactions on Machine Learning Research (TMLR)*, 2023.

*The full publication can be found in the appendix on page 133.*

### Motivation

The morphology of cortical neurons is highly complex and exhibits a great variety between neurons. To analyze the intricate 3D structures of neurons, there mainly have been two approaches so far. Traditionally, experts have manually classified neurons into cell types based on their morphology by visual inspection (Ramón y Cajal, 1911; Larkman, 1991; DeFelipe et al., 2013; Markram et al., 2015). More recently, researchers have tried to automate this process by manually defining features that can be computed from the skeletons of the neurons (Oberlaender et al., 2011; Marx and Feldmeyer, 2012; Narayanan et al., 2017; Gouwens et al., 2019). These summary statistics, known as morphometrics, are then used as the input features for classifiers. Manually classifying neurons is restricted to small dataset sizes. Furthermore, this method is prone to bias as experts do not necessarily agree on the cell type (DeFelipe et al., 2013). The computation of summary statistics solves the scalability issue but is still biased by the manual feature definition process towards features that experts deem important and that are visible to the human eye. Thus, it might not be a comprehensive representation of the underlying morphology.

In computer vision, the paradigm shift from manually defining features to learning features in a data-driven way has greatly accelerated progress on difficult tasks. Recent advances in recording techniques and the resulting increase in datasets sizes in neuroscience provide the means to use data-driven unsupervised machine learning methods for the analysis of neuronal morphologies and promise that the same principle can be applied here. Therefore, we designed a self-supervised machine learning model to learn low-dimensional embeddings that capture the essence of 3D morphologies of neurons in a data-driven way. This alleviates the need to manually annotate the data and does not pre-impose human biases through manual feature selection. The resulting embeddings can be used as a compact representation of the neuronal morphologies for downstream analyses such as cell type classification.

### Results

We proposed GRAPHDINO, a self-supervised learning algorithm, that can encode complex 3D neuronal skeletons in the form of graphs into low-dimensional latent embeddings. GRAPHDINO is based on a novel attention mechanism that takes both the global graph structure as well as local node neighborhoods into account. Thus, it can efficiently process the sparse and long-ranging branches typical for neuronal skeletons. Furthermore, we defined tailored data augmentation strategies to enable self-supervised learning on spatially-embedded graphs and to incorporate domain knowledge into the model.

Our results demonstrated that our proposed model, GRAPHDINO, successfully captures the essence of 3D graphs in low-dimensional latent embeddings. As a proof of concept, we used a synthetic dataset, that gives us access to ground truth class information, to show that our novel adjacency-conditioned attention mechanism (AC-ATTENTION) is able to recover information contained only in the graph connectivity. We additionally demonstrated that GRAPHDINO using AC-ATTENTION outperformed traditional message passing based GNNs for neuronal cell type classification. GRAPHDINO’s morphological embeddings learned to differentiate between spiny and aspiny neurons based solely on their dendritic morphology without the need for supervision during training. Furthermore, we were able to recover known excitatory cell types by clustering the latent embeddings and outperformed approaches that use manually selected morphometrics. In subsequent analyses, we showed that the embeddings can be used to predict distinct morphological characteristics such as tuftedness of pyramidal cells or anatomical features such as the cortical layer of origin of the neurons. GRAPHDINO performed on par or better than previous data-driven approaches on these tasks. Taken together, we found that our learned morphological embeddings are able to capture the essence of intricate 3D morphologies of neurons and are well suited as representations of neuronal morphologies for further downstream analysis.

## Discussion

Our results demonstrated that compact representations of neuronal morphologies can be learned in a data-driven way even on comparatively small morphological datasets.

While we hypothesize that our novel AC-ATTENTION mechanism can be beneficial in various graph learning settings outside of the neuroscientific domain, this remains to be empirically validated. Preliminary experiments on a botanical tree dataset indicated that GRAPHDINO is useful in other domains where individual samples are graphs and node features are spatially embedded. For datasets with a more pronounced domain shift, it will likely be necessary to adapt the augmentation strategies to fit the respective data domain. Likewise, the model is not tied to the unsupervised learning objective, but can equally be applied in a supervised setting if the necessary ground truth annotations are available.

The dataset sizes in this study are rather small for self-supervised learning as they only entail a few hundred to a thousand samples. While the well-aligned inductive biases of the model allow us to learn useful features without overfitting on the training data, more data would presumably lead to better and more robust representations. We explore this in Section 3.3.

Our results confirm that learning a representation of neuronal morphologies is possible in a data-driven fashion. This opens up avenues for the discovery of new cell types as the model learns which features are relevant to distinguish between the individual cells without being restricted to predefined features. As the focus of this study was to demonstrate our ability to replicate known findings in an unsupervised way and reproduce cell types that are already established in the literature, an in-depth biological analysis of the identified cell types was beyond the scope of this work. However, the next section focuses on exactly that: We use our model to embed a novel dataset of unprecedented size in a low-dimensional representation space and analyze the cortical organization of excitatory neurons in the mouse visual cortex.

### 3.3 Discovery of excitatory morphological cell types

*This section summarizes:*

Marissa A. Weis, Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G. Fahey, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, R. Clay Reid, Casey M. Schneider-Mizell, H. Sebastian Seung, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Jacob Reimer, Andreas S. Tolias, and Alexander S. Ecker. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex. *bioRxiv*, 2023.

*The full publication can be found in the appendix on page 163.*

#### Motivation

Since Ramón y Cajal, neuronal morphologies have been used to classify neurons into cell types (Ramón y Cajal, 1911). However, whether there exist distinct morphological types of cortical excitatory neurons or if instead they form a continuum of variation is still an open research question.

Previous studies that characterized neuronal morphologies have two main limitations. First, they often consider only a low number of neurons (Zeng and Sanes, 2017), the selection of which is biased by which neurons can be successfully reconstructed. Second, they rely on hand-picked features that might not capture all axes of variations of the morphologies. In Section 3.2, we introduced GRAPHDINO. GRAPHDINO embeds neuronal skeletons into morphological embeddings in a data-driven way. While the previous publication was a proof of concept that it is possible to learn a low-dimensional embedding of highly complex neuronal morphologies that can be used to reproduce known cell types and organization principles (see Section 3.2), in this study we applied GRAPHDINO to a novel neuronal dataset of unprecedented size to discover cortical organization principles in the mouse visual system.

Over the course of five years, the MICrONS project densely imaged and reconstructed all neurons and their connectivity in a millimeter cube of one individual mouse brain, more specifically of the visual cortex of the mouse (MICrONS Consortium et al., 2023). By densely imaging and reconstructing all cells in the volume, the data allows for a complete census of neurons in this brain area, as it is not prone to selection bias based on which cells were sampled for imaging. While the MICrONS data is an extremely valuable source for the research of cortical organization, it is also very challenging to analyze. The raw data encompasses over one petabyte of image data (MICrONS Consortium et al., 2023). The scale of the dataset makes it challenging to analyze with traditional methods. But it is the perfect setting for unsupervised machine learning, which has been shown to excel in high-data regimes. We therefore use GRAPHDINO to extract low-dimensional embeddings of more than 50,000 neurons in the mouse visual cortex from areas V1, RL and AL and analyze the differences in cortical organization between layers and visual areas.



## Results

We used GRAPHDINO to extract morphological embeddings of approximately 55,000 neurons and neuronal fragments. We found that our learned morphological embeddings can be used for quality control of the MICrONS dataset (MICrONS Consortium et al., 2023), enabling us to separate neuronal fragments from full neurons and distinguishing excitatory from inhibitory neurons. Our subsequent analysis was focused on the excitatory neurons. We demonstrated that the morphological embeddings learn to encode the soma depth of the neurons, even though the soma location was not provided to the model. We therefore used the morphological embeddings to predict the layer of origin of the cells based on a small annotated subset of the neurons. As a proof of concept that the learned embeddings are indeed an expressive representation of the neuronal morphologies, we related our embeddings to a subset of expert-annotated cells and showed good correspondence to previously described excitatory cell types.

We proceeded with a layer-wise analysis, investigating the main axis of variation of the dendritic morphologies of each layer. We found that excitatory neurons in layers II/III and IV form a continuum of morphological variation. Neurons in layer II/III showed a gradual decrease in the width of their apical dendritic arbor as well as smaller apical tufts with increasing cortical depth. For layer IV, we found a strong area difference between the primary visual cortex and higher visual areas. Namely, atufted L4 neurons were primarily located in the primary visual cortex, while L4 neurons with apical tufts were more frequent in higher visual areas. Layer V showed a less continuous variation of neuronal morphologies. We found that L5-ET cells form a rather distinct cluster from the rest of the L5 cells, while L5-NP cells were more the end of a continuous spectrum. Layer VI exhibited the most diversity in terms of morphological shapes featuring inverse and horizontal cells as well as narrow pyramidal cells with a spectrum of apical termination points in different higher layers.

Lastly, we identified a new morphological feature, the “basal bias”. While the basal dendrites of most neurons accumulated more mass below the soma, we found one group of neurons on the border of layer IV and V that exhibited a striking pattern of basal dendrites. They tended to curve upwards with respect to the soma and avoided reaching into layer V. These neurons were mainly located in the primary visual cortex as opposed to higher visual areas and were mostly atufted.

Taken together, the studied population of excitatory neurons from the mouse visual cortex showed significant variation of dendritic morphologies, both within and across cortical layers. This variation mostly formed a continuum, with only a few distinct cell clusters in deeper layers.

## Discussion

In summary, our learned morphological embeddings enabled us to characterize the cortical organization of neuronal morphologies in the mouse visual cortex, both corroborating known features as well as making novel observations.

First, we found that atufted L4 neurons are mostly present in the primary visual cortex, while L4 neurons in the higher visual areas were more tufted. One possible explanation for this pattern is that the primary visual cortex as the first stage in cortical visual processing and layer IV as the input layer might be less modulated by feedback connections, which are often sampled through apical tufts reaching into layer I (Larkum, 2013; Fişek et al., 2023). Second, we found a novel L4 cell type that is characterized by a bias of the basal dendrites to avoid reaching into layer V that primarily occurs in V1. Avoiding layer V and therefore input from L5 neurons could be an additional mechanism to focus on the thalamic input and the forward drive of the microcircuit as L5 neurons

sample feedback heavily (Kim et al., 2015). Further analyses are required that take connectivity patterns and functional properties of the neurons into account to explore these hypotheses.

In general, we found that the morphologies of excitatory neurons mostly display continuous variation rather than forming discrete clusters of cell types. However, this does not prove that there are no discrete neuronal cell types. While this study suggests that the global structure of excitatory dendritic morphologies follows a continuum, it is possible that if other features or modalities are considered, the cells might indeed form distinct clusters (for further discussion see Section 3.4).

An alternative explanation for the continuous variation that needs further study is the high level of noise in the data that could lead to a more continuous appearance. The reconstruction of neuronal morphologies from electron microscopy images involves a multi-step pipeline over which noise can accumulate (MICrONS Consortium et al., 2023). The reconstruction of the dendritic trees might fail at random positions, which could lead to a more continuous appearance of the dendritic morphologies across the dataset. While we were able to remove large fractions of fragmented cells based on the morphological embeddings, further analyses and more extensive manual proof-reading are required to fully eliminate this possibility.

Going forward, an important step is to quantify the notion of discrete versus continuous variation in the data. We show qualitative results that indicate a continuous variation. But a statistical measure or hypothesis test is missing that can quantify the probability of the underlying data lying on a continuum as opposed to originating from discrete clusters. Here more theoretical work is needed to model discrete versus continuous variation in a principled way.

Summarizing, we characterized multiple differences in neuronal morphologies between visual areas and layers as well as identified a novel cell type. However, while a descriptive study is the prerequisite, it is only the first step towards understanding cortical organization. The next step is to analyze why these differences exist and how they relate to the functional properties of the neurons as well as their role in the microcircuits of the brain.

### 3.4 Limitations and future directions

Sections 3.2 and 3.3 summarize our two studies that introduced and applied the self-supervised graph learning model, GRAPHDINO, to neuronal datasets to learn low-dimensional descriptors of neuronal morphologies. However, GRAPHDINO is only the first step in analyzing neuronal morphologies in a data-driven way. In the following section, we discuss some notable limitations and future research directions for modeling neuronal morphologies.

**Morphological features** GRAPHDINO operates on graphs that represent the skeletons of neurons. This representation abstracts away many of the more fine-grained details of neuronal morphology such as the width of neuronal segments or the presence of spines. The model thus only learns to encode the global structure of the neuronal morphology. This facilitates the optimization process and prevents the model from overfitting on those details. However, the omitted fine-scale structure of neurons can be predictive of cell types (Dorkenwald et al., 2022; Elabbady et al., 2022). For instance, previous studies found variation in the density and size of synapses between neuronal types (Schneider-Mizell et al., 2023; Celii et al., 2023). Others used soma shape and fine-grained soma features to differentiate between cell types (Ascoli et al., 2008; Elabbady et al., 2022). Integrating additional morphological information into our model, especially across multiple scales, represents a promising future research avenue. There are multiple ways this could be achieved: First, specific properties such as the number of spines could be extracted from the reconstructed neurons prior to skeletonization (Celii et al., 2023) and integrated as additional features of the graph nodes. In this case, augmentations on these additional node features are required to avoid creating a short-cut by which the model can trivially learn to differentiate between individual cells. Alternatively, instead of manually selecting certain fine-scale details such as the number of spines and using them as node features, local segments of the reconstructed cells can be encoded into learned representations prior to skeletonization and adopted as the initial node features for GRAPHDINO. By doing so, data-driven features across multiple scales of the morphology can be learned. Dorkenwald et al. (2022) proposed a model to encode local segments using a convolutional encoder and a contrastive learning objective. A similar approach could be used to obtain representations of the local morphology of neurons. Instead of averaging over the local encodings to extract a global cell embedding, as was done by Dorkenwald et al. (2022), the local representations could then be aggregated using GRAPHDINO, obtaining a hierarchical model of neuronal morphologies across different scales.

Due to insufficient reconstruction quality in the data, we removed axons prior to learning the morphological embeddings. However, axonal projections have long been used to characterize excitatory cells, differentiating them broadly into intratelencephalic (IT), pyramidal tract (PT) and corticothalamic (CT) neurons based on the regions that their axons target (Harris and Shepherd, 2015). Reconstructing axons of excitatory cortical neurons is challenging as they can span the entire brain and therefore need imaging of a large volume of the brain to be fully captured (Zeng and Sanes, 2017; Peng et al., 2021). Additionally, they are difficult to trace due to their long and thin branches. When reliable axon reconstructions for a sufficient number of neurons become available, GRAPHDINO can be trained on these without any alterations. The model itself is not tuned to dendritic morphologies, but currently restricted by the availability of data.

Integrating further features into our data-driven representation to account for local morphological properties as well as axonal projections might change the interpretation of a continuous variation across dendritic arbors and lead to a more distinct clustering of excitatory cortical cell types. Similarly, integrating features from other modalities such as transcriptomic or electrophysiological properties might lead to a more distinct separation (see discussion below).

**Inductive biases** As discussed in Chapter 1, every learning algorithm has its implicit and explicit inductive biases. While these are necessary for learning and generalization, they also strongly influence the learned representations. While we eliminated biases through human-defined features, GRAPHDINO comes with biases on its own. We chose a graph-based input representation as opposed to a voxel or point-cloud representation. This fits the data, as neuronal morphologies are rather sparse with long-ranging branches, which makes using voxels computationally inefficient, whereas point clouds would disregard all information contained in the connectivity. Both GNNs and transformers are not well suited to process neuronal morphologies on their own. GNNs based on message passing cannot handle aggregating information over the long-ranging branches without becoming prohibitively deep (Dwivedi et al., 2022). Furthermore, they tend to over-smooth the node features with increasing depth of the GNN (Rusch et al., 2023). Transformers applied to graphs on the other hand can capture long-range dependencies as they leverage full-connections between all nodes, but they cannot use the information encoded in the graph connectivity as information flow between nodes is solely based on feature similarity. Thus, we chose to create our own architecture with a novel attention mechanism that interpolates between transformer attention and graph message passing, to allow for information exchange between distant nodes while simultaneously being able to use the information given by the graph connectivity.

Furthermore, we designed graph augmentations that are aligned with biological appropriate invariances and that take advantage of the spatial meaning of node features. But our proposed graph augmentations encode specific invariances in the learned representations which might not be optimal for all downstream tasks. For example, we use rotation around the axis orthogonal to the pia as a data augmentation leading to rotation-invariant embeddings, as we hypothesize that rotation is not a discriminating property for cell types. However, Weiler et al. (2022) found a correlation between the orientation tuning of L2/3 cells and the maximal horizontal extent of their apical dendrites, suggesting that the asymmetries in the dendritic arbors fulfil a functional role. To study such phenomena, the proposed augmentations need to be adapted to fit the underlying research question.

One inductive bias that is missing in GRAPHDINO, but which might be helpful to model the hierarchical structure of neurons is compositionality. Compositionality as an inductive bias is crucial for object-centric learning, but has so far been neglected for neuronal morphologies. However, just as scenes consist of different objects which in turn are composed of object parts, the brain consists of individual cells that are made of different compartments. While compositionality of neurons is not as well understood as compositionality of visual scenes, leveraging compositional modeling could still offer similar advantages. As we argued above, learning a hierarchical representation of neuronal morphologies could improve our modeling capabilities and leveraging compositionality in this context might be advantageous (for further discussion see Chapter 4).

**Interpretability of embeddings** While our results demonstrate that the learned embeddings capture morphologies of neurons well and can be used for diverse downstream tasks, one downside of using learned features is their lack of interpretability. This is in contrast to classical morphometrics and a clear limitation of the approach. While the features are more flexible than manually selected features, additional steps are required in order to make them human interpretable. For instance, clusters identified based on the learned embeddings need to be analyzed by experts or with the help of known morphometrics to determine their differences. While this certainly requires effort, examining tens of clusters is feasible while analyzing thousands of neurons individually is impossible. Furthermore, by doing so we identified a novel morphometric that is a characteristic feature of neurons at the layer IV – V boundary. Going forward, this feature can now be included in automatic analysis pipelines.

**Are there discrete cell types of excitatory cortical neurons?** The mammalian brain consists of 26 million to 86 billion neurons depending on the species (Herculano-Houzel et al., 2015). These neurons show an extremely high variability in their shapes and functions. To permit studying the brain, neurons have been discretized into cell types to reduce the complexity of analyzing their organization and function (Zeng, 2022). Despite the long history of cell typing, there is no universal definition of how to delineate neuronal cell types (Masland, 2004; Fishell and Heintz, 2013; Zeng and Sanes, 2017; Mukamel and Ngai, 2019; Zeng, 2022). While there exists a consensus that neurons can be broadly divided into excitatory, spiny neurons and inhibitory, aspiny neurons (DeFelipe et al., 2002; Zeng and Sanes, 2017), the exact definition and number of more granular cell types is often disputed (Kanari et al., 2019). Previous studies used disparate sets of features across multiple modalities to define cell types. There have been efforts of unifying cell typing schemes (Ascoli et al., 2008), but finding a universal definition is complicated by the fact that cell properties are heterogeneous between species, brain areas and individual cells and classifications schemes based on different modalities do not necessarily align (Zeng, 2022).

Neurons have traditionally been categorized into cell types based on their morphology. Following this approach in Section 3.3, we analyzed the neuronal morphologies of a densely imaged volume in the mouse visual cortex containing tens of thousands of neurons. Our results suggest that the dendritic morphologies of excitatory cortical neurons in the mouse visual cortex mostly form a continuum, with some notable exceptions of distinct classes such as L5-ET neurons. In contrast, previous studies examining dendritic morphology of cortical neurons usually defined distinct cell types and categorized the neurons accordingly (Oberlaender et al., 2011; Markram et al., 2015; Gouwens et al., 2019; Kanari et al., 2019). While there is reasonable agreement on broad cell classes in these studies (see Section 3.1), the number of defined cell types differs from study to study. In rodents, estimates range between nine and nineteen morphological types of excitatory neurons in cortical sensory areas (Oberlaender et al., 2011; Markram et al., 2015; Narayanan et al., 2017; Kanari et al., 2019; Gouwens et al., 2019; Schneider-Mizell et al., 2023). These discrepancies can arise due to a number of reasons: Studies differ in recording techniques as well as classification methods, sample sizes vary and considered morphological features are disparate.

Sample size and selection bias of samples can influence classification outcomes. The lower sample sizes of previous studies could have contributed to the impressions of discrete cell types. When comparing two neurons of different ends of a continuous spectrum, they appear distinct. By densely reconstructing neurons, the continuous variation becomes observable. Therefore, only the more recent large-scale datasets that aim at densely recording whole brain regions are suitable to comprehensively study this question of continuous versus discrete variation as they do not suffer from selection biases (MICrONS Consortium et al., 2023; BRAIN Initiative Cell Census Network, 2021).

Second, most studies employ clustering methods on the respective morphological features to discover cell types (Oberlaender et al., 2011; Marx and Feldmeyer, 2012; Narayanan et al., 2017; Gouwens et al., 2019). However, clustering inherently assumes discrete groups, hence encouraging the notion of distinct cell types. While previous studies usually report distinct classes, they also note variability within their proposed classes and further (continuous) variability between classes can be observed in their visualizations of morphometrical properties (Gouwens et al., 2019; Kanari et al., 2019; Schneider-Mizell et al., 2023).

While morphology is probably the oldest (Ramón y Cajal, 1911), it is only one of multiple modalities that can be considered for cell typing (Zeng, 2022).<sup>9</sup> Nowadays, transcriptomics in the form of RNA-

---

<sup>9</sup>Other modalities that have been used for cell typing include electrophysiological properties, molecular features such as transcriptomics, and the connectivity of neurons (see Section 3.1).

sequencing is widely used to generate cell type taxonomies due to its scalability both in the number of expressed genes as well as the number of cells that can be recorded (BRAIN Initiative Cell Census Network, 2021; Yao et al., 2021; Zeng, 2022; Yao et al., 2023). Noteworthy, similar to our findings, current transcriptomics studies observed continuous variation between types of cortical (Tasic et al., 2016, 2018; Gouwens et al., 2020; Scala et al., 2021; Yao et al., 2021) and subcortical neurons (Stanley et al., 2019; Wang and Lefebvre, 2022). Furthermore, they demonstrated that variation within transcriptomic types aligns with variation in other modalities (Gouwens et al., 2020; Scala et al., 2021). For instance, the excitatory IT type as defined by transcriptomics has been shown to exhibit continuous variation over cortical depth (Scala et al., 2021; Yao et al., 2021), analogous to the gradual change of dendritic morphologies that we found along increasing soma depth.

Together recent studies in morphology and transcriptomics suggest that there are a few broad families of cortical cell types with continuous variation within those classes (Gouwens et al., 2020; Scala et al., 2021; Yao et al., 2021; Weis et al., 2023). This continuous variation makes it difficult to discretize the cell types further and puts into question whether a comprehensive catalogue of cell types can be defined. However, so far a principled statistical analysis of how to characterize discrete versus continuous variation in the data, especially in a hierarchical setting, is lacking. Thus, theoretical work on how to model these is needed.

Going forward, to answer the question how to best characterize the hierarchy of (cortical) neuron classes and how to integrate meaningful continuous variation into that schema, a joint classification of (1) comprehensive properties of (2) multiple modalities (3) in densely recorded brain volumes is needed. While different modalities have been found to co-vary to a certain degree, using multiple of them would give a more complete characterization of the neurons and enable a more holistic cell typing schema (Zeng and Sanes, 2017). Additional modalities might differentiate classes that are not separable in one modality. For instance, a distinct projection pattern separating L2/3- from L5-IT neurons has been described (Peng et al., 2021). Furthermore, considered features within a modality need to be reasonably complete. For morphological characterization, the common approach of defining a fixed set of morphometrics might not capture all factors of variation of the morphologies. A comprehensive set of morphological features across scales needs to be taken into account. Hence, building on top of our work and advancing representation learning for neuronal morphologies might offer new insights (see Section 3.2). Last, large scale datasets that densely record neurons without selection biases and preferably including multiple modalities are needed. Datasets such as MICrONS (MICrONS Consortium et al., 2023) and BICCN (BRAIN Initiative Cell Census Network, 2021) are excellent starting points to further investigate multi-modal variation of cortical neurons.

## 4 Discussion

In this dissertation, we studied two different applications of unsupervised representation learning and how they are influenced by inductive biases. We saw that too strong inductive biases in machine learning models can hinder learning and progress towards more general applications (see Chapter 2). On the other hand, using inductive biases to encode domain knowledge into machine learning models can lead to sample efficient models and new scientific discoveries in that domain (see Chapter 3). For a detailed discussion of the results of the individual publications, we refer to Sections 2.2, 3.2 and 3.3 as well as the publications themselves. In the following chapter, we discuss some overarching themes from a broader perspective.

**Evaluation of unsupervised representations** Evaluating unsupervised representations is challenging. As discussed in Section 1.1, representation learning is not a clearly defined task. What constitutes a good representation is dependent on the subsequent use case of the representation. Often representation learning aims at learning a generally good representation that can be used for several downstream tasks which might not be defined or anticipated at the time of model training (Oquab et al., 2023). Consequently, it is not straightforward to design a quantitative measure that conclusively evaluates the representation. Nevertheless, there are several strategies to assess the representational quality: (1) Evaluation can be done qualitatively by for example using visualization techniques. Alternatively, for a quantitative evaluation, the representation can be assessed (2) using metrics that do not rely on ground truth labels such as the unsupervised loss on held-out data or pretext tasks based on pseudo labels or (3) by evaluating (proxy) tasks based on labeled test data. The latter is beneficial as it enables quantitative evaluation of the model but alleviates the annotation costs compared to labeling a whole training set. However, this is only possible if the downstream task is a priori known and the data can be annotated with reasonable effort. Since oftentimes the goal of unsupervised representation learning is to obtain a representation that is useful for several downstream tasks, representational quality is commonly judged based on one or a selected subset of proxy tasks. For example, the quality of self-supervised vision models is often evaluated by training a linear classifier on the ImageNet dataset (Russakovsky et al., 2015) on top of the learned representations and assessing its object classification accuracy (Jing and Tian, 2019). However, evaluating a representation based on proxy tasks can only give an incomplete picture as it is unclear how the representation would perform for others tasks (Bengio et al., 2013).

In this dissertation, we evaluated unsupervised representation learning models in two different application domains. Both come with their own challenges depending on the underlying data and the aim of the project and hence the evaluation strategies differ significantly.

In case of the object-centric models (see Chapter 2), the objective of the models is clearly defined, namely to decompose visual scenes into their respective objects, but it is difficult to specify a simple rule of what constitutes an object. Hence, object-centric learning tries to recover the underlying structure of the scene without supervision, commonly using reconstruction as a pretext task for training. For evaluation, we can exploit that object-centric representation learning is tightly linked to other computer vision tasks such as tracking and instance segmentation for which numerous datasets with annotations exist (Lin et al., 2014; Cordts et al., 2016; Milan et al., 2016; Dendorfer et al., 2020). Annotations can be generated with comparably little effort as untrained workers are able to perform the labeling. Furthermore, there are good generative models in the form of graphic engines to generate synthetic data with ground truth labels that can be used for evaluation (Greff et al., 2022; Raistrick et al., 2023). Hence, we can leverage segmentation and tracking as proxy tasks

to quantitatively evaluate object-centric models (Weis et al., 2021; Karazija et al., 2021; Greff et al., 2022; Kipf et al., 2022; Zadaianchuk et al., 2023). However, as discussed in Section 2.3, these proxy tasks only evaluate certain aspects of the object representations. In our benchmark, we therefore additionally evaluated the models on out-of-distribution test sets that featured specific, challenging scenarios of visual scenes to get a better understanding of the models' performances.

The most challenging scenario for evaluating unsupervised representations is when the ground truth is not a priori known and the representations are meant to help with the discovery of new knowledge. This was the case for our study of excitatory morphologies in the mouse visual cortex (see Section 3.3). In contrast to segmentation in computer vision, generating test data to evaluate models of neuronal morphologies is much more challenging. There does not yet exist a good generative model of neuronal morphologies to generate synthetic data and labeling of existing data can only be done by trained experts. Even if experts are available to annotate data, whether and which morphological cell types of excitatory cortical neurons exist is an open research question. Hence, we cannot easily generate datasets with ground truth annotations for quantitative evaluation. Consequently, we resorted to two strategies for evaluation: (1) Relating the representations to already known features of the data based on existing knowledge in neuroscience such as previously described cell types and morphometrics and (2) qualitatively evaluating the representations using visualization techniques such as t-SNE (van der Maaten and Hinton, 2008) and manually inspecting the results of cluster analyses. First, while there is no consensus on the identity of morphological cell types of excitatory cortical neurons, there are patterns in the morphologies that have consistently been described in previous literature. We can leverage these to show that our learned embeddings reflect these patterns. To do so, expert annotations are needed, which was done for a small subset of the data in the case of the MICrONS dataset (Schneider-Mizell et al., 2023). Furthermore, we can relate the embeddings to anatomical features such as the soma depth that can be automatically extracted from the raw data, showing that the embeddings capture meaningful variation in the data. Second, extensive manual inspection of the learned embeddings was done. To make this feasible, we performed clustering of the learned embeddings and manually analyzed the found clusters. Furthermore, we used t-SNE and traversals in latent space to visualize patterns in the embeddings. However, in the end, machine learning models for scientific discovery can only generate hypotheses about the underlying domain. Hence, the ultimate evaluation needs to be done by experimentally testing and validating the proposed hypotheses. In the case of neuronal morphologies, the next step would be to relate the found organization principles of the morphologies to other modalities to determine whether for instance the found differences between visual areas are reflected in the connectivity as well as the functional properties of the neurons.

In summary, evaluating unsupervised representations is challenging and requires tailored strategies depending on the task and data. The important aspects of object-centric representations, namely the successful discovery of objects, can be evaluated using the numerous datasets for segmentation available in computer vision. While the limitation remains that other aspects of the object-centric representations are neglected in the evaluation, it gives us a good understanding of the strengths and weaknesses of the models. In contrast, evaluating models that are meant to generate new knowledge in scientific fields are more difficult to evaluate as data is expensive to generate and labeling is often impossible since the underlying ground truth is not known. Thus, evaluating unsupervised representations in these domains requires a lot of manual work and a cycle of repeated modeling and experimental validation of the hypotheses generated by the models.



**Compositionality** Compositionality postulates that complex structures can be constructed from simpler building blocks using a set of compositional rules (Schulz et al., 2017). Humans use compositionality for many areas of cognition such as language (Chomsky, 1965) or visual perception (Lake et al., 2015). Compositionality is linked to systematic generalization by decomposing knowledge into independent parts that can be recombined flexibly (Bahdanau et al., 2019; Ruis et al., 2020; Goyal and Bengio, 2022). Systematic generalization here denotes that “the meaning for a novel composition of existing concepts can be derived systematically from the meaning of the composed concepts” (Goyal and Bengio, 2022). Equipping models with a notion of compositionality is advantageous as it enables models to generalize to novel configurations of learned concepts, which leads to better sample efficiency (Diuk et al., 2008; Lake et al., 2015). Furthermore, compositionality is thought to help with continual learning as new concepts can be learned without overwriting old ones (Li et al., 2020; Mendez and Eaton, 2021, 2023). However, so far deep learning models struggle with capturing and utilizing compositionality in the data (Lake and Baroni, 2018; Loula et al., 2018; Keysers et al., 2020; Bogin et al., 2021; Ma et al., 2023; Wiedemer et al., 2023).

Compositionality is relevant in both data domains that we considered in this dissertation: Visual scenes are composed of individual objects which in turn are made of individual parts. Likewise, the brain is a composition of cells which in turn consist of different morphological compartments.

In Chapter 2, we analyzed object-centric models that explicitly build compositionality into their architectures to perform unsupervised object discovery inspired by human object perception. However, current object-centric models hard-code the modularity and only encode one layer of abstraction with commonly a fixed number of objects (Burgess et al., 2019; Locatello et al., 2019; Greff et al., 2019; Kipf et al., 2022; Seitzer et al., 2023). But the visual world is more complex and exhibits numerous hierarchical levels of compositionality: Scenes consists of objects, which are composed of parts which can often be broken down even further. Hence, going forward it would be desirable to design models that can more flexible route information on multiple hierarchical levels and for different numbers of objects to take full advantage of the compositionality of the visual world (Hinton, 2021).

In Chapter 3, we proposed a model that learns representations of neuronal morphologies. Neurons are complex structures which can be decomposed into different neuronal compartments, such as the soma, dendrites and the axon (Kandel et al., 2000). Recursively, the different compartments can be subdivided into smaller segments. However, compositionality of neurons is not as well understood as for visual scenes. Dendritic trees of neurons might exhibit recurring branching patterns or motifs that are not easily visible to humans. So far, compositionality has not been taken into account when modeling neuronal morphologies. Going forward building compositionality into the neuronal representations by building them up hierarchically would be an interesting future research direction (see discussion in Section 3.4).

Thus, while the two data domains are different in many regards, they are both hierarchically structured. But, while object-centric models already use compositionality as an inductive bias, neuronal morphology models currently do not. This is due to the fact that less work in general has been done in modeling neuronal morphologies using machine learning and we have a much better understanding of what constitutes the individual parts in visual scenes. With GraphDINO we now have a capable model for the global dendritic morphology of neurons that in the future can be adapted to exploit compositionality. Going forward, both domains could profit from models that can utilize systematic compositionality. However, research on leveraging compositionality in deep learning models is still in its infancy and further work is needed to find the correct inductive biases to build models that can take advantage of compositionality over abstract concepts (Smolensky et al., 2022).

## 5 Outlook

**Are strong inductive biases still necessary?** In last years, the availability of data and compute resources for machine learning has massively increased. There has been a rapid progress in the development of increasingly powerful and specialized hardware for machine learning. Compute resources used for training machine learning models have doubled every six months between 2010 and 2022 (Sevilla et al., 2022). Similarly, dataset sizes, especially those used for training large language models and large computer vision models, have surged (Schuhmann et al., 2022; Common Crawl; Kirillov et al., 2023). The increased availability of data and compute resources has changed the way new machine learning methods are developed. It poses the question if is still necessary or desirable to build prior knowledge and structure into machine learning systems in the form of inductive biases or whether the right way forward is to have mostly unconstrained methods that learn from the massive amounts of data everything they need to know about the world from scratch (LeCun and Manning, 2018; Sutton, 2019; Welling, 2019).

Looking back at artificial intelligence history, general approaches that leveraged large amounts of computation and data often outperformed models that were based on human-knowledge systems (Sutton, 2019). Chess was solved by performing deep search over the action space (Campbell et al., 2002). Similarly, Go was solved using search strategies in combination with reinforcement learning without explicitly encoding prior knowledge about the game into the model (Silver et al., 2017).<sup>10</sup> Current large language models like GPT-4 show remarkable text-generating capabilities without imposing formalized linguistic knowledge on the systems (OpenAI, 2023). Scaling data and compute resources and solving the engineering challenges that accompany training models at scale has brought machine learning systems a long way in solving a wide array of tasks. However, today's models are still far away from common sense reasoning and general intelligence. Consequently, the question on how to best spend research resources, on finding the right structure or focusing on scalable methods, is still highly relevant.

As discussed in Chapter 1, every model contains explicit and implicit inductive biases and they are essential in the classical machine learning setting in which we try to learn general rules from finite datasets. There is no generalization without them and they are especially important in the low data regime. However, there is a trade-off between how much structure needs to be set a priori depending on how much data is available. With increasing dataset sizes and the creation of so called foundation models (Bommasani et al., 2022), picking narrow and specific inductive biases becomes less important. Instead of constraining the hypothesis space with inductive biases, more data can be used to learn the rules from the data directly (Tolstikhin et al., 2021; Bachmann et al., 2023).

Following the recent trends in machine learning, one could conclude that researching the right inductive biases for a task has become secondary and the more promising avenue is to invest in scaling model size, compute resources and dataset sizes further. But equipping machine learning models with appropriate priors has several advantages: (1) There will always be data domains in which we cannot collect enough data to train large-scale models from scratch. As long as the correct hypothesis is included, a constrained hypothesis space leads to more efficient learning. (2) The goal of science is to generate knowledge about the world. Creating large, well-performing models does not necessarily further our understanding about the underlying data distribution. The objective

---

<sup>10</sup>While the first version of ALPHAGo still relied on learning from human game play (Silver et al., 2016), the successor model ALPHAGo ZERO was trained without human knowledge (Silver et al., 2017). However, others have argued that the design choices of ALPHAGo ZERO encode prior knowledge and cannot be seen as *tabular rasa* learning as claimed by the authors (Marcus, 2018).

should rather be to find simple, constrained models that explain a phenomenon well. (3) Related to that, training general models purely from data gives us less knowledge and control over what they learn. Building in inductive biases or analyzing the ones of trained models helps model explainability and interpretability.

In domains such as natural language processing (NLP) and computer vision, nowadays large datasets are available for numerous tasks and ample data can be collected at comparably little cost by crawling the internet. Consequently, those are the domains with substantial progress in machine learning in the last decade. However, this is not true for all data domains where machine learning could be of use. There are many domains with limited available data and in which more data cannot easily be generated. Even for domains with abundant data such as computer vision or NLP, data is restricted to specific subdomains and tasks. For example, deep learning models exhibit extraordinary performance in object classification for a fixed set of object classes. But the distributions of all object classes in the world follows a long-tail distribution (Zhu et al., 2014). For rare classes very little data is available. Similarly, large language models work incredibly well for languages such as English which are well represented in the machine learning community and their datasets. However, this is not the case for a large fraction of the world's languages (Joshi et al., 2020; Bommasani et al., 2022). There are numerous other domains for which data cannot easily be generated such as medical data or experimental data for scientific research. Neuronal morphologies is one such domain (see Chapter 3). The MICrONS dataset has an immense sample size for neuronal morphologies (MICrONS Consortium et al., 2023), but it is far smaller than current datasets in computer vision or NLP. Generating experimental data is incredibly time consuming and costly. Hence, data availability in this realm will always be limited and researching the appropriate inductive biases to leverage machine learning for data analysis in these domains is required.

Second, science aims to understand how the natural world works. In order to do so, scientists build models that are simpler representations of a system or natural phenomenon in order to understand its inner workings. If we look at machine learning models from the perspective of using them as tools to generate scientific insights, creating large, general models with high accuracy on a certain task does not necessarily further our understanding of the underlying data distribution. In contrast, if we can find the right assumptions to train a simple model on limited data, the incorporated inductive biases can give us insights into the underlying phenomena, since successful generalization implies that the assumptions and the underlying problem are well aligned. Reversely, domain knowledge can help us to build simpler and more efficient machine learning models in that realm (see Chapter 3).

Third, machine learning models are more and more influencing our everyday life as part of many applications. Training general models from large amounts of data gives us little control over which decision functions they learn and makes it difficult to understand which features were crucial for a given prediction. However, especially in safety-critical applications it is imperative to know how and when models work and even more importantly when they fail. Consciously equipping models with inductive biases or analyzing the ones of trained models can help to understand their strengths and weaknesses and therefore enhances model explainability and interpretability.

Last, progress in machine learning is usually measured as gain in performance. However, better performance is not the only objective that should be taken into account when developing and deploying machine learning models. We also need to consider the consequences and effects on a broader societal level. We are in the midst of a human-induced climate change. Energy production is (currently) not an unlimited resource and consequently we need to use it efficiently. Training large models from scratch is problematic since it is very energy intensive (Strubell et al., 2019; Bender et al., 2021). Additionally, while inference is usually cheap in comparison, when done in large models

millionfold per day the required energy amount is substantial (Desislavov et al., 2023). Moreover, data has to be collected or generated and is not unbiased. If everything is learned from data, we have to ensure that the models do not replicate the unfavourable or discriminating biases in the data or even enhance them (Bender et al., 2021). Additionally, already today there is a concerning development of an increasing number of underpaid workers, mostly in the Global South, to generate, annotate and filter the massive amounts of data that are needed to feed the data-hungry models. Depending on the data domain, there are also privacy issues to consider (Papernot et al., 2017). Taken together, the performance of machine learning models should not be the only criterion in their development, but societal consequences need to be considered as well.

All in all, finding machine learning algorithms with better inductive biases that allow us to learn efficiently with less energy consumption, from less data and that ensure not to blindly replicate biases in the data is highly desirable. Furthermore, in the context of using machine learning for scientific discovery the goal should be to find simple, constrained models that distill knowledge about the world. Therefore, investing in researching the right inductive biases and finding the optimal trade-off between innate structure and learning is still highly relevant.

# Bibliography

- Amir, Shir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. *arXiv*, 2112.05814, 2022.
- Ascoli, Giorgio A., Lidia Alonso-Nanclares, Stewart A. Anderson, Germán Barrionuevo, Ruth Benavides-Piccione, Andreas Burkhalter, Gyorgy Buzsáki, Bruno Cauli, Javier DeFelipe, Alfonso Fairen, Dirk Feldmeyer, Gord Fishell, Yves Fregnac, Tamas F. Freund, Daniel Gardner, Esther P. Gardner, Jesse H. Goldberg, Moritz Helmstaedter, Shaul Hestrin, Fuyuki Karube, Zoltán F. Kisvárdy, Bertrand Lambolez, David A. Lewis, Oscar Marin, Henry Markram, Alberto Muñoz, Adam Packer, Carl C. H. Petersen, Kathleen S. Rockland, Jean Rossier, Bernardo Rudy, Peter Somogyi, Jochen F. Staiger, Gabor Tamas, Alex M. Thomson, Maria Toledo-Rodriguez, Yun Wang, David C. West, and Rafael Yuste. Petilla terminology: nomenclature of features of gabaergic interneurons of the cerebral cortex. *Nature Reviews Neuroscience*, 9:557–568, 2008. doi: 10.1038/nrn2402.
- Bachmann, Gregor, Sotiris Anagnostidis, and Thomas Hofmann. Scaling MLPs: A tale of inductive bias. *arXiv*, 2306.13575, 2023.
- Bahdanau, Dzmitry, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Baillargeon, Renee. Representing the existence and the location of hidden objects: Object permanence in 6- and 8-month-old infants. *Cognition*, 23(1):21–41, 1986. doi: 10.1016/0010-0277(86)90052-1.
- Bao, Zhipeng, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- Bao, Zhipeng, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Object discovery from motion-guided tokens. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Battaglia, Peter W., Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çağlar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv*, 1806.01261, 2018.
- Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proc. of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021. doi: 10.1145/3442188.3445922.
- Bengio, Yoshua, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.
- Bernardin, Keni and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008. doi: 10.1155/2008/246309.

- Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, New York, NY, 2006. ISBN 978-0-387-31073-2.
- Bogin, Ben, Shivanshu Gupta, Matt Gardner, and Jonathan Berant. COVR: A test-bed for visually grounded compositional generalization with real images. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9824–9846, 2021. doi: 10.18653/v1/2021.emnlp-main.774.
- Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillepie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avani Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv*, 2108.07258, 2022.
- Brady, Jack, Roland S. Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. *arXiv*, 2305.14229, 2023.
- BRAIN Initiative Cell Census Network, (BICCN). A multimodal cell census and atlas of the mammalian primary motor cortex. *Nature*, 598:86–102, 2021. doi: 0.1038/s41586-021-03950-0.
- Brendel, Wieland and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Brodman, Korbinian. *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. J.A. Barth, Leipzig, 1909.
- Bronstein, Michael M., Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. doi: 10.1109/MSP.2017.2693418.
- Burgess, Christopher P., Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv*, 1901.11390, 2019.

- Cabannes, Vivien, Bobak T. Kiani, Randall Balestriero, Yann LeCun, and Alberto Bietti. The SSL interplay: Augmentations, inductive bias, and generalization. In *Proc. of the International Conf. on Machine learning (ICML)*, 2023.
- Cadwell, Cathryn R., Athanasia Palasantza, Xiaolong Jiang, Philipp Berens, Qiaolin Deng, Marlene Yilmaz, Jacob Reimer, Shan Shen, Matthias Bethge, Kimberley F. Tolias, Rickard Sandberg, and Andreas S. Tolias. Electrophysiological, transcriptomic and morphologic profiling of single neurons using patch-seq. *Nature Biotechnology*, 34:199–203, 2016. doi: 10.1038/nbt.3445.
- Campbell, Murray, A. Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial Intelligence*, 134: 57–83, 2002. doi: 10.1016/S0004-3702(01)00129-1.
- Carey, Susan. *The Origin of Concepts*. Oxford University Press, New York, 2009. ISBN 978-0-19-536763-8.
- Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.
- Celii, Brendan, Stelios Papadopoulos, Zhuokun Ding, Paul G. Fahey, Eric Wang, Christos Papadopoulos, Alexander B. Kunin, Saumil Patel, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Erick Cobos, Sven Dorckenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, Casey M. Schneider-Mizell, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Szi-chieh Yu, Wenjing Yin, Daniel Xenos, Lindsey M. Kitchell, Patricia K. Rivlin, Victoria A. Rose, Caitlyn A. Bishop, Brock Wester, Emmanouil Froudarakis, Edgar Y. Walker, Fabian Sinz, H. Sebastian Seung, Forrest Collman, Nuno Maçarico da Costa, R. Clay Reid, Xaq Pitkow, Andreas S. Tolias, and Jacob Reimer. Neurd: automated proofreading and feature extraction for connectomics. *bioRxiv*, 2023. doi: 10.1101/2023.03.14.532674.
- Chen, Siyi, Hermann J. Müller, and Markus Conci. Amodal completion in visual working memory. *Journal of Experimental Psychology. Human Perception and Performance*, 42(9):1344–1353, 2016. doi: 10.1037/xhp0000231.
- Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.
- Chomsky, Noam. *Aspects of the Theory of Syntax*. The MIT Press, Cambridge, Massachusetts, 1965.
- Common Crawl. Common crawl corpus. URL: <https://commoncrawl.org/>. Accessed 2023-07-23.
- Coogan, Thomas A. and Andreas Burkhalter. Hierarchical organization of areas in rat visual cortex. *Journal of Neuroscience*, 13(9):3749–3772, 1993. doi: 10.1523/JNEUROSCI.13-09-03749.1993.
- Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- Crawford, Eric and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2019.

- Dalal, Navneet and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005. doi: 10.1109/CVPR.2005.177.
- de Vries, Saskia E. J., Jerome A. Lecoq, Michael A. Buice, Peter A. Groblewski, Gabriel K. Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, Kate Roll, Marina Garrett, Tom Keenan, Leonard Kuan, Stefan Mihalas, Shawn Olsen, Carol Thompson, Wayne Wakeman, Jack Waters, Derric Williams, Chris Barber, Nathan Berbesque, Brandon Blanchard, Nicholas Bowles, Shiella D. Caldejon, Linzy Casal, Andrew Cho, Sissy Cross, Chinh Dang, Tim Dolbeare, Melise Edwards, John Galbraith, Nathalie Gaudreault, Terri L. Gilbert, Fiona Griffin, Perry Hargrave, Robert Howard, Lawrence Huang, Sean Jewell, Nika Keller, Ulf Knoblich, Josh D. Larkin, Rachael Larsen, Chris Lau, Eric Lee, Felix Lee, Arielle Leon, Lu Li, Fuhui Long, Jennifer Luviano, Kyla Mace, Thuyanh Nguyen, Jed Perkins, Miranda Robertson, Sam Seid, Eric Shea-Brown, Jianghong Shi, Nathan Sjoquist, Cliff Slaughterbeck, David Sullivan, Ryan Valenza, Casey White, Ali Williford, Daniela M. Witten, Jun Zhuang, Hongkui Zeng, Colin Farrell, Lydia Ng, Amy Bernard, John W. Phillips, R. Clay Reid, and Christof Koch. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23: 138–151, 2020. doi: 10.1038/s41593-019-0550-9.
- DeFelipe, Javier and Isabel Fariñas. The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs. *Progress in Neurobiology*, 39(6):563–607, 1993. doi: 10.1016/0301-0082(92)90015-7.
- DeFelipe, Javier, Lidia Alonso-Nanclares, and Jon I. Arellano. Microstructure of the neocortex: Comparative aspects. *Journal of Neurocytology*, 31:299–316, 2002.
- DeFelipe, Javier, Pedro L. López-Cruz, Ruth Benavides-Piccione, Concha Bielza, Pedro Larrañaga, Stewart Anderson, Andreas Burkhalter, Bruno Cauli, Alfonso Fairén, Dirk Feldmeyer, Gord Fishell, David Fitzpatrick, Tamás F. Freund, Guillermo González-Burgos, Shaul Hestrin, Sean Hill, Patrick R. Hof, Josh Huang, Edward G. Jones, Yasuo Kawaguchi, Zoltán Kisvárdy, Yoshiyuki Kubota, David A. Lewis, Oscar Marín, Henry Markram, Chris J. McBain, Hanno S. Meyer, Hannah Monyer, Sacha B. Nelson, Kathleen Rockland, Jean Rossier, John L. R. Rubenstein, Bernardo Rudy, Massimo Scanziani, Gordon M. Shepherd, Chet C. Sherwood, Jochen F. Staiger, Gábor Tamás, Alex Thomson, Yun Wang, Rafael Yuste, and Giorgio A. Ascoli. New insights into the classification and nomenclature of cortical gabaergic interneurons. *Nature Reviews Neuroscience*, 14:202–216, 2013. doi: 10.1038/nrn3444.
- Dendorfer, Patrick, Hamid RezaTofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv*, 2003.09003, 2020.
- Desislavov, Radosvet, Fernando Martínez-Plumed, and José Hernández-Orallo. Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38, 2023. doi: 10.1016/j.suscom.2023.100857.
- Didolkar, Aniket, Anirudh Goyal, and Yoshua Bengio. Cycle consistency driven object discovery. *arXiv*, 2306.02204, 2023.
- Dittadi, Andrea, Samuele S. Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In *Proc. of the International Conf. on Machine learning (ICML)*, 2022.



- Diuk, Carlos, Andre Cohen, and Michael L. Littman. An object-oriented representation for efficient reinforcement learning. In *Proc. of the International Conf. on Machine learning (ICML)*, 2008.
- Dorkenwald, Sven, Peter H. Li, Michał Januszewski, Daniel R. Berger, Jeremy Maitin-Shepard, Agnes L. Bodor, Forrest Collman, Casey M. Schneider-Mizell, Nuno Maçarico da Costa, Jeff W. Lichtman, and Viren Jain. Multi-layered maps of neuropil with segmentation-guided contrastive learning. *bioRxiv*, 2022. doi: 10.1101/2022.03.29.486320.
- Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2021.
- D’Souza, Rinaldo D., Quanxin Wang, Weiqing Ji, Andrew M. Meier, Henry Kennedy, Kenneth Knoblauch, and Andreas Burkhalter. Hierarchical and nonhierarchical features of the mouse visual cortical network. *Nature Communications*, 13(503), 2022. doi: 10.1038/s41467-022-28035-y.
- Dwivedi, Vijay Prakash and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.
- Dwivedi, Vijay Prakash, Ladislav Rampásek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- Elabbady, Leila, Sharmishta Seshamani, Shang Mu, Gayathri Mahalingam, Casey Schneider-Mizell, Agnes Bodor, J. Alexander Bae, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Erick Cobos, Sven Dorkenwald, Paul G. Fahey, Emmanouil Froudarakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Barak Nehoran, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, Jacob Reimer, William Silversmith, Fabian H. Sinz, Marc Takeno, Russel Torres, Nicholas Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Andreas Tolias, H. Sebastian Seung, R. Clay Reid, Nuno Maçarico Da Costa, and Forrest Collman. Quantitative census of local somatic features in mouse visual cortex. *bioRxiv*, 2022. doi: 10.1101/2022.07.20.499976.
- Elsayed, Gamaleldin F., Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Engelcke, Martin, Oiwi Parker Jones, and Ingmar Posner. Reconstruction bottlenecks in object-centric generative models. *ICML 2020 Workshop: Object-Oriented Learning (OOL)*, 2020a.
- Engelcke, Martin, Adam R. Kosiorok, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative scene inference and sampling with object-centric latent representations. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020b.
- Engelcke, Martin, Oiwi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring unordered object representations without iterative refinement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Eslami, S. M. Ali, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

- Espeholt, Lasse, Shreya Agrawal, Casper Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gizen, Rob Carver, Marcin Andrychowicz, Jason Hickey, Aaron Bell, and Nal Kalchbrenner. Deep learning for twelve hour precipitation forecasts. *Nature Communications*, 13(1):5145, 2022. doi: 10.1038/s41467-022-32483-x.
- Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017. doi: 10.1038/nature21056.
- Esteva, Andre, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019. doi: 10.1038/s41591-018-0316-z.
- Felleman, Daniel J. and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991. doi: 10.1093/cercor/1.1.1-a.
- Fishell, Gord and Nathaniel Heintz. The neuron identity problem: Form meets function. *Neuron*, 80(3):602–12, 2013. doi: 10.1016/j.neuron.2013.10.035.
- Fişek, Mehmet, Dustin Herrmann, Alexander Egea-Weiss, Matilda Cloves, Lisa Bauer, Tai-Ying Lee, Lloyd Russell, and Michael Häusser. Cortico-cortical feedback engages active dendrites in visual cortex. *Nature*, 617:769–776, 2023. doi: 10.1038/s41586-023-06007-6.
- Fuzik, János, Amit Zeisel, Zoltán Máté, Daniela Calvigioni, Yuchio Yanagawa, Gábor Szabó, Sten Linnarsson, and Tibor Harkany. Integration of electrophysiological recordings with single-cell RNA-seq data identifies neuronal subtypes. *Nature Biotechnology*, 34:175–183, 2015. doi: 10.1038/nbt.3443.
- Garcia, John and Robert A. Koelling. Relation of cue to consequence in avoidance learning. *Psychonomic Science*, 4:123–124, 1966.
- Garcia, John, Donald J. Kimeldorf, and Robert A. Koellino. Conditioned aversion to saccharin resulting from exposure to gamma radiation. *Science*, 122(3160):157–158, 1955. doi: 10.1126/science.122.3160.157.
- Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. doi: 10.1177/0278364913491297.
- Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z.
- Gilbert, Charles D. and Wu Li. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14:350–363, 2013. doi: 10.1038/nrn3476.

- Glickfeld, Lindsey L. and Shawn R. Olsen. Higher-order areas of the mouse visual cortex. *Annual review of vision science*, 3:251–273, 2017. doi: 10.1146/annurev-vision-102016-061331.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, Cambridge, Massachusetts, 2016.
- Gouwens, Nathan W., Staci A. Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M. Sunkin, David Feng, Costas A. Anastassiou, Eliza Barkan, Kris Bickley, Nicole Blesie, Thomas Braun, Krissy Brouner, Agata Budzillo, Shiella Caldejon, Tamara Casper, Dan Castelli, Peter Chong, Kirsten Crichton, Christine Cuhaciyan, Tanya L. Daigle, Rachel Dalley, Nick Dee, Tsega Desta, Song-Lin Ding, Samuel Dingman, Alyse Doperalski, Nadezhda Dotson, Tom Egdorf, Michael Fisher, Rebecca A. de Frates, Emma Garren, Marissa Garwood, Amanda Gary, Nathalie Gaudreault, Keith Godfrey, Melissa Gorham, Hong Gu, Caroline Habel, Kristen Hadley, James Harrington, Julie A. Harris, Alex Henry, DiJon Hill, Sam Josephsen, Sara Kebede, Lisa Kim, Matthew Kroll, Brian Lee, Tracy Lemon, Katherine E. Link, Xiaoxiao Liu, Brian Long, Rusty Mann, Medea McGraw, Stefan Mihalas, Alice Mukora, Gabe J. Murphy, Lindsay Ng, Kiet Ngo, Thuc Nghi Nguyen, Philip R. Nicovich, Aaron Oldre, Daniel Park, Sheana Parry, Jed Perkins, Lydia Potekhina, David Reid, Miranda Robertson, David Sandman, Martin Schroedter, Cliff Slaughterbeck, Gilberto Soler-Llavina, Josef Sulc, Aaron Szafer, Bosiljka Tasic, Naz Taskin, Corinne Teeter, Nivretta Thatra, Herman Tung, Wayne Wakeman, Grace Williams, Rob Young, Zhi Zhou, Colin Farrell, Hanchuan Peng, Michael J. Hawrylycz, Ed Lein, Lydia Ng, Anton Arkhipov, Amy Bernard, John W. Phillips, Hongkui Zeng, and Christof Koch. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature Neuroscience*, 22:1182–1195, 2019. doi: 10.1038/s41593-019-0417-0.
- Gouwens, Nathan W., Staci A. Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R. Lee, Tim Jarsky, Lauren Alfiler, Katherine Baker, Eliza Barkan, Kyla Berry, Darren Bertagnolli, Kris Bickley, Jasmine Bomben, Thomas Braun, Krissy Brouner, Tamara Casper, Kirsten Crichton, Tanya L. Daigle, Rachel Dalley, Rebecca A. de Frates, Nick Dee, Tsega Desta, Samuel Dingman Lee, Nadezhda Dotson, Tom Egdorf, Lauren Ellingwood, Rachel Enstrom, Luke Esposito, Colin Farrell, David Feng, Olivia Fong, Rohan Gala, Clare Gamlin, Amanda Gary, Alexandra Glandon, Jeff Goldy, Melissa Gorham, Lucas Graybuck, Hong Gu, Kristen Hadley, Michael J. Hawrylycz, Alex M. Henry, DiJon Hill, Madie Hupp, Sara Kebede, Tae Kyung Kim, Lisa Kim, Matthew Kroll, Changkyu Lee, Katherine E. Link, Matthew Mallory, Rusty Mann, Michelle Maxwell, Medea McGraw, Delissa McMillen, Alice Mukora, Lindsay Ng, Lydia Ng, Kiet Ngo, Philip R. Nicovich, Aaron Oldre, Daniel Park, Hanchuan Peng, Osnat Penn, Thanh Pham, Alice Pom, Zoran Popović, Lydia Potekhina, Ramkumar Rajanbabu, Shea Ransford, David Reid, Christine Rimorin, Miranda Robertson, Kara Ronellenfitch, Augustin Ruiz, David Sandman, Kimberly Smith, Josef Sulc, Susan M. Sunkin, Aaron Szafer, Michael Tieu, Amy Torkelson, Jessica Trinh, Herman Tung, Wayne Wakeman, Katelyn Ward, Grace Williams, Zhi Zhou, Jonathan T. Ting, Anton Arkhipov, Uygur Sümbül, Ed S. Lein, Christof Koch, Zizhen Yao, Bosiljka Tasic, Jim Berg, Gabe J. Murphy, and Hongkui Zeng. Integrated morphoelectric and transcriptomic classification of cortical gabaergic cells. *Cell*, 183(4):935–953, 2020. doi: 10.1016/j.cell.2020.09.057.
- Goyal, Anirudh and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 2022. doi: 10.1098/rspa.2021.0068.
- Greff, Klaus, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

- Greff, Klaus, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019.
- Greff, Klaus, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv*, 2012.05208, 2020.
- Greff, Klaus, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- Greig, Luciano C., Mollie B. Woodworth, Maria J. Galazo, Hari Padmanabhan, and Jeffrey D. Macklis. Molecular logic of neocortical projection neuron specification, development and diversity. *Nature Reviews Neuroscience*, 14:755–769, 2013. doi: 10.1038/nrn3586.
- Harris, Kenneth D. and Gordon M.G. Shepherd. The neocortical circuit: Themes and variations. *Nature Neuroscience*, 18:170–181, 2015. doi: 10.1038/nn.3917.
- Hattox, Alexis M. and Sacha B. Nelson. Layer V neurons in mouse cortex projecting to different targets have distinct physiological properties. *Journal of Neurophysiology*, 98:3330–3340, 2007. doi: 10.1152/jn.00397.2007.
- Hauser, Marc D., Pogen MacNeilage, and Molly Ware. Numerical representations in primates. *Proceedings of the National Academy of Sciences*, 93(4):1514–1517, 1996. doi: 10.1073/pnas.93.4.1514.
- He, Zhen, Jian Li, Daxue Liu, Hangen He, and David Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- Herculano-Houzel, Suzana, Kenneth Catania, Paul R. Manger, and Jon H. Kaas. Mammalian brains are made of these: A dataset of the numbers and densities of neuronal and nonneuronal cells in the brain of glires, primates, scandentia, eulipotyphlans, afrotherians and artiodactyls, and their relationship with body mass. *Brain Behavior and Evolution*, 86(3-4):145–163, 2015. doi: 10.1159/000437413.
- Hinton, Geoffrey E. How to represent part-whole hierarchies in a neural network. *arXiv*, 2102.12627, 2021.
- Hinton, Geoffrey E., James L. McClelland, and David E. Rumelhart. Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. The MIT Press, Cambridge, Massachusetts, 1984.
- Hüllermeier, Eyke, Thomas Fober, and Marco Mernberger. Inductive bias. In *Encyclopedia of Systems Biology*, pages 1018–1018. Springer New York, 2013. doi: 10.1007/978-1-4419-9863-7\_927.
- Hume, David. *A treatise of human nature*. John Noon, London, 1739.

- Jacobsen, Jörn-Henrik, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Jaderberg, Max, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- Jiang, Jindong and Sungjin Ahn. Generative neurosymbolic machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Jiang, Jindong, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative world models with scalable object representations. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Jiang, Xiaolong, Shan Shen, Cathryn R. Cadwell, Philipp Berens, Fabian Sinz, Alexander S. Ecker, Saumil Patel, and Andreas S. Tolias. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science*, 350, 2015. doi: 10.1126/science.aac9462.
- Jing, Longlong and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv*, 1902.06162, 2019.
- Joshi, Pratik, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, 2020. doi: 10.18653/v1/2020.acl-main.560.
- Kabra, Rishabh, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONE: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Kanari, Lida, Srikanth Ramaswamy, Ying Shi, Sébastien Morand, Julie Meystre, Rodrigo Perin, Marwan Abdellah, Yun Wang, Kathryn Hess, and Henry Markram. Objective morphological classification of neocortical pyramidal cells. *Cerebral cortex*, 29(4):1719–1735, 2019. doi: 10.1093/cercor/bhy339.
- Kandel, Eric R., James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, A. J. Hudspeth, and Sarah Mack. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.
- Karazija, Laurynas, Iro Laina, and Christian Rupprecht. ClevrTex: A texture-rich benchmark for unsupervised multi-object segmentation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.
- Karazija, Laurynas, Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised multi-object segmentation by predicting probable motion patterns. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Keysers, Daniel, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

- Kim, Euiseok J., Ashley L. Juavinett, Espoir M. Kyubwa, Matthew W. Jacobs, and Edward M. Callaway. Three types of cortical layer 5 neurons that differ in brain-wide connectivity and function. *Neuron*, 88(6):1253–1267, 2015. doi: 10.1016/j.neuron.2015.11.002.
- Kipf, Thomas, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022.
- Kirillov, Alexander, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv*, 2304.02643, 2023.
- Kosioerek, Adam, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.
- Lake, Brenden and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proc. of the International Conf. on Machine learning (ICML)*, 2018.
- Lake, Brenden M., Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050.
- Larkman, Alan U. Dendritic morphology of pyramidal neurones of the visual cortex of the rat: I. Branching patterns. *Journal of Comparative Neurology*, 306(2):307–319, 1991. doi: 10.1002/cne.903060207.
- Larkum, Matthew. A cellular mechanism for cortical associations: An organizing principle for the cerebral cortex. *Trends in Neurosciences*, 36(3):141–151, 2013. doi: 10.1016/j.tins.2012.11.006.
- Laternus, Sophie, Adam von Daranyi, Ziwei Huang, and Philipp Berens. Morphopy: A python package for feature extraction of neural morphologies. *Journal of Open Source Software*, 5(52):2339, 2020. doi: 10.21105/joss.02339.
- Lea, Stephen E. G., Alan M. Slater, and Catriona M. E. Ryan. Perception of object unity in chicks: A comparison with the human infant. *Infant Behavior and Development*, 19(4):501–504, 1996. doi: 10.1016/S0163-6383(96)90010-7.
- LeCun, Yann and Christopher Manning. What innate priors should we build into the architecture of deep learning systems? 2018. URL: <http://www.abigailsee.com/2018/02/21/deep-learning-structure-and-innate-priors.html>. Accessed 2023-07-16.
- LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

- LeCun, Yann, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521:436–444, 2015. doi: 10.1038/nature14539.
- Lewis, Martha, Nihal V. Nayak, Peilin Yu, Qinan Yu, Jack Merullo, Stephen H. Bach, and Ellie Pavlick. Does CLIP bind concepts? Probing compositionality in large image models. *arXiv*, 2212.10537, 2023.
- Li, Yuanpeng, Liang Zhao, Kenneth Church, and Mohamed Elhoseiny. Compositional language continual learning. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *arXiv*, 1405.0312, 2014.
- Lin, Zhixuan, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019.
- Locatello, Francesco, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Loula, João, Marco Baroni, and Brenden M. Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv*, 1807.07545, 2018.
- Lowe, David G. Object recognition from local scale-invariant features. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 1999. doi: 10.1109/ICCV.1999.790410.
- Ma, Zixian, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. CREPE: Can vision-language foundation models reason compositionally? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Malhotra, Gaurav, Benjamin D. Evans, and Jeffrey S. Bowers. Hiding a plane with a pixel: Examining shape-bias in cnns and the benefit of building in biological constraints. *Vision Research*, 174:57–68, 2020. doi: 10.1016/j.visres.2020.04.013.
- Marcus, Gary. Innateness, alphazero, and artificial intelligence. *arXiv*, 1801.05667, 2018.
- Markram, Henry, Eilif Muller, Srikanth Ramaswamy, Michael W. Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, Guy Antoine Atenekeg Kahou, Thomas K. Berger, Ahmet Bilgili, Nenad Buncic, Athanassia Chalimourda, Giuseppe Chindemi, Jean-Denis Courcol, Fabien Delalondre, Vincent Delattre, Shaul Druckmann, Raphael Dumusc, James Dynes, Stefan Eilemann, Eyal Gal, Michael Emiel Gevaert, Jean-Pierre Ghobril, Albert Gidon, Joe W. Graham, Anirudh Gupta, Valentin Haenel, Etay Hay, Thomas Heinis, Juan B. Hernandez, Michael Hines, Lida Kanari, Daniel Keller, John Kenyon, Georges Khazen, Yihwa Kim, James G. King, Zoltan Kisvarday, Pramod Kumbhar, Sébastien Lasserre, Jean-Vincent Le Bé, Bruno R.C. Magalhães, Angel Merchán-Pérez, Julie Meystre,

- Benjamin Roy Morrice, Jeffrey Muller, Alberto Muñoz-Céspedes, Shruti Muralidhar, Keerthan Muthurasa, Daniel Nachbaur, Taylor H. Newton, Max Nolte, Aleksandr Ovcharenko, Juan Palacios, Luis Pastor, Rodrigo Perin, Rajnish Ranjan, Imad Riachi, José-Rodrigo Rodríguez, Juan Luis Riquelme, Christian Rössert, Konstantinos Sfyarakis, Ying Shi, Julian C. Shillcock, Gilad Silberberg, Ricardo Silva, Farhan Tauheed, Martin Telefont, Maria Toledo-Rodriguez, Thomas Tränkler, Werner Van Geit, Jafet Villafranca Díaz, Richard Walker, Yun Wang, Stefano M. Zaninetta, Javier DeFelipe, Sean L. Hill, Idan Segev, and Felix Schürmann. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015. doi: 10.1016/j.cell.2015.09.029.
- Marshel, James H., Marina E. Garrett, Ian Nauhaus, and Edward M. Callaway. Functional specialization of seven mouse visual cortical areas. *Neuron*, 72(6):1040–1054, 2011. doi: 10.1016/j.neuron.2011.12.004.
- Marx, Manuel and Dirk Feldmeyer. Morphology and physiology of excitatory neurons in layer 6b of the somatosensory rat barrel cortex. *Cerebral cortex*, 23(12):2803–2817, 2012. doi: 10.1093/cercor/bhs254.
- Masland, Richard H. Neuronal cell types. *Current biology*, 14(13):497–500, 2004. doi: 10.1016/j.cub.2004.06.035.
- Mendez, Jorge A. and Eric Eaton. Lifelong learning of compositional structures. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2021.
- Mendez, Jorge A. and Eric Eaton. How to reuse and compose knowledge for a lifetime of tasks: A survey on continual learning and functional composition. *Transactions on Machine Learning Research (TMLR)*, 2023. ISSN 2835-8856.
- MICrONS Consortium, The, J. Alexander Bae, Mahaly Baptiste, Caitlyn A. Bishop, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Brendan Celii, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Sven Dorckenwald, Leila Elabbady, Paul G. Fahey, Tim Fliss, Emmanouil Froudarakis, Jay Gager, Clare Gamlin, William Gray-Roncal, Akhilesh Halageri, James Hebditch, Zhen Jia, Emily Joyce, Justin Joyce, Chris Jordan, Daniel Kapner, Nico Kemnitz, Sam Kinn, Lindsey M. Kitchell, Selden Koolman, Kai Kuehner, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Jordan Matelsky, Sarah McReynolds, Elanine Miranda, Eric Mitchell, Shanka Subhra Mondal, Merlin Moore, Shang Mu, Taliah Muhammad, Barak Nehoran, Oluwaseun Ogedengbe, Christos Papadopoulos, Stelios Papadopoulos, Saamil Patel, Xaq Pitkow, Sergiy Popovych, Anthony Ramos, R. Clay Reid, Jacob Reimer, Patricia K. Rivlin, Victoria Rose, Casey M. Schneider-Mizell, H. Sebastian Seung, Ben Silverman, William Silversmith, Amy Sterling, Fabian H. Sinz, Cameron L. Smith, Shelby Suckow, Marc Takeno, Zheng H. Tan, Andreas S. Tolias, Russel Torres, Nicholas L. Turner, Edgar Y. Walker, Tianyu Wang, Adrian Wanner, Brock A. Wester, Grace Williams, Sarah Williams, Kyle Willie, Ryan Willie, William Wong, Jingpeng Wu, Chris Xu, Runzhe Yang, Dimitri Yatsenko, Fei Ye, Wenjing Yin, Rob Young, Szi-chieh Yu, Daniel Xenos, and Chi Zhang. Functional connectomics spanning multiple areas of mouse visual cortex. *bioRxiv*, 2023. doi: 10.1101/2021.07.28.454025.
- Milan, Anton, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv*, 1603.00831, 2016.
- Mitchell, Tom M. The need for biases in learning generalizations. Technical report, Rutgers University, 1980.



- Mukamel, Eran A. and John Ngai. Perspectives on defining cell types in the brain. *Current Opinion in Neurobiology*, 56:61–68, 2019. doi: 10.1016/j.conb.2018.11.007.
- Narayanan, Rajeevan T., Daniel Udvary, and Marcel Oberlaender. Cell type-specific structural organization of the six layers in rat barrel cortex. *Frontiers in Neuroanatomy*, 11, 2017. doi: 10.3389/fnana.2017.00091.
- Needham, Amy. Infants' use of featural information in the segregation of stationary objects. *Infant Behavior and Development*, 21(1):47–76, 1998. doi: 10.1016/S0163-6383(98)90054-6.
- Nelson, Sacha B., Ken Sugino, and Chris M. Hempel. The problem of neuronal cell types: a physiological genomics approach. *Trends in Neurosciences*, 29(6):339–345, 2006. doi: 10.1016/j.tins.2006.05.004.
- Oberlaender, Marcel, Christiaan P. J. de Kock, Randy M. Bruno, Alejandro Ramirez, Hanno S. Meyer, Vincent J. Dercksen, Moritz Helmstaedter, and Bert Sakmann. Cell type-specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissal cortex. *Cerebral Cortex*, 22(10): 2375–2391, 2011. doi: 10.1093/cercor/bhr317.
- OpenAI. GPT-4 technical report. *arXiv*, 2303.08774, 2023.
- Oquab, Maxime, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv*, 2304.07193, 2023.
- Oxford English Dictionary. morphometrics, n. Oxford University Press, 2023. doi: 10.1093/OED/6506928067. Accessed 2023-06-27.
- Papa, Samuele, Ole Winther, and Andrea Dittadi. Inductive biases for object-centric representations in the presence of complex textures. In *UAI 2022 Workshop on Causal Representation Learning*, 2022.
- Papernot, Nicolas, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017.
- Peng, Hanchuan, Peng Xie, Lijuan Liu, Xiuli Kuang, Yimin Wang, Lei Qu, Hui Gong, Shengdian Jiang, Anan Li, Zongcai Ruan, Liya Ding, Zizhen Yao, Chao Chen, Mengya Chen, Tanya L. Daigle, Rachel Dalley, Zhangcan Ding, Yanjun Duan, Aaron Feiner, Ping He, Chris Hill, Karla E. Hirokawa, Guodong Hong, Lei Huang, Sara Kebede, Hsien-Chi Kuo, Rachael Larsen, Phil Lesnar, Longfei Li, Qi Li, Xiangning Li, Yaoyao Li, Yuanyuan Li, An Liu, Donghuan Lu, Stephanie Mok, Lydia Ng, Thuc Nghi Nguyen, Qiang Ouyang, Jintao Pan, Elise Shen, Yuanyuan Song, Susan M. Sunkin, Bosiljka Tasic, Matthew B. Veldman, Wayne Wakeman, Wan Wan, Peng Wang, Quanxin Wang, Tao Wang, Yaping Wang, Feng Xiong, Wei Xiong, Wenjie Xu, Min Ye, Lulu Yin, Yang Yu, Jia Yuan, Jing Yuan, Zhixi Yun, Shaoqun Zeng, Shichen Zhang, Sujun Zhao, Zijun Zhao, Zhi Zhou, Z. Josh Huang, Luke Esposito, Michael J. Hawrylycz, Staci A. Sorensen, X. William Yang, Yefeng Zheng, Zhongze Gu, Wei Xie, Christof Koch, Qingming Luo, Julie A. Harris, Yun Wang, and Hongkui Zeng. Morphological diversity of single neurons in molecularly defined cell types. *Nature*, 598: 174–181, 2021. doi: 10.1038/s41586-021-03941-1.

- Peters, Benjamin and Nikolaus Kriegeskorte. Capturing the objects of vision with neural networks. *Nature Human Behaviour*, 5:1127–1144, 2021. doi: 10.1038/s41562-021-01194-6.
- Raistrick, Alexander, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Ramón y Cajal, Santiago. *Histologie du système nerveux de l’homme et des vertébrés*. Paris, 1911.
- Regolin, Lucia and Giorgio Vallortigara. Perception of partly occluded objects by young chicks. *Perception & psychophysics*, 57:971–976, 1995. doi: 10.3758/BF03205456.
- Robinson, Gene E. Beyond nature and nurture. *Science*, 304(5669):397–399, 2004. doi: 10.1126/science.1095766.
- Rosenblatt, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958. doi: 10.1037/h0042519.
- Ruis, Laura, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2003.05161, 2020.
- Rusch, T. Konstantin, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv*, 2303.10993, 2023.
- Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- Sajjadi, Mehdi S. M., Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Saunders, Arpiar, Evan Macosko, Alec Wysoker, Melissa Goldman, Fenna Krienen, Heather Rivera, Elizabeth Bien, Matthew Baum, Laura Bortolin, Shuyu Wang, Aleksandrina Goeva, James Nemes, Nolan Kamitaki, Sara Brumbaugh, David Kulp, and Steven Mccarroll. Molecular diversity and specializations among the cells of the adult mouse brain. *Cell*, 174(4):1015–1030, 2018. doi: 10.1016/j.cell.2018.07.028.
- Scala, Federico, Dmitry Kobak, Shan Shen, Yves Bernaerts, Sophie Laturus, Cathryn Cadwell, Leonard Hartmanis, Emmanouil Froudarakis, Jesus Castro, Zheng Tan, Stelios Papadopoulos, Saumil Patel, Rickard Sandberg, Philipp Berens, Xiaolong Jiang, and Andreas S. Tolias. Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas. *Nature Communications*, 10:1–12, 2019. doi: 10.1038/s41467-019-12058-z.
- Scala, Federico, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn R. Cadwell, Jesus Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Laturus, Elanina Miranda, Shalaka Mulherkar, Zheng H. Tan, Zizhen Yao, Hongkui Zeng, Rickard Sandberg, Philipp Berens, and Andreas S. Tolias. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598: 144–150, 2021. doi: 10.1038/s41586-020-2907-3.

- Scarselli, Franco, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.
- Schneider-Mizell, Casey M., Agnes Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Leila Elabbady, Daniel Kapner, Sam Kinn, Gayathri Mahalingam, Sharmishta Seshamani, Shelby Suckow, Marc Takeno, Russel Torres, Wenjing Yin, Sven Dorkenwald, J. Alexander Bae, Manuel A. Castro, Paul G. Fahey, Emmanouil Froudakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Nico Kemnitz, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, William Silversmith, Fabian H. Sinz, Nicholas L. Turner, William Wong, Jingpeng Wu, Szi-chieh Yu, MICrONS Consortium, Jacob Reimer, Andreas S. Tolia, H. Sebastian Seung, R. Clay Reid, Forrest Collman, and Nuno Maçarico da Costa. Cell-type-specific inhibitory circuitry from a connectomic census of mouse visual cortex. *bioRxiv*, 2023. doi: 10.1101/2023.01.23.525290.
- Schuhmann, Christoph, Romain Beaumont, Richard Vencu, Cade W. Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R. Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.
- Schulz, Eric, Joshua B. Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel J. Gershman. Compositional inductive biases in function learning. *Cognitive Psychology*, 99:44–79, 2017. doi: 10.1016/j.cogpsych.2017.11.002.
- Scorcioni, Ruggero, Sridevi Polavaram, and Giorgio Ascoli. L-measure: A web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols*, 3:866–876, 2008. doi: 10.1038/nprot.2008.51.
- Seitzer, Maximilian, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2023.
- Senzai, Yuta, Antonio Fernández-Ruiz, and György Buzsáki. Layer-specific physiological features and interlaminar interactions in the primary visual cortex of the mouse. *Neuron*, 101(3):500–513, 2019.
- Sevilla, Jaime, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. In *International Joint Conference on Neural Networks (IJCNN)*, 2022. doi: 10.1109/IJCNN55064.2022.9891914.
- Silver, David, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. doi: 10.1038/nature16961.
- Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(5050):354–359, 2017. doi: 10.1038/nature24270.

- Singh, Gautam, Fei Deng, and Sungjin Ahn. Illiterate DALL-e learns to compose. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022a.
- Singh, Gautam, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.
- Smolensky, Paul, Richard T. McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. Neuro-compositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine*, 43(3):308–322, 2022. doi: 10.1002/aaai.12065.
- Spelke, Elizabeth S. Principles of object perception. *Cognitive Science*, 14(1):29–56, 1990. doi: 10.1207/s15516709cog1401\_3.
- Spelke, Elizabeth S. Nativism, empiricism, and the origins of knowledge. *Infant Behavior & Development*, 21(2):181–200, 1998. doi: 10.1016/S0163-6383(98)90002-9.
- Spelke, Elizabeth S. What makes us smart? Core knowledge and natural language. In *Language in mind: Advances in the study of language and thought*, chapter 10, pages 277–311. The MIT Press, Cambridge, Massachusetts, 2003.
- Spelke, Elizabeth S. and Katherine D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007. doi: 10.1111/j.1467-7687.2007.00569.x.
- Stanley, Geoffrey, Ozgun Gokce, Robert C. Malenka, Thomas Südhof, and Stephen R. Quake. Continuous and discrete neuron types of the adult murine striatum. *Neuron*, 105(4):688–699, 2019. doi: 10.1016/j.neuron.2019.11.004.
- Strubell, Emma, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, July 2019. doi: 10.18653/v1/P19-1355.
- Sun, Pei, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- Sun, Peize, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. DanceTrack: Multi-object tracking in uniform appearance and diverse motion. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- Sutton, Richard. The bitter lesson. *Incomplete Ideas (blog)*, 2019. URL: <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>. Accessed 2023-06-23.
- Tangemann, Matthias, Steffen Schneider, Julius Von Kügelgen, Francesco Locatello, Peter V. Gehler, Thomas Brox, Matthias Kuemmerer, Matthias Bethge, and Bernhard Schölkopf. Unsupervised object learning via common fate. In *Proc. of the Conf. on Causal Learning and Reasoning*, 2023.
- Tasic, Bosiljka, Vilas Menon, Thuc Nghi Nguyen, Tae Kyung Kim, Tim Jarsky, Zizhen Yao, Boaz Levi, Lucas T. Gray, Staci A Sorensen, Tim Dolbeare, Darren Bertagnolli, Jeff Goldy, Nadiya Shapovalova, Sheana Parry, Changkyu Lee, Kimberly Smith, Amy Bernard, Linda Madisen, Susan M. Sunkin, Michael Hawrylycz, Christof Koch, and Hongkui Zeng. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience*, 19:335–346, 2016. doi: 10.1038/nn.4216.

- Tasic, Bosiljka, Zizhen Yao, Lucas T. Graybiuck, Kimberly A. Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Sheana Parry, Nadiya V. Shapovalova, Daniel Hirschstein, Julie Pendergraft, Heather A. Sullivan, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Boaz P. Levi, Susan M. Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018. doi: 10.1038/s41586-018-0654-5.
- Tenenbaum, Joshua B., Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011. doi: 10.1126/science.1192788.
- Thomson, Alex M. Neocortical layer 6, a review. *Frontiers in Neuroanatomy*, 4, 2010. doi: 10.3389/fnana.2010.00013.
- Tolstikhin, Ilya, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Traynor, Bryan J. and Andrew B. Singleton. Nature versus nurture: Death of a dogma, and the road ahead. *Neuron*, 68(2):196–200, 2010. doi: 10.1016/j.neuron.2010.10.002.
- Ullman, Shimon, Daniel Harari, and Nimrod Dorfman. From simple innate biases to complex visual concepts. *Proceedings of the National Academy of Sciences*, 109(44):18215–18220, 2012. doi: 10.1073/pnas.1207690109.
- van der Maaten, Laurens and Geoffrey E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008.
- van Steenkiste, Sjoerd, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018.
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Veerapaneni, Rishi, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua B. Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Proc. Conf. on Robot Learning (CoRL)*, 2020.
- von Kügelgen, Julius, Ivan Ustyuzhaninov, Peter Gehler, Matthias Bethge, and Bernhard Schölkopf. Towards causal generative scene models via competition of experts. In *ICLR 2020 workshop “Causal learning for decision making”*, 2020.
- von Luxburg, Ulrike. Lecture notes in statistical machine learning, 2020. Department of Computer Science, University of Tübingen.

- Wagor, Earl, Nancy J. Mangini, and Alan L. Pearlman. Retinotopic organization of striate and extrastriate visual cortex in the mouse. *Journal of Comparative Neurology*, 193(1):187–202, 1980. doi: 10.1002/cne.901930113.
- Wang, Quanxin and Andreas Burkhalter. Area map of mouse visual cortex. *Journal of Comparative Neurology*, 502(2):339–357, 2007. doi: 10.1002/cne.21286.
- Wang, Wendy X. and Julie L. Lefebvre. Morphological pseudotime ordering and fate mapping reveal diversification of cerebellar inhibitory interneurons. *Nature Communication*, 13, 2022. doi: 10.1038/s41467-022-30977-2.
- Wang, Yanbo, Letao Liu, and Justin Dauwels. Slot-VAE: Object-centric scene generation with slot attention. *arXiv*, 2306.06997, 2023a.
- Wang, Yun, Min Ye, Xiuli Kuang, Yaoyao Li, and Shisi Hu. A simplified morphological classification scheme for pyramidal cells in six layers of primary somatosensory cortex of juvenile rats. *IBRO Reports*, 5:74–90, 2018. doi: 10.1016/j.ibror.2018.10.001.
- Wang, Ziyu, Mike Zheng Shou, and Mengmi Zhang. Object-centric learning with cyclic walks between parts and whole. *arXiv*, 2302.08023, 2023b.
- Weiler, Simon, Drago Guggiana Nilo, Tobias Bonhoeffer, Mark Hübener, Tobias Rose, and Volker Scheuss. Orientation and direction tuning align with dendritic morphology and spatial connectivity in mouse visual cortex. *Current Biology*, 32(8):1743–1753, 2022. doi: 10.1016/j.cub.2022.02.048.
- Weis, Marissa A., Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Benchmarking unsupervised object representations for video sequences. *Journal of Machine Learning Research (JMLR)*, 22(183):1–61, 2021.
- Weis, Marissa A., Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G. Fahey, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, R. Clay Reid, Casey M. Schneider-Mizell, H. Sebastian Seung, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Jacob Reimer, Andreas S. Tolias, and Alexander S. Ecker. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex. *bioRxiv*, 2023. doi: 10.1101/2022.12.22.521541.
- Welling, Max. Do we still need models or just more data and compute? 2019. URL: <https://staff.fnwi.uva.nl/m.welling/wp-content/uploads/Model-versus-Data-AI.pdf>. Accessed 2023-07-20.
- Wertheimer, Max. Untersuchungen zur Lehre von der Gestalt. II. *Psychologische Forschung*, 4:301–350, 1923. doi: 10.1007/BF00410640.
- Wiedemer, Thaddäus, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. *arXiv*, 2307.05596, 2023.
- Wolpert, David H. and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997. doi: 10.1109/4235.585893.

- Yang, Yafei and Bo Yang. Promising or elusive? Unsupervised object segmentation from real-world single images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Yao, Zizhen, Cindy T.J. van Velthoven, Thuc Nghi Nguyen, Jeff Goldy, Adriana E. Sedenio-Cortes, Fahimeh Baftizadeh, Darren Bertagnolli, Tamara Casper, Megan Chiang, Kirsten Crichton, Song-Lin Ding, Olivia Fong, Emma Garren, Alexandra Glandon, Nathan W. Gouwens, James Gray, Lucas T. Graybuck, Michael J. Hawrylycz, Daniel Hirschstein, Matthew Kroll, Kanan Lathia, Changkyu Lee, Boaz Levi, Delissa McMillen, Stephanie Mok, Thanh Pham, Qingzhong Ren, Christine Rimorin, Nadiya Shapovalova, Josef Sulc, Susan M. Sunkin, Michael Tieu, Amy Torkelson, Herman Tung, Katelyn Ward, Nick Dee, Kimberly A. Smith, Bosiljka Tasic, and Hongkui Zeng. A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation. *Cell*, 184(12):3222–3241, 2021. doi: 10.1016/j.cell.2021.04.021.
- Yao, Zizhen, Cindy T. J. van Velthoven, Michael Kunst, Meng Zhang, Delissa McMillen, Changkyu Lee, Won Jung, Jeff Goldy, Aliya Abdelhak, Pamela Baker, Eliza Barkan, Darren Bertagnolli, Jazmin Campos, Daniel Carey, Tamara Casper, Anish Bhaswanth Chakka, Rushil Chakrabarty, Sakshi Chavan, Min Chen, Michael Clark, Jennie Close, Kirsten Crichton, Scott Daniel, Tim Dolbeare, Lauren Ellingwood, James Gee, Alexandra Glandon, Jessica Gloe, Joshua Gould, James Gray, Nathan Guilford, Junitta Guzman, Daniel Hirschstein, Windy Ho, Kelly Jin, Matthew Kroll, Kanan Lathia, Arielle Leon, Brian Long, Zoe Maltzer, Naomi Martin, Rachel McCue, Emma Meyerderks, Thuc Nghi Nguyen, Trangthanh Pham, Christine Rimorin, Augustin Ruiz, Nadiya Shapovalova, Cliff Slaughterbeck, Josef Sulc, Michael Tieu, Amy Torkelson, Herman Tung, Nasmil Valera Cuevas, Katherine Wadhvani, Katelyn Ward, Boaz Levi, Colin Farrell, Carol L. Thompson, Shoaib Mufti, Chelsea M. Pagan, Lauren Kruse, Nick Dee, Susan M. Sunkin, Luke Esposito, Michael J. Hawrylycz, Jack Waters, Lydia Ng, Kimberly A. Smith, Bosiljka Tasic, Xiaowei Zhuang, and Hongkui Zeng. A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *bioRxiv*, 2023. doi: 10.1101/2023.03.06.531121.
- Young, Hedi, Beatriz Belbut, Margarida Baeta, and Leopoldo Petreanu. Laminar-specific cortico-cortical loops in mouse visual cortex. *eLife*, 10:e59551, 2021. doi: 10.7554/eLife.59551.
- Yuan, Jinyang, Tonglin Chen, Bin Li, and Xiangyang Xue. Compositional scene representation learning via reconstruction: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2023. doi: 10.1109/TPAMI.2023.3286184.
- Zadaianchuk, Andrii, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *arXiv*, 2306.04829, 2023.
- Zeisel, Amit, Ana Machado, Simone Codeluppi, P. Lonnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015. doi: 10.1126/science.aaa1934.
- Zeng, Hongkui. What is a cell type and how to define it? *Cell*, 185(15):2739–2755, 2022. doi: 10.1016/j.cell.2022.06.031.
- Zeng, Hongkui and Joshua R. Sanes. Neuronal cell-type classification: Challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18:530–546, 2017. doi: 10.1038/nrn.2017.85.
- Zhang, Jiawei, Haopeng Zhang, Congying Xia, and Li Sun. Graph-Bert: Only attention is needed for learning graph representations. *arXiv*, 2001.05140, 2020.

- Zhang, Meng, Stephen Eichhorn, Brian Zingg, Zizhen Yao, Kaelan Cotter, Hongkui Zeng, Hongwei Dong, and Xiaowei Zhuang. Spatially resolved cell atlas of the mouse primary motor cortex by MERFISH. *Nature*, 598:137–143, 2021. doi: 10.1038/s41586-021-03705-x.
- Zhang, Yifu, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.
- Zhu, Xiangxin, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. doi: 10.1109/CVPR.2014.122.
- Zimmermann, Roland S., Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Thomas Kipf, and Klaus Greff. Sensitivity of slot-based object-centric models to their number of slots. *arXiv*, 2305.18890, 2023.
- Zong, Zhuofan, Guanglu Song, and Yu Liu. DETRs with collaborative hybrid assignments training. *arXiv*, 2211.12860, 2023.
- Zoran, Daniel, Rishabh Kabra, Alexander Lerchner, and Danilo J. Rezende. PARTS: Unsupervised segmentation with slots, attention and independence maximization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021. doi: 10.1109/ICCV48922.2021.01027.







# Appendix

*The publications are ordered chronologically.*

Benchmarking unsupervised object representations for video sequences . . . . .	69
Self-supervised graph representation learning for neuronal morphologies . . . . .	133
Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex . . . . .	163



## Benchmarking unsupervised object representations for video sequences

*The following 61 pages have been published as:*

Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Benchmarking unsupervised object representations for video sequences. *Journal of Machine Learning Research (JMLR)*, 22(183):1–61, 2021.

*A summary of the motivation, results, and discussion can be found in Section 2.2 on page 20.*

### Abstract

Perceiving the world in terms of objects and tracking them through time is a crucial prerequisite for reasoning and scene understanding. Recently, several methods have been proposed for unsupervised learning of object-centric representations. However, since these models were evaluated on different downstream tasks, it remains unclear how they compare in terms of basic perceptual abilities such as detection, figure-ground segmentation and tracking of objects. To close this gap, we design a benchmark with four data sets of varying complexity and seven additional test sets featuring challenging tracking scenarios relevant for natural videos. Using this benchmark, we compare the perceptual abilities of four object-centric approaches: ViMON, a video-extension of MONET, based on recurrent spatial attention, OP3, which exploits clustering via spatial mixture models, as well as TBA and SCALOR, which use explicit factorization via spatial transformers. Our results suggest that the architectures with unconstrained latent representations learn more powerful representations in terms of object detection, segmentation and tracking than the spatial transformer based architectures. We also observe that none of the methods are able to gracefully handle the most challenging tracking scenarios despite their synthetic nature, suggesting that our benchmark may provide fruitful guidance towards learning more robust object-centric video representations.



# Benchmarking Unsupervised Object Representations for Video Sequences

Marissa A. Weis<sup>1,4</sup>

MARISSA.WEIS@UNI-GOETTINGEN.DE

Kashyap Chitta<sup>3,6</sup>

KASHYAP.CHITTA@TUE.MPG.DE

Yash Sharma<sup>4</sup>

YASH.SHARMA@UNI-TUEBINGEN.DE

Wieland Brendel<sup>4,5</sup>

WIELAND.BRENDEL@UNI-TUEBINGEN.DE

Matthias Bethge<sup>4,5</sup>

MATTHIAS.BETHGE@UNI-TUEBINGEN.DE

Andreas Geiger<sup>3,6</sup>

A.GEIGER@UNI-TUEBINGEN.DE

Alexander S. Ecker<sup>1,2,7</sup>

ECKER@CS.UNI-GOETTINGEN.DE

<sup>1</sup>*Institute of Computer Science, University of Göttingen, Germany*

<sup>2</sup>*Campus Institute Data Science, Göttingen, Germany*

<sup>3</sup>*Department of Computer Science, University of Tübingen, Germany*

<sup>4</sup>*Institute for Theoretical Physics, University of Tübingen, Germany*

<sup>5</sup>*Bernstein Center for Computational Neuroscience, Tübingen, Germany*

<sup>6</sup>*Max Planck Institute for Intelligent Systems, Tübingen, Germany*

<sup>7</sup>*Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany*

**Editor:** Christoph Lampert

## Abstract

Perceiving the world in terms of objects and tracking them through time is a crucial prerequisite for reasoning and scene understanding. Recently, several methods have been proposed for unsupervised learning of object-centric representations. However, since these models were evaluated on different downstream tasks, it remains unclear how they compare in terms of basic perceptual abilities such as detection, figure-ground segmentation and tracking of objects. To close this gap, we design a benchmark with four data sets of varying complexity and seven additional test sets featuring challenging tracking scenarios relevant for natural videos. Using this benchmark, we compare the perceptual abilities of four object-centric approaches: VIMON, a video-extension of MONET, based on recurrent spatial attention, OP3, which exploits clustering via spatial mixture models, as well as TBA and SCALOR, which use explicit factorization via spatial transformers. Our results suggest that the architectures with unconstrained latent representations learn more powerful representations in terms of object detection, segmentation and tracking than the spatial transformer based architectures. We also observe that none of the methods are able to gracefully handle the most challenging tracking scenarios despite their synthetic nature, suggesting that our benchmark may provide fruitful guidance towards learning more robust object-centric video representations.

**Keywords:** Unsupervised learning, object-centric representations, benchmark, tracking, segmentation

## 1. Introduction

Humans understand the world in terms of objects (Marr, 1982; Spelke and Kinzler, 2007; Johnson, 2018). Being able to decompose our environment into independent objects that can interact with each other is an important prerequisite for reasoning and scene understanding (Spelke, 1988). Similarly, an artificial intelligence system would benefit from the ability to represent visual scenes in a structured way (Ullman et al., 2017; Lake et al., 2017; Greff et al., 2020) by both extracting objects and their interactions from video streams, and keeping track of them over time.

Recently, there has been an increased interest in *unsupervised* learning of *object-centric representations*. The key insight of these methods is that the compositionality of visual scenes can be used to both discover (Eslami et al., 2016; Greff et al., 2019; Burgess et al., 2019) and track objects in videos (Greff et al., 2017; van Steenkiste et al., 2018; Veerapaneni et al., 2020) without supervision. However, it is currently not well understood how the learned visual representations of different models compare to each other quantitatively, since the models have been developed with different downstream tasks in mind and have not been evaluated using a common protocol. Hence, in this work, we propose a benchmark based on procedurally generated video sequences to test basic perceptual abilities of object-centric video models under various challenging tracking scenarios.

An unsupervised object-based video representation should (1) effectively identify objects as they enter a scene, (2) accurately segment objects, as well as (3) maintain a consistent representation for each individual object in a scene over time. These perceptual abilities can be evaluated quantitatively in the established *multi-object tracking* framework (Bernardin and Stiefelhagen, 2008; Milan et al., 2016). We propose to utilize this protocol for analyzing the strengths and weaknesses of different object-centric representation learning methods, independent of any specific downstream task, in order to uncover the different inductive biases hidden in their choice of architecture and loss formulation. We therefore compiled a benchmark consisting of four procedurally generated video data sets of varying levels of visual complexity and two generalization tests. Using this benchmark, we quantitatively compared three classes of object-centric models, leading to the following insights:

- All of the models have shortcomings handling occlusion, albeit to different extents.
- OP3 (Veerapaneni et al., 2020) performs strongest in terms of quantitative metrics, but exhibits a surprisingly strong dependency on color to separate objects and accumulates false positives when fewer objects than slots are present.
- Spatial transformer models, TBA (He et al., 2019) and SCALOR (Jiang et al., 2020), train most efficiently and feature explicit depth reasoning in combination with amodal masks, but are nevertheless outperformed by the simpler model, ViMON, lacking a depth or interaction model, suggesting that the proposed mechanisms may not yet work as intended.

Our code, data, as well as a public leaderboard of results is available at <https://eckerlab.org/code/weis2021/>.



## 2. Related work

Several recent lines of work propose to learn object-centric representations from visual inputs for static and dynamic scenes without explicit supervision. Though their results are promising, methods are currently restricted to handling synthetic data sets and as of yet are unable to scale to complex natural scenes. Furthermore, a systematic quantitative comparison of methods is lacking.

Selecting and processing parts of an image via *spatial attention* has been one prominent approach for this task (Mnih et al., 2014; Eslami et al., 2016; Kosiorek et al., 2018; Burgess et al., 2019; Yuan et al., 2019; Crawford and Pineau, 2019). As an alternative, *spatial mixture* models decompose scenes by performing image-space clustering of pixels that belong to individual objects (Greff et al., 2016, 2017, 2019; van Steenkiste et al., 2018; Locatello et al., 2020). Xu et al. (2019) use optic flow to learn to segment objects or object parts in dynamic scenes. While some approaches aim at learning a suitable representation for downstream tasks (Watters et al., 2019a; Veerapaneni et al., 2020), others target scene generation (Engelcke et al., 2020; Kügelgen et al., 2020; Ehrhardt et al., 2020; Jiang and Ahn, 2020). We analyze three classes of models for processing videos, covering three models based on spatial attention and one based on spatial mixture modeling.

**Spatial attention models with unconstrained latent representations** use per-object variational autoencoders, as introduced by Burgess et al. (2019). Kügelgen et al. (2020) adapt this approach for scene generation. So far, such methods have been designed for static images, but not for videos. We therefore extend MONET (Burgess et al., 2019) to be able to accumulate evidence over time for tracking, enabling us to include this class of approaches in our evaluation. Recent concurrent work on AlignNet (Creswell et al., 2020) applies MONET frame-by-frame and tracks objects by subsequently ordering the extracted objects consistently.

**Spatial attention models with factored latents** use an explicit factorization of the latent representation into properties such as position, scale and appearance (Eslami et al., 2016; Crawford and Pineau, 2019). These methods use spatial transformer networks (Jaderberg et al., 2015) to render per-object reconstructions from the factored latents (Kosiorek et al., 2018; He et al., 2019; Jiang et al., 2020). SQAIR (Kosiorek et al., 2018) does not perform segmentation, identifying objects only at the bounding-box level. Henderson and Lampert (2020) explicitly reason about the underlying 3D scene structure. We select Tracking-by-Animation (TBA) (He et al., 2019) and SCALOR (Jiang et al., 2020) for analyzing spatial transformer methods in our experiments, which explicitly disentangle object shape and appearance, providing access to object masks.

**Spatial mixture models** cluster pixels using a deep neural network trained with expectation maximization (Greff et al., 2017; van Steenkiste et al., 2018). IODINE (Greff et al., 2019) extends these methods with an iterative amortised variational inference procedure (Marino et al., 2018), improving segmentation quality. SPACE (Lin et al., 2020) combines mixture models with spatial attention to improve scalability. To work with video sequences, OP3 (Veerapaneni et al., 2020) extends IODINE by modeling individual objects' dynamics as well as pairwise interactions. We therefore include OP3 in our analysis as a representative spatial mixture model.

Data Set	Objects						Background	
	Shape	Motion	Count Over Sequence	Size Variation	Orientation	Color	Motion	Color
SpMOT	4 Templates (2D)	Linear	Varies (0-3)	Minimal	Fixed	6 Colors	None	Black
VOR	2 Templates (3D)	Static	Varies (0-4)	Moderate	Random	6 Colors	Moving Camera	Random
VMDS	3 Templates (2D)	Non-Linear	Fixed (1-4)	Moderate	Random	256 <sup>3</sup> Colors	None	256 <sup>3</sup> Colors
texVMDS	3 Templates (2D)	Non-Linear	Fixed (1-4)	Moderate	Random	ImageNet textures	Linear	ImageNet crop

Table 1: Summary of data sets and example video sequences. See Appendix B for details.

### 3. Object-Centric Representation Benchmark

To compare the different object-centric representation learning models on their basic perceptual abilities, we use the well-established multi-object tracking (MOT) protocol (Bernardin and Stiefelhagen, 2008). In this section, we describe the data sets and metrics considered in our benchmark, followed by a brief description of the models evaluated.

#### 3.1 Data Sets

Current object-centric models are not capable of modeling complex natural scenes (Burgess et al., 2019; Greff et al., 2019; Lin et al., 2020). Hence, we focus on synthetic data sets that resemble those which state-of-the-art models were designed for. Specifically, we evaluate on four synthetic data sets<sup>1</sup> (see Table 1), which cover multiple levels of visual and motion complexity. Synthetic stimuli enable us to precisely generate challenging scenarios in a controllable manner in order to disentangle sources of difficulty and glean insights on what models specifically struggle with. We design different scenarios that test complexities that would occur in natural videos such as partial or complete occlusion as well as similar object appearances and complex textures.

**Sprites-MOT (SpMOT)**, Table 1 left), as proposed by He et al. (2019), features simple 2D sprites moving linearly on a black background with objects moving in and out of frame during the sequence. **Video-Multi-dSprites (VMDS)**, Table 1 right) is a video data set we generated based on a colored, multi-object version of the dSprites data set (Matthey et al., 2017). Each video contains one to four sprites that move non-linearly and independently of each other with the possibility of partial or full occlusion. Besides the i.i.d. sampled training, validation and test sets of VMDS, we generate seven additional challenge sets that we use to study specific test situations we observed to be challenging, such as guaranteed occlusion, specific object properties, or out-of-distribution appearance variations. **Video Objects Room (VOR)**, Table 1 middle) is a video data set we generated based on the static Objects Room data set (Burgess et al., 2019), which features static objects in a 3D room with a moving camera. **Textured Video-Multi-dSprites (texVMDS)**, Table 3) is based on the generative model of VMDS regarding shapes, number and movement dynamics of the objects, but instead of being uniformly colored the objects feature random texture crops taken from ImageNet (Deng et al., 2009). Additionally, the background is a linearly moving ImageNet crop. For full details on the data sets and their generation, see Appendix B.

<sup>1</sup>Data sets are available at this URL.

### 3.2 Metrics

Our evaluation protocol follows the multi-object tracking (MOT) challenge, a standard and widely-used benchmark for supervised object tracking (Milan et al., 2016). The MOT challenge uses the CLEAR MOT metrics (Bernardin and Stiefelwagen, 2008), which quantitatively evaluate different performance aspects of object detection, tracking and segmentation. To compute these metrics, predictions have to be matched to ground truth. Unlike Bernardin and Stiefelwagen (2008) and Milan et al. (2016), we use binary segmentation masks for this matching instead of bounding boxes, which helps us better understand the models’ segmentation capabilities. We consider an Intersection over Union (IoU) greater than 0.5 as a match (Voigtlaender et al., 2019). We include an ablation experiment to test how this threshold affects the results. The error metrics used are the fraction of **Misses (Miss)**, **ID switches (ID S.)** and **False Positives (FPs)** relative to the number of ground truth masks. In addition, we report the **Mean Squared Error (MSE)** of the reconstructed image outputs summed over image channels and pixels.

To quantify the overall number of failures, we use the **MOT Accuracy (MOTA)**, which measures the fraction of all failure cases compared to the total number of objects present in all frames. A model with 100% MOTA perfectly tracks all objects without any misses, ID switches or false positives. To quantify the segmentation quality, we define **MOT Precision (MOTP)** as the average IoU of segmentation masks of all matches. A model with 100% MOTP perfectly segments all tracked objects, but does not necessarily track all objects. Further, to quantify detection and tracking performance independent of false positives, we measure the **Mostly Detected (MD)** and **Mostly Tracked (MT)** metrics, the fraction of ground truth objects that have been detected and tracked for at least 80% of their lifespan, respectively. If an ID switch occurs, an object is considered detected but not tracked. For full details regarding the matching process and the evaluation metrics, refer to Appendix A.

### 3.3 Models

We consider one color-segmentation baseline and three classes of unsupervised object-centric representation learning models: (1) a spatial attention model with unconstrained latents, ViMON, which is our video extension of MONET (Burgess et al., 2019); (2) spatial transformer-based attention models, TBA (He et al., 2019) and SCALOR (Jiang et al., 2020); (3) a scene mixture model, OP3 (Veerapaneni et al., 2020). At a high-level, these methods share a common structure which is illustrated in Fig. 1a. They decompose an image into a fixed number of *slots* (Burgess et al., 2019), each of which contains an embedding  $\mathbf{z}_{t,k}$  and a mask  $\mathbf{m}_{t,k}$  of (ideally) a single object. These slots are then combined in a decoding step to reconstruct the image. Below, we briefly describe each method. Appendix C provides a detailed explanation of the methods in a unified mathematical framework.

As a simple baseline, we evaluate the performance of **k-Means** clustering on the RGB pixel values on a per frame basis and tracking performance by frame-to-frame matching based on the IoU of the segmentation masks using Hungarian matching (Kuhn, 1955). For full details refer to Appendix D.1.

**Video MONet (ViMON)** is our video extension of MONET (Burgess et al., 2019). MONET recurrently decomposes a static scene into slots, using an attention network to sequentially extract attention masks  $\mathbf{m}_k \in [0, 1]^{H \times W}$  of individual objects  $k$ . A Variational

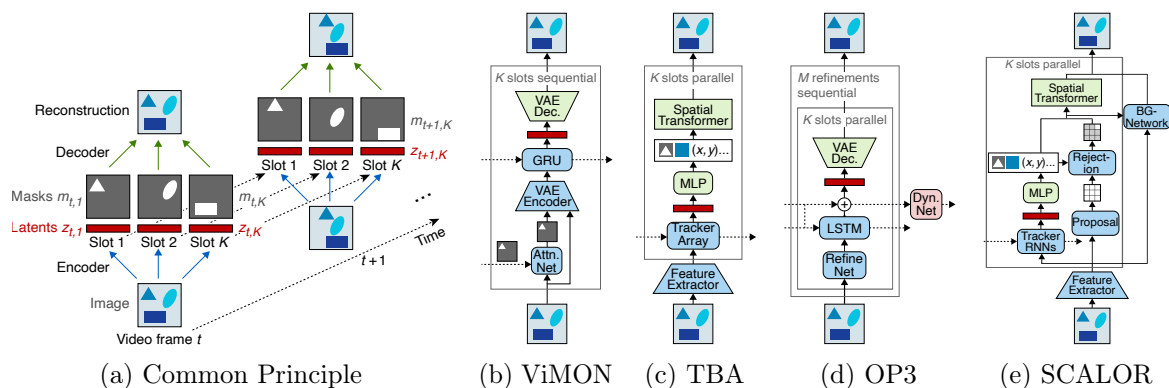


Figure 1: Common principles of all models: Decomposition of an image into a fixed number of slots, each of which contains an embedding  $z_{t,k}$  and a mask  $m_{t,k}$  of (ideally) a single object. Dotted lines: temporal connections. Solid lines: information flow within one frame.

Autoencoder (VAE) (Kingma and Welling, 2014) encodes each slot into a latent representation  $\mathbf{z}_k \in \mathbb{R}^L$  of the corresponding object. We use MONET as a simple frame-by-frame baseline for detection and segmentation that does not employ temporal information. ViMON accumulates evidence about the objects over time to maintain a consistent object-slot assignment throughout the video. This is achieved by (1) seeding the attention network the predicted mask  $\hat{\mathbf{m}}_{t,k} \in [0, 1]^{H \times W}$  from the previous time step and (2) introducing a gated recurrent unit (GRU) (Cho et al., 2014), which aggregates information over time for each slot separately, enabling it to encode motion information. For full details on MONET and ViMON, as well as information regarding hyperparameter tuning and ablations to provide context for the design decisions, refer to Appendix C.1, C.2 and E.3.

**Tracking-by-Animation (TBA)** (He et al., 2019) is a spatial transformer-based attention model. Frames are encoded by a convolutional feature extractor  $f$  before being passed to a recurrent block  $g$  called Reprioritized Attentive Tracking (RAT). RAT re-weights slot input features based on their cosine similarity with the slots from the previous time step and outputs latent representations for all  $K$  slots in parallel. Each slot latent is further decoded into a mid-level representation  $\mathbf{y}_{t,k}$  consisting of pose and depth parameters, as well as object appearance and shape templates (see Fig. 1c). For rendering, a Spatial Transformer Network (STN) (Jaderberg et al., 2015) is used with an additional occlusion check based on the depth estimate. TBA is trained on frame reconstruction with an additional penalty for large object sizes to encourage compact bounding boxes. TBA can only process scenes with static backgrounds, as it preprocesses sequences using background subtraction (Bloisi and Iocchi, 2012). For full details on TBA as well as information regarding hyperparameter tuning, refer to Appendix C.3.

**Object-centric Perception, Prediction, and Planning (OP3)** (Veerapaneni et al., 2020) extends IODINE (Greff et al., 2019) to operate on videos. IODINE decomposes an image into objects and represents them independently by starting from an initial guess of the segmentation of the entire frame, and subsequently iteratively refines it using the refinement network  $f$  (Marino et al., 2018). In each refinement step  $m$ , the image is represented by  $K$  slots with latent representations  $\mathbf{z}_{m,k}$ . OP3 applies IODINE to each frame  $x_t$  to extract latent representations  $\mathbf{z}_{t,m,k}$ , which are subsequently processed by a dynamics network  $d$  (see

BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

Model	MOTA $\uparrow$	MOTP $\uparrow$	MD $\uparrow$	MT $\uparrow$	Match $\uparrow$	Miss $\downarrow$	ID S. $\downarrow$	FPS $\downarrow$	MSE $\downarrow$
<b>SpMOT</b>									
K-MEANS	36.2 $\pm$ 0.0	77.6 $\pm$ 0.0	76.8 $\pm$ 0.0	76.3 $\pm$ 0.0	80.5 $\pm$ 0.0	19.3 $\pm$ 0.0	<b>0.2</b> $\pm$ 0.0	44.3 $\pm$ 0.0	-
MONET	70.2 $\pm$ 0.8	89.6 $\pm$ 1.0	92.4 $\pm$ 0.6	50.4 $\pm$ 2.4	75.3 $\pm$ 1.3	4.4 $\pm$ 0.4	20.3 $\pm$ 1.6	5.1 $\pm$ 0.5	13.0 $\pm$ 2.0
ViMON	92.9 $\pm$ 0.2	<b>91.8</b> $\pm$ 0.2	87.7 $\pm$ 0.8	87.2 $\pm$ 0.8	95.0 $\pm$ 0.2	4.8 $\pm$ 0.2	<b>0.2</b> $\pm$ 0.0	2.1 $\pm$ 0.1	11.1 $\pm$ 0.6
TBA	79.7 $\pm$ 15.0	71.2 $\pm$ 0.3	83.4 $\pm$ 9.7	80.0 $\pm$ 13.6	87.8 $\pm$ 9.0	9.6 $\pm$ 6.0	2.6 $\pm$ 3.0	8.1 $\pm$ 6.0	11.9 $\pm$ 1.9
OP3	89.1 $\pm$ 5.1	78.4 $\pm$ 2.4	92.4 $\pm$ 4.0	91.8 $\pm$ 3.8	<b>95.9</b> $\pm$ 2.2	3.7 $\pm$ 2.2	0.4 $\pm$ 0.0	6.8 $\pm$ 2.9	13.3 $\pm$ 11.9
SCALOR	<b>94.9</b> $\pm$ 0.5	80.2 $\pm$ 0.1	<b>96.4</b> $\pm$ 0.1	<b>93.2</b> $\pm$ 0.7	<b>95.9</b> $\pm$ 0.4	<b>2.4</b> $\pm$ 0.0	1.7 $\pm$ 0.4	<b>1.0</b> $\pm$ 0.1	<b>3.4</b> $\pm$ 0.1
<b>VOR</b>									
K-MEANS	-38.0 $\pm$ 0.1	76.5 $\pm$ 0.0	69.9 $\pm$ 0.1	62.9 $\pm$ 0.1	72.7 $\pm$ 0.1	22.1 $\pm$ 0.0	5.2 $\pm$ 0.0	110.7 $\pm$ 0.1	-
MONET	37.0 $\pm$ 6.8	81.7 $\pm$ 0.5	76.9 $\pm$ 2.2	37.3 $\pm$ 7.8	64.4 $\pm$ 5.0	15.8 $\pm$ 1.6	19.8 $\pm$ 3.5	27.4 $\pm$ 2.3	12.2 $\pm$ 1.4
ViMON	<b>89.0</b> $\pm$ 0.0	<b>89.5</b> $\pm$ 0.5	<b>90.4</b> $\pm$ 0.5	<b>90.0</b> $\pm$ 0.4	<b>93.2</b> $\pm$ 0.4	<b>6.5</b> $\pm$ 0.4	0.3 $\pm$ 0.0	4.2 $\pm$ 0.4	6.4 $\pm$ 0.6
OP3	65.4 $\pm$ 0.6	89.0 $\pm$ 0.6	88.0 $\pm$ 0.6	85.4 $\pm$ 0.5	90.7 $\pm$ 0.3	8.2 $\pm$ 0.4	1.1 $\pm$ 0.2	25.3 $\pm$ 0.6	<b>3.0</b> $\pm$ 0.1
SCALOR	74.6 $\pm$ 0.4	86.0 $\pm$ 0.2	76.0 $\pm$ 0.4	75.9 $\pm$ 0.4	77.9 $\pm$ 0.4	22.1 $\pm$ 0.4	<b>0.0</b> $\pm$ 0.0	<b>3.3</b> $\pm$ 0.2	6.4 $\pm$ 0.1
<b>VMDS</b>									
K-MEANS	-3.3 $\pm$ 0.0	89.8 $\pm$ 0.0	<b>98.3</b> $\pm$ 0.0	93.3 $\pm$ 0.1	96.4 $\pm$ 0.0	<b>1.0</b> $\pm$ 0.0	2.6 $\pm$ 0.0	99.7 $\pm$ 0.0	-
MONET	49.4 $\pm$ 3.6	78.6 $\pm$ 1.8	74.2 $\pm$ 1.7	35.7 $\pm$ 0.8	66.7 $\pm$ 0.7	13.6 $\pm$ 1.0	19.7 $\pm$ 0.6	17.2 $\pm$ 3.1	22.2 $\pm$ 2.2
ViMON	86.8 $\pm$ 0.3	86.8 $\pm$ 0.0	86.2 $\pm$ 0.3	85.0 $\pm$ 0.3	92.3 $\pm$ 0.2	7.0 $\pm$ 0.2	0.7 $\pm$ 0.0	5.5 $\pm$ 0.1	10.7 $\pm$ 0.1
TBA	54.5 $\pm$ 12.1	75.0 $\pm$ 0.9	62.9 $\pm$ 5.9	58.3 $\pm$ 6.1	75.9 $\pm$ 4.3	21.0 $\pm$ 4.2	3.2 $\pm$ 0.3	21.4 $\pm$ 7.8	28.1 $\pm$ 2.0
OP3	<b>91.7</b> $\pm$ 1.7	<b>93.6</b> $\pm$ 0.4	96.8 $\pm$ 0.5	<b>96.3</b> $\pm$ 0.4	<b>97.8</b> $\pm$ 0.1	2.0 $\pm$ 0.1	<b>0.2</b> $\pm$ 0.0	6.1 $\pm$ 1.5	<b>4.3</b> $\pm$ 0.2
SCALOR	74.1 $\pm$ 1.2	87.6 $\pm$ 0.4	67.9 $\pm$ 1.1	66.7 $\pm$ 1.1	78.4 $\pm$ 1.0	20.7 $\pm$ 1.0	0.8 $\pm$ 0.0	<b>4.4</b> $\pm$ 0.4	14.0 $\pm$ 0.1

Table 2: Analysis of SOTA object-centric representation learning models for MOT. Results shown as mean  $\pm$  standard deviation of three runs with different random training seeds.

Fig. 1d), which models both the individual dynamics of each slot  $k$  as well as the pairwise interaction between all combinations of slots, aggregating them into a prediction of the posterior parameters for the next time step  $t + 1$  for each slot  $k$ . For full details on IODINE and OP3, refer to Appendix C.4 and C.5, respectively.

**SCALable Object-oriented Representation (SCALOR)** (Jiang et al., 2020) is a spatial transformer-based model that factors scenes into background and multiple foreground objects, which are tracked throughout the sequence. Frames are encoded using a convolutional LSTM  $f$ . In the proposal-rejection phase, the current frame  $t$  is divided into  $H \times W$  grid cells. For each grid cell a object latent variable  $\mathbf{z}_{t,h,w}$  is proposed, that is factored into existence, pose, depth and appearance parameters. Subsequently, proposed objects that significantly overlap with a propagated object are rejected. In the propagation phase, per object GRUs are updated for all objects present in the scene. Additionally, SCALOR has a background module to encode the background and its dynamics. Frame reconstructions are rendered using a background decoder and foreground STNs for object masks and appearance. For full details on SCALOR as well as information regarding hyperparameter tuning, refer to Appendix C.6.

## 4. Results

We start with a summary of our overall results across the three data sets and four models (Table 2) before analyzing more specific challenging scenarios using variants of the VMDS data set.

A simple color segmentation algorithm is not sufficient to solve tracking even in simplistic data sets (Table 2), in which color is a good predictor for object identity. While  $\kappa$ -MEANS does well at detecting objects, on VMDS even outperforming the other models, its overall tracking performance is significantly worse than the other models, mainly due to a high number of false positives and in SpMOT and VOR also due to a high number of misses, which occur in these two data sets due to multiple objects often having the same color, which a color segmentation algorithm is by definition not able to separate. In the following, we will therefore focus on analyzing the object-centric methods.

We first ask whether tracking could emerge automatically in an image-based model like MONET, which may produce consistent slot assignments through its learned object-slot assignment. This is not the case: MONET exhibits poor tracking performance (Table 2). While MONET correctly finds and segments objects, it does not assign them to consistent slots over time (Fig. E.4). In the following, we will thus focus on the video models: ViMON, TBA, OP3 and SCALOR.

**SpMOT.** All models perform tracking well on SpMOT with the exception of one training run of TBA with poor results leading to high standard deviation (cp. best TBA model: 89.8% MT; Table E.1). SCALOR outperforms the other models on the detection and tracking metrics MD and MT, while ViMON exhibits the highest MOTP, highlighting its better segmentation performance on SpMOT.

**VOR.** TBA is not applicable to VOR due to the dynamic background which cannot be resolved using background subtraction. ViMON and OP3 show similarly good performance on detection (MD) and segmentation (MOTP), while ViMON outperforms OP3 on the tracking metrics MOTA and MT. OP3 accumulates a high number of false positives leading to a low MOTA due to the splitting of objects into multiple masks as well as randomly segmenting small parts of the background (Fig. E.5). In contrast, SCALOR has almost no false positives or ID switches, but accumulates a high number of misses leading to a poor MOTA. It often segments two objects as one that are occluding each other in the first frame, which is common in VOR due to the geometry of its scenes (Fig. F.17, last row).

**VMDS.** OP3 outperforms the other models on VMDS, on which TBA performs poorly, followed by SCALOR, which again accumulates a high number of misses. We will analyze the models on VMDS qualitatively and quantitatively in more detail in the following.

#### 4.1 Impact of IoU Threshold for Matching

To examine whether the IoU threshold used for matching object predictions to ground truth impacts the ordering of the models, we compute all metrics with regard to the IoU thresholds [0.1 .. 0.9] in steps of 0.1 on the VMDS test set. The ordering of the models with regard to their MOTA is consistent over all matching thresholds except for a threshold of 0.9, but the performance difference between the models increases with higher thresholds (Fig. 2a). Thus, we will perform all following experiments with an IoU threshold of 0.5. For all metrics, see Fig. E.6 in Appendix.

#### 4.2 Accumulation of Evidence over Time

Recognition and tracking of objects should improve if models can exploit prior knowledge about the objects in the scene from previous video frames. To test whether the models

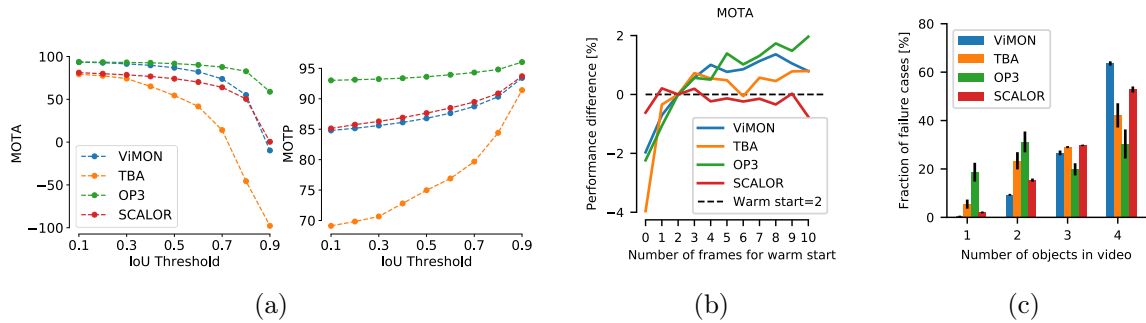


Figure 2: a) MOTA and MOTP performance of models shown for different IoU threshold used for matching model predictions to ground truth objects. Note that MOTA can become negative, since the number of FPs is unbounded. b) MOTA on frames 11–20 of the VMDS test set with warm starts of 1–10 frames (0 = no warm start). Difference to performance of warm start = 2 shown. c) Distribution of failure cases dependent on number of objects in VMDS videos. Mean of three training runs. Error bars: standard deviation.

exploit such knowledge, we evaluate their MOTA performance on VMDS after warm starting with up to 10 frames which are not included in evaluation (Fig. 2b). Note that the models were trained on sequences of length 10, but are run for 20 frames in the case of a warm start of 10 frames. The performance of ViMON improves with longer warm starts, showing that the GRU accumulates evidence over time. TBA, in contrast, does not use temporal information beyond 2–3 frames, while SCALOR’s performance slightly drops after 3 frames. OP3 appears to most strongly rely on past information and is able to integrate information over longer time scales: its performance does not even saturate with a warm start of 10 frames. However, the effect for all models is rather small.

### 4.3 Challenging Scenarios for Different Models

*The number of objects in the sequence* matters for ViMON, TBA and SCALOR: more objects increase the number of failure cases (Fig. 2c). In contrast, OP3 does not exhibit this pattern: it accumulates a higher number of false positives (FPs) in videos with fewer (only one or two) objects (Fig. E.1), as it tends to split objects into multiple slots if fewer objects than slots are present.

*Occlusion* leads to failure cases for all models (Fig. 3a–b). Partial occlusion can lead to splitting of objects into multiple slots (Fig. 3a). Objects that reappear after full occlusion are often missed when only a small part of them is visible (Fig. 3a). In particular, SCALOR tends to segment two objects as one when they overlap while entering the scene, leading to a high number of misses.

*Color* of the object is important. TBA often misses dark objects (Fig. 3b). In contrast, ViMON, OP3 and SCALOR struggle with scenes that feature objects of similar colors as well as objects that have similar colors to the background (Fig. 3c,e).

*False positives* are more prevalent for OP3 and TBA than for ViMON and SCALOR (Table 2). FPs of OP3 are due to objects split in multiple masks (Fig. 3a) and random

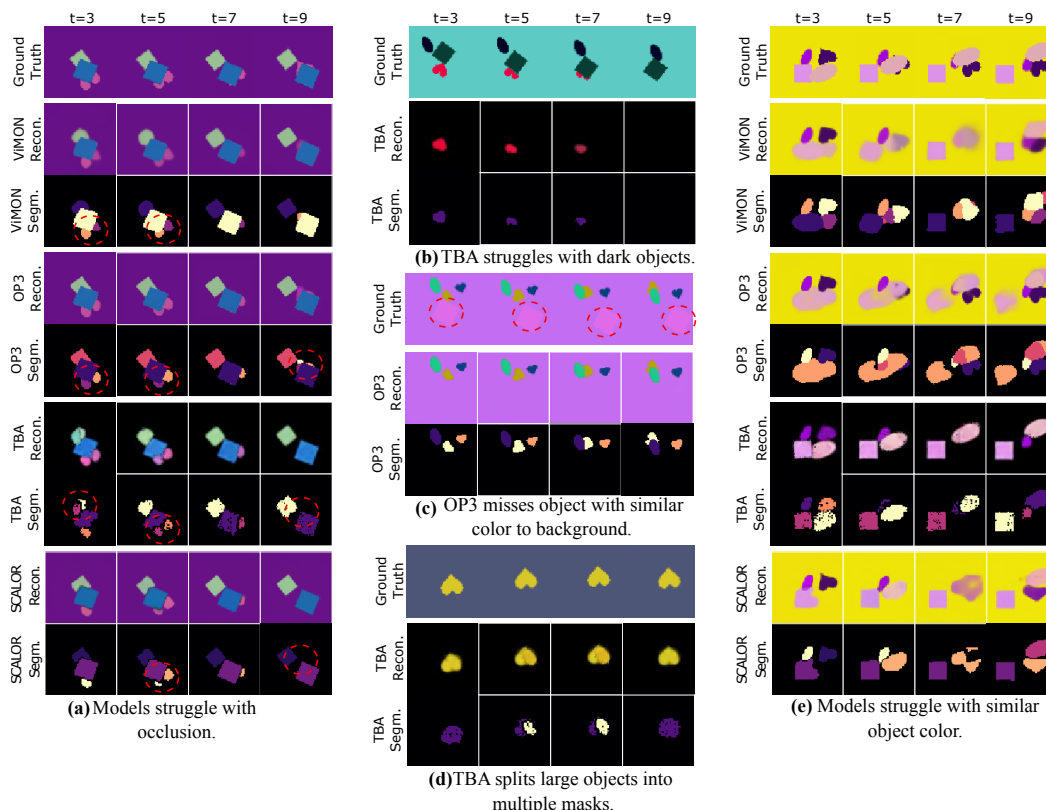


Figure 3: Example failure cases for all models on VMDS. Segmentation masks are binarized and color-coded to signify slot assignment.

small parts of the background being individually segmented (Fig. 3e), while TBA tends to compose larger objects using multiple smaller, simpler components (Fig. 3d).

#### 4.4 Performance on Challenge Sets

Based on the challenging scenarios identified above, we design multiple ‘challenge sets’: videos featuring (1) heavy occlusion, (2) objects with same colors, (3) only small objects and (4) only large objects (Fig. 4a, top). For details, see Appendix B.1.1.

Occlusion reduces performance of all models compared with the i.i.d. sampled VMDS test set, albeit to different degrees (Fig. 4a; for absolute performance see Table E.3). OP3 is more robust to occlusion than the other models.

Tracking objects with the same color is challenging for all models (Fig. 4a). In particular, OP3 appears to rely on object color as a way to separate objects.

OP3, VIMON and SCALOR are not sensitive to object size (Fig. 4a). They exhibit only slightly decreased performance on the large objects test set, presumably because large objects cause more occlusion (Fig. 4a). TBA shows increased performance on small objects but performs poorly on the large objects set.



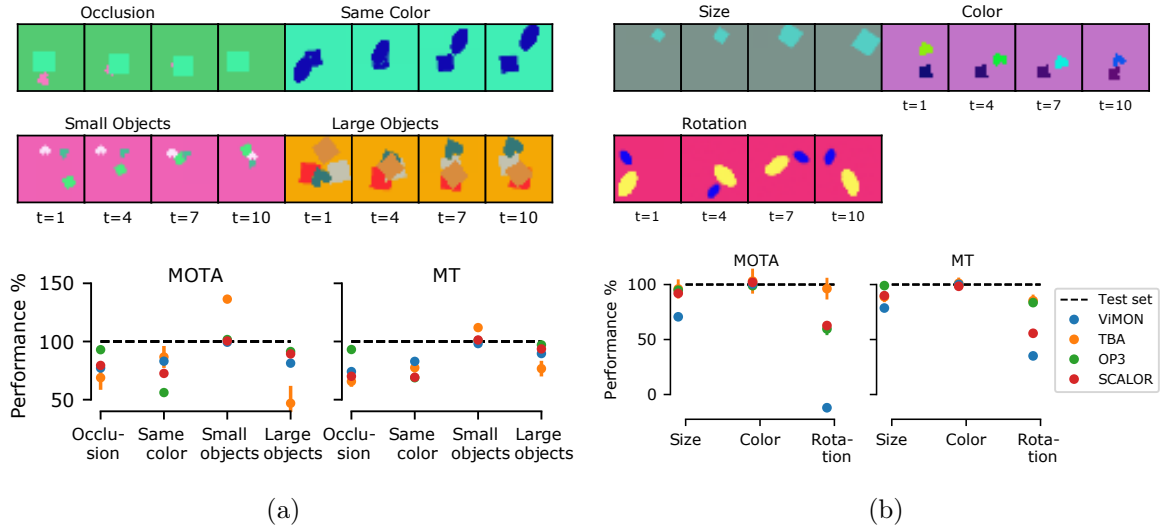


Figure 4: a) Performance on challenge sets relative to performance on VMDS test set (100%) and example video sequences. b) Performance on out-of-distribution sets relative to VMDS test set (100%) and example video sequences. Note that MOTA can become negative, since the number of FPs is unbounded.

#### 4.5 Performance on Out-of-distribution Test Sets

Next, we assess generalization to out-of-distribution (o.o.d.) changes in object appearance that are not encountered during training. In the training set of VMDS, object color, size and orientation are constant throughout a video. To test o.o.d. generalization, we evaluate models trained on VMDS on three data sets that feature unseen object transformations (Fig. 4b and Table E.4): continuous changes in object **color** or **size** as well as continuous **rotation** around the object’s centroid while moving. For details, see Appendix B.1.2.

Continuous changes in object size do not pose a serious problem to TBA, OP3 and SCALOR, while ViMON’s performance drops (Fig. 4b). Surprisingly, continuous color changes of objects do not impact the performance of any model. Tracking performance of ViMON drops significantly for rotated objects, while OP3 and SCALOR are affected less. TBA’s tracking performance is not as strongly influenced by object rotation (for absolute values, see Table E.4).

#### 4.6 Stability of Training and Runtime

TBA and SCALOR train faster and require less memory than OP3 and ViMON (see Table E.5 for details). However, two training runs converged to suboptimal minima for TBA (note the high variance in Table 2 and see Table E.1 and Table E.2 for individual runs). Training OP3 is sensitive to the learning rate and unstable, eventually diverging in all experiments. Interestingly, it often reached its best performance directly prior to divergence. ViMON and TBA are less sensitive to hyperparameter settings in our experiments. For a more detailed analysis of the runtime, see Appendix E.2.

Model	MOTA $\uparrow$	MOTP $\uparrow$	MD $\uparrow$	MT $\uparrow$	Match $\uparrow$	Miss $\downarrow$	ID S. $\downarrow$	FPS $\downarrow$	MSE $\downarrow$
<b>texVMDS</b>									
K-MEANS	-99.5 $\pm$ 0.0	<b>76.4</b> $\pm$ 0.0	<b>25.3</b> $\pm$ 0.0	<b>24.3</b> $\pm$ 0.1	30.3 $\pm$ 0.0	69.2 $\pm$ 0.0	0.5 $\pm$ 0.0	129.8 $\pm$ 0.0	-
MONET	<b>-73.3</b> $\pm$ 5.5	67.7 $\pm$ 1.1	16.0 $\pm$ 3.4	12.3 $\pm$ 3.1	24.7 $\pm$ 4.7	73.1 $\pm$ 5.1	2.2 $\pm$ 0.8	<b>98.0</b> $\pm$ 1.7	200.5 $\pm$ 5.7
VIMON	-85.5 $\pm$ 2.8	69.0 $\pm$ 0.6	24.2 $\pm$ 1.3	23.8 $\pm$ 1.4	<b>34.7</b> $\pm$ 1.7	<b>65.0</b> $\pm$ 1.7	0.3 $\pm$ 0.0	120.2 $\pm$ 2.5	171.4 $\pm$ 3.3
OP3	-110.4 $\pm$ 4.3	70.6 $\pm$ 0.6	16.5 $\pm$ 5.1	16.2 $\pm$ 5.0	22.9 $\pm$ 6.6	76.9 $\pm$ 6.7	<b>0.2</b> $\pm$ 0.1	133.4 $\pm$ 2.9	<b>132.8</b> $\pm$ 16.2
SCALOR	-99.2 $\pm$ 11.7	74.0 $\pm$ 0.5	6.5 $\pm$ 0.6	6.3 $\pm$ 0.6	12.3 $\pm$ 0.4	87.5 $\pm$ 0.4	<b>0.2</b> $\pm$ 0.0	111.5 $\pm$ 11.4	<b>133.7</b> $\pm$ 11.1


  


Table 3: MOT performance on texVMDS and example video sequences. Results shown as mean  $\pm$  standard deviation of three runs with different random training seeds. Note that MOTA can become negative, since the number of FPS is unbounded.

#### 4.7 Towards natural data

Object-centric methods have mostly been shown to work on synthetic data sets and seem to rely to a certain extent on color segmentation. To quantify how these methods fare with more realistic images, we created a fourth data set that keeps the simple scene composition of VMDS (1–4 object and few shapes), but has a higher level of visual complexity. To this end, we generated moving object masks as in VMDS, but filled each object and the background with the content of a different image taken from ImageNet (Table 3, bottom). We refer to this data set as *texVMDS*. Humans can easily detect objects in these video through motion cues,<sup>2</sup> while the evaluated models fail at detecting and tracking objects in this data set (Table 3).

Most of the statistics of the objects in texVMDS are identical to those of VMDS (shape, size, velocity, number of objects); the main difference is the visual complexity of the object and background texture. We therefore tune the dimensionality of the latent space for MONET and SCALOR, to account for the higher visual complexity, while keeping all other hyperparameters identical to VMDS. OP3 already has a significantly larger latent dimensionality and uses that for synthetic as well as natural data in their publication. Therefore we keep these hyperparameters. For full details see Appendix D.

TBA is not applicable to texVMDS due to the dynamic background which cannot be resolved using background subtraction. All other models perform poorly. They fail to learn the object boundaries despite the fact that there are only four different object shapes and objects and background move independently, which makes them easy to tell apart for humans. Moreover, they generate large amounts of false positives, often using multiple slots for the background or an object, with the split between slots mostly following color changes (see Fig. F.8, Fig. F.14, Fig. F.18).

## 5. Discussion

Our experimental results provide insights into the inductive biases and failure cases of object-centric models that were not apparent from their original publications. Despite

<sup>2</sup>See <https://eckerlab.org/code/weis2021/> for example videos.

the positive results shown in each of the papers for the evaluated methods, a controlled, systematic analysis demonstrates that they do not convincingly succeed at tracking, which is fundamentally what object-centric video methods should enable.

TBA has a significantly lower MOTP than the other models on all datasets, suggesting that the simple rendering-based decoder using a fixed **template** might be less suitable to generate accurate segmentation masks (see also Fig. F.10 and Fig. F.9) compared to the VAE-based approaches of ViMON, OP3 and SCALOR.

Handling **occlusion** of objects during the video is a key property object-centric representations should be capable of. Qualitatively and quantitatively, OP3 is more robust to occlusion than the other models, suggesting that its dynamics network which models interactions between objects is currently most successful at modeling occluded objects. Surprisingly, TBA and SCALOR, which explicitly encode depth, do not handle occlusion more gracefully than ViMON, whose simpler architecture has no explicit way of dealing with depth. Moving forward, occlusion handling is a key component that object-centric video models need to master. It can be addressed by either equipping the model with a potent interaction module, that takes pairwise interaction between objects (including occlusion) into account, similar to OP3’s dynamics model, or ensuring that the depth reasoning of the models works as intended, which may be preferable, as explained below.

All models struggle with detecting objects that have similar **color** as the background (for TBA: dark objects, since background is removed and set to black in a pre-processing step). Color is a reliable cue to identify objects in these datasets. However, the auto-encoding objective incurs little extra loss for missing objects with similar color as the background and, thus, the models appear to not to learn to properly reconstruct them. In order to scale to data with more visual complexity, one might want to replace the pixel-wise reconstruction with for instance a loss based in feature space in order to focus more on reconstructing semantic content rather than high-frequency texture, as is done when using perceptual loss functions (Gatys et al., 2015; Hou et al., 2017) or by using contrastive learning (Kipf et al., 2020). Furthermore, the models—particularly so OP3—struggle with separating objects of similar colors from each other. This result hints at a mismatch between the intuitions motivating these models and what the models actually learn: it should be more efficient in terms of the complexity of the latent representation to decompose two objects—even of similar colors—into two masks with simple shapes, rather than encoding the more complicated shape of two objects simultaneously in one slot. However, since none of the models handle occlusion with amodal segmentation masks (i. e. including the occluded portion of the object) successfully, they learn to encode overly complex (modal) mask shapes. As a consequence, they tend to merge similarly colored objects into one slot. This result suggests that resolving the issues surrounding the depth reasoning in combination with amodal segmentation masks would enable much more compact latents and could also resolve the merging of similarly colored objects.

A major difference between models is the **spatial transformer** based model formulation of TBA and SCALOR, compared to ViMON and OP3, which operate on image-sized masks. The parallel processing of objects and the processing of smaller bounding boxes renders training TBA and SCALOR to be significantly faster and more memory efficient, enabling them to scale to a larger number of objects. On the downside, the spatial transformer introduces its own complications. TBA depends strongly on its prior on object size and

performs well only when this prior fits the data well as well as when the data contains little variation in object sizes, as in SpMOT (Table 2). However, it is not able to handle VMDS and its larger variation in object sizes and shapes. SCALOR performs tracking well in scenes where objects are clearly separated, but struggles to separate objects that partially occlude each other when entering the scene. This difficulty is caused by its discovery mechanism which can propose at most one bounding box per grid cell, leading to a high number of misses on data sets which feature significant occlusion (VOR and VMDS). Unfortunately, simply increasing the number of proposals does not provide a simple solution, as SCALOR’s performance is sensitive to properly tweaking the number of proposals.

Choosing a class of models is therefore dependent on the data set one wants to apply it to as well as the computational resources at one’s disposal. Data sets that feature a high number of objects ( $>10$ ) that are well separated from each other make a method like SCALOR, which can process objects in parallel, advisable. On data sets with a lower number of objects per scene which feature heavy occlusion, methods like OP3 and ViMON will likely achieve better results, but require a high computational budget for training.

In conclusion, our analysis shows that none of the models solve the basic challenges of tracking even for relatively simple synthetic data sets. Future work should focus on developing robust mechanisms for reliably handling depth and occlusion, additionally combining the transformer-based efficiency of TBA and SCALOR with the stable training of ViMON and the interaction model of OP3. The key open challenges for scaling these models to natural videos include their computational inefficiency, complex training dynamics, as well as over-dependence on simple appearance cues like color.

## Acknowledgments

This work was supported by the German Federal Ministry of Education and Research through the Tübingen AI Center (FKZ: 01IS18039A, 01IS18039B); Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via grant EC 479/1-1 to A.S.E; and Collaborative Research Center (Projektnummer 276693517 – SFB 1233: Robust Vision). We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Marissa A. Weis, Kashyap Chitta, and Yash Sharma. The authors would also like to thank Eshed Ohn-Bar for helpful discussions, as well as Rishi Veerapaneni, John D. Co-Reyes, and Michael Chang particularly with regards to applying OP3 to our experimental setting.

## Appendices

In the appendix, we first discuss the metrics used (Section A) and describe the data generation process (Section B). We then describe the methods MONET, ViMON, TBA, IODINE, OP3 and SCALOR (Section C). Section D contains information regarding the implementation details and training protocols. Finally, we provide additional qualitative and quantitative experimental results in Section E.

### Appendix A. Evaluation Protocol Details

We quantitatively evaluate all models on three data sets using the standard CLEAR MOT metrics (Bernardin and Stiefelhagen, 2008). Our evaluation protocol is adapted from the multi-object tracking (MOT) challenge (Milan et al., 2016), a standard computer vision benchmark for supervised object tracking. In particular, we focus on the metrics provided by the py-motmetrics package<sup>3</sup>.

#### A.1 Mapping

In each frame, object predictions of each model in the form of binary segmentation masks are mapped to the ground truth object segmentation masks. We require that each pixel is uniquely assigned to at most one object in the ground truth and the predictions, respectively. Matching is based on the Intersection over Union (IoU) between the predictions and the ground truth masks (Voigtlaender et al., 2019). A valid correspondence between prediction and object has to exceed a threshold in IoU of 0.5. Predictions that are not mapped to any ground truth mask are classified as false positives (FPs). Ground truth objects that are not matched to any prediction are classified as misses. Following (Bernardin and Stiefelhagen, 2008), ground truth objects that are mapped to two different hypothesis IDs in subsequent frames are classified as ID switches for that frame.

#### A.2 MOT Metrics

**MOT Accuracy (MOTA)** measures the fraction of all failure cases, i.e. false positives (FPs), misses and ID switches compared to total number of objects present in all frames. **MOT Precision (MOTP)** measures the total accuracy in position for matched object hypothesis pairs, relative to total number of matches made. We use percentage Intersection over Union (IoU) of segmentation masks as the accuracy in position for each match. **Mostly Tracked (MT)** is the ratio of ground truth objects that have been tracked for at least 80% of their lifespan.(i.e. 80% of the frames in which they are visible). MT as implemented by py-motmetrics counts trajectories of objects as correctly tracked even if ID switches occur. We use a strictly more difficult definition of MT that counts trajectories with ID switches as correctly detected but not correctly tracked. Consequently, we add the **Mostly Detected (MD)** measure which does not penalize ID switches. **Match, Miss, ID Switches (ID S.)**

---

<sup>3</sup><https://pypi.org/project/motmetrics/>

and **FPs** are reported as the fraction of the number of occurrences divided by the total number of object occurrences.

$$\text{MOTA} = 1 - \frac{\sum_{t=1}^T M_t + \text{FP}_t + \text{IDS}_t}{\sum_{t=1}^T O_t} \quad (1)$$

where  $M_t$ ,  $\text{FP}_t$ , and  $\text{IDS}_t$  are the number of misses, false positives and ID switches, respectively, for time  $t$ , and  $O_t$  is the number of objects present in frame  $t$ . Note that MOTA can become negative, since the number of FPs is unbounded.

$$\text{MOTP} = \frac{\sum_{t=1}^T \sum_{i=1}^I d_t^i}{\sum_{t=1}^T c_t} \quad (2)$$

where  $d_t^i$  is the total accuracy in position for the  $i^{\text{th}}$  matched object-hypothesis pair measured in IoU between the respective segmentation masks and  $c_t$  is the number of matches made in frame  $t$ .

Note that we exclude the background masks for ViMON and OP3 before evaluating tracking based on IoU. The Video Object Room (VOR) data set can contain up to three background segments, namely the floor and up to two wall segments. In order to exclude all background slots regardless of whether the model segments the background as one or as multiple masks, we remove all masks before the tracking evaluation that have an IoU of more than 0.2 with one of the ground truth background masks; we empirically tested that this heuristic is successful in removing background masks regardless of whether the models segments it as one or as three separate ones.

## Appendix B. Data Set Generation Details

### B.1 Video Multi-dSprites (VMDS)

The Multi-DSprites Video data set consists of 10-frame video sequences of  $64 \times 64$  RGB images with multiple moving sprites per video. In order to test temporal aggregation properties of the models, the test set contains 20 frame-long sequences. Each video contains one to four sprites following the data set proposed in Burgess et al. (2019) that move independently of each other and might partially or fully occlude one another. The sprites are sampled uniformly from the dSprites data set (Matthey et al., 2017) and colored with a random RGB color. The background is uniformly colored with a random RGB color. Random trajectories are sampled per object by drawing  $x$  and  $y$  coordinates from a Gaussian process with squared exponential covariance kernel  $\text{cov}[x_s, x_t] = \exp[-(x_s - x_t)^2 / (2\tau^2)]$  and time constant  $\tau = 10$  frames, and then shifted by an initial  $(x, y)$ -position of the sprite centroid, which is uniformly sampled from  $[10, 54]$  to ensure that the object is within the image boundaries. Trajectories that leave these boundaries are rejected. In occlusion scenarios, larger objects are always in front of smaller objects to disambiguate prediction of occlusion. The training set consists of 10,000 examples whereas the validation set as well as the test set contain 1,000 examples each. Additionally, we generated four challenge sets and three out-of-distribution test sets for VMDS that contain specifically challenging scenarios. Each test set consists of 1,000 videos of length 10 frames, which we describe in the following.

## B.1.1 VMDS CHALLENGE SETS

**Occlusion test set.** In each video, one or more objects are heavily occluded and thus often are not visible at all for a few frames. This is ensured by sampling object trajectories that cross path, i.e. at least in one video frame, two objects are centered on the same pixel. The time step and spatial position of occlusion is sampled randomly. Object trajectories are sampled independently as described above and then shifted such that they are at the sampled position of occlusion at time  $t$ . Videos contain two to four sprites (Fig. E.3), since at least two objects are necessary for occlusion.

**Small Objects.** Videos contain one to four sprites with all sprites being of the smallest size present in the original dSprites (Matthey et al., 2017) data set (Fig. E.3). Other than that, it follows the generation process of the regular training and test set.

**Large Objects.** Videos contain one to four sprites with all sprites being of the largest size present in the original dSprites (Matthey et al., 2017) data set (Fig. E.3). Other than that, it follows the generation process of the regular training and test set.

**Same Color.** Videos contain two to four sprites which are identically colored with a randomly chosen color. Other than that, it follows the generation process of the regular training and test set (Fig. E.3).

## B.1.2 VMDS OUT-OF-DISTRIBUTION TEST SETS

**Rotation test set.** Sprites rotate around their centroid while moving. The amount of rotation between two video frames is uniformly sampled between 5 and 40 degrees, and is constant for each object over the course of the video. Direction of rotation is chosen randomly. Rotation is not included as a transformation in the training set (Fig. E.4).

**Color change test set.** Sprites change their color gradually during the course of the video. The initial hue of the color is chosen randomly as well as the direction and amount of change between two frames, which stays the same for each object over the course of the video. Saturation and value of the color are kept constant. Color changes are not part of the training set (Fig. E.4).

**Size change test set.** Sprites change their size gradually during the course of the video. The original dSprites data set (Matthey et al., 2017) contains six different sizes per object. For each object, its size is sampled as either the smallest or largest in the first frame as well as a random point in time, at which it starts changing its size. At this point in time, it will either become larger or smaller, respectively, increasing or decreasing each frame to the next larger or smaller size present in the original dSprites data set, until the largest or smallest size is reached. Size changes are not part of the training set (Fig. E.4).

## B.1.3 TEXVMDS

The Textured Multi-DSprites Video data set consists of 10-frame video sequences of  $64 \times 64$  RGB images with 1–4 moving sprites per video. The background is a linearly moving crop from a uniformly sampled image (Deng et al., 2009) from the ImageNet Large Scale Visual Recognition Challenge dataset (ILSVRC2012) and the texture of each sprite is similarly cropped from a uniformly sampled ILSVRC2012 image (Deng et al., 2009). The direction of the moving background is uniformly sampled from one of 8 directions (up, down, left,

right, upper-right, upper-left, lower-right, lower-left). Other than that, texVMDS follows the generation process of the regular VDMS.

## B.2 Sprites-MOT (SpMOT)

Sprites-MOT, originally introduced by He et al. (2019), consists of video sequences of length 20 frames. Each frame is a  $128 \times 128$  RGB image. It features multiple sprites moving linearly on a black background. The sprite can have one of four shapes and one of six colors. For more information, refer to the original paper (He et al., 2019). We generate a training set consisting of 9600 examples, validation set of 384 samples and test set of 1,000 examples using the author-provided public codebase<sup>4</sup>. However, instead of using the default setting of 20 frames per sequence, we instead generate sequences of length 10, in order to facilitate comparison to the other data sets in our study which have only 10 frames per sequence.

Frames are downsampled to a resolution of  $64 \times 64$  for training ViMON, OP3 and SCALOR.

## B.3 Video Objects Room (VOR)

We generate a video data set based on the static 3D Objects Room data set (Burgess et al., 2019), with sequences of length 10 frames each at a resolution of  $128 \times 128$ . This data set is rendered with OpenGL using the gym-miniworld<sup>5</sup> reinforcement learning environment. It features a 3D room with up to four static objects placed in one quadrant of the room, and a camera initialized at the diagonally opposite quadrant. The objects are either static cubes or spheres, assigned one of 6 colors and a random orientation on the ground plane of the room. The camera then follows one of five trajectories moving towards the objects, consisting of a small fixed distance translation and optional small fixed angle of rotation each time step. The wall colors and room lighting are randomized, but held constant throughout a sequence. The training set consists of 10,000 sequences whereas the validation set and the test set contain 1,000 sequences each.

Frames are downsampled to a resolution of  $64 \times 64$  for training ViMON, OP3 and SCALOR.

## Appendix C. Methods

In this section we describe the various methods in a common mathematical framework. For details about implementation and training, please refer to Section D.

### C.1 MONET

Multi-Object-Network (MONET) (Burgess et al., 2019) is an object-centric representation model designed for static images. It consists of a recurrent attention network that sequentially extracts attention masks of individual objects and a variational autoencoder (VAE) (Kingma and Welling, 2014) that reconstructs the image region given by the attention mask in each processing step.

<sup>4</sup><https://github.com/zhen-he/tracking-by-animation>

<sup>5</sup><https://github.com/maximecb/gym-miniworld>



**Attention Network:** The attention network is a U-Net (Ronneberger et al., 2015) parameterized by  $\psi$ . At each processing step  $k$ , the attention network receives the full image  $\mathbf{x} \in [0, 1]^{H \times W \times 3}$  as input together with the scope variable  $\mathbf{s}_k \in [0, 1]^{H \times W}$ . The scope  $\mathbf{s}_k$  keeps track of the regions of the image that haven’t been attended to in the previous processing steps and thus remain to be explained. The attention network outputs a soft attention mask  $\mathbf{m}_k \in [0, 1]^{H \times W}$  and the updated scope with the current mask subtracted:

$$\mathbf{m}_k = \mathbf{s}_{k-1} \alpha_\psi(\mathbf{x}, \mathbf{s}_{k-1}), \quad (3)$$

$$\mathbf{s}_{k+1} = \mathbf{s}_k (1 - \alpha_\psi(\mathbf{x}, \mathbf{s}_k)), \quad (4)$$

where  $\alpha_\psi(\mathbf{x}, \mathbf{s}_k) \in [0, 1]^{H \times W}$  is the output of the U-net and  $\mathbf{s}_0 = \mathbf{1}$ . The attention mask for the last slot is given by  $\mathbf{m}_K = \mathbf{s}_{K-1}$  to ensure that the image is fully explained, i.e.  $\sum_{k=1}^K \mathbf{m}_k = \mathbf{1}$ .

**VAE:** The VAE consists of an encoder  $g : [0, 1]^{H \times W \times 3} \times [0, 1]^{H \times W} \rightarrow \mathbb{R}^{L \times 2}$  and a decoder  $h : \mathbb{R}^L \rightarrow [0, 1]^{H \times W \times 3} \times [0, 1]^{H \times W}$  which are two neural networks parameterized by  $\phi$  and  $\theta$ , respectively. The VAE encoder receives as input the full image  $\mathbf{x}$  and the attention mask  $\mathbf{m}_k$  and computes  $(\boldsymbol{\mu}_k, \log \boldsymbol{\sigma}_k)$ , which parameterize the Gaussian latent posterior distribution  $q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{m}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k I)$ . Using the reparametrization trick (Kingma and Welling, 2014),  $\mathbf{z}_k \in \mathbb{R}^L$  is sampled from the latent posterior distribution.  $\mathbf{z}_k$  is decoded by the VAE decoder into a reconstruction of the image component  $\hat{\mathbf{x}}_k \in [0, 1]^{H \times W \times 3}$  and mask logits, which are used to compute the reconstruction of the mask  $\hat{\mathbf{m}}_k \in [0, 1]^{H \times W}$  via a pixelwise softmax across slots. The reconstruction of the whole image is composed by summing over the  $K$  masked reconstructions of the VAE:  $\hat{\mathbf{x}} = \sum_{k=1}^K \hat{\mathbf{m}}_k \odot \hat{\mathbf{x}}_k$ .

**Loss:** MONET is trained end-to-end with the following loss function:

$$\begin{aligned} L(\phi; \theta; \psi; \mathbf{x}) = & -\log \sum_{k=1}^K \mathbf{m}_k p_\theta(\mathbf{x} | \mathbf{z}_k) + \beta D_{\text{KL}}\left(\prod_{k=1}^K q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{m}_k) \parallel p(\mathbf{z})\right) \\ & + \gamma \sum_{k=1}^K D_{\text{KL}}(q_\psi(\mathbf{m}_k | \mathbf{x}) \parallel p_\theta(\mathbf{m}_k | \mathbf{z}_k)), \end{aligned} \quad (5)$$

where  $p_\theta(\mathbf{x} | \mathbf{z}_k)$  is the Gaussian likelihood of the VAE decoder and  $\mathbf{z}_k \in \mathbb{R}^L$  is the latent representation of slot  $k$ .

The first two loss terms are derived from the standard VAE objective, the Evidence Lower Bound (ELBO) (Kingma and Welling, 2014), i.e. the negative log-likelihood of the decoder and the Kullback–Leibler divergence between the unit Gaussian prior  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$  and the latent posterior distribution  $q_\phi(\mathbf{z}_k | \mathbf{x}, \mathbf{m}_k)$  factorized across slots. Notably, the decoder log-likelihood term  $p_\theta(\mathbf{x} | \mathbf{z}_k)$  constrains only the reconstruction within the mask, since it is weighted by the mask  $\mathbf{m}_k$ . Additionally, as a third term, the Kullback–Leibler divergence of the attention mask distribution  $q_\psi(\mathbf{m}_k | \mathbf{x})$  with the VAE mask distribution  $p_\theta(\hat{\mathbf{m}}_k | \mathbf{z}_k)$  is minimized, to encourage the VAE to learn a good reconstruction of the masks.

## C.2 Video MONet

We propose an extension of MONET (Burgess et al., 2019), called Video MONet (VIMON), which accumulates evidence over time about the objects in the scene (Fig. C.1).

VIMON processes a video recurrently by reconstructing one frame at a time and predicting the next frame of the video. The processing of each frame follows a logic similar to MONET with some notable differences. In the following, we use  $t$  to indicate the time step in the video and  $k$  to indicate the processing step within one video frame.

**Attention Network:** The attention network of VIMON outputs an attention mask  $\mathbf{m}_{t,k} \in [0, 1]^{H \times W}$  in each step  $k$  conditioned on the full frame  $\mathbf{x}_t \in [0, 1]^{H \times W \times 3}$ , the scope  $\mathbf{s}_{t,k} \in [0, 1]^{H \times W}$  and additionally the mask  $\hat{\mathbf{m}}_{t,k} \in [0, 1]^{H \times W}$  that was predicted by the VAE in the previous time step, in order to provide it with information about which object it should attend to in this specific slot  $k$ :

$$\mathbf{m}_{t,k} = \mathbf{s}_{t,k-1} \alpha_\psi(\mathbf{x}_t, \mathbf{s}_{t,k-1}, \hat{\mathbf{m}}_{t,k}). \quad (6)$$

**VAE:** The VAE of VIMON consists of an encoder  $g(\mathbf{x}_t, \mathbf{m}_{t,k}; \phi)$  and a decoder  $h(\mathbf{z}_{t,k}; \theta)$ . In contrast to MONET, the encoder in VIMON is followed by a gated recurrent unit (GRU) (Cho et al., 2014) with a separate hidden state  $h_{t,k}$  per slot  $k$ . Thus, the GRU aggregates information over time for each object separately. The GRU outputs  $(\boldsymbol{\mu}_{t,k}, \log \boldsymbol{\sigma}_{t,k})$  which parameterize the Gaussian latent posterior distribution  $q_\phi(\mathbf{z}_{t,k} | \mathbf{x}_t, \mathbf{m}_{t,k})$  where  $\mathbf{z}_{t,k} \in \mathbb{R}^L$  is the latent representation for slot  $k$  at time  $t$ :

$$\mathbf{z}'_{t,k} = g(\mathbf{x}_t, \mathbf{m}_{t,k}; \phi), \quad (7)$$

$$(\boldsymbol{\mu}_{t,k}, \log \boldsymbol{\sigma}_{t,k}), \mathbf{h}_{t,k} = f(\text{GRU}(\mathbf{z}'_{t,k}, \mathbf{h}_{t-1,k})), \quad (8)$$

$$q_\phi(\mathbf{z}_{t,k} | \mathbf{x}_t, \mathbf{m}_{t,k}) = \mathcal{N}(\boldsymbol{\mu}_{t,k}, \boldsymbol{\sigma}_{t,k} I) \quad \forall t, k, \quad (9)$$

where  $g$  is the VAE encoder and  $f$  is a linear layer. The latent representation  $\mathbf{z}_{t,k}$  is sampled from the latent posterior distribution using the reparametrization trick (Kingma and Welling, 2014). Subsequently,  $\mathbf{z}_{t,k}$  is linearly transformed into  $\hat{\mathbf{z}}_{t+1,k}$  via a learned transformation  $\mathbf{A} \in \mathbb{R}^{L \times L}$ :  $\hat{\mathbf{z}}_{t+1,k} = \mathbf{A} \mathbf{z}_{t,k}$  with  $\hat{\mathbf{z}}_{t+1,k}$  being the predicted latent code for the next time step  $t + 1$ . Both  $\mathbf{z}_{t,k}$  and  $\hat{\mathbf{z}}_{t+1,k}$  are decoded by the shared VAE decoder  $h_\theta$  into a reconstruction of the image  $\hat{\mathbf{x}}_{t,k} \in [0, 1]^{H \times W \times 3}$  and a reconstruction of the mask  $\hat{\mathbf{m}}_{t,k} \in [0, 1]^{H \times W}$  as well as  $\hat{\mathbf{x}}_{t+1,k}$  and  $\hat{\mathbf{m}}_{t+1,k}$ , respectively.

**Loss:** VIMON is trained in an unsupervised fashion with the following objective adapted from the MONET loss (Eq. (5)) for videos. To encourage the model to learn about object motion, we include a prediction objective in the form of a second decoder likelihood on the next-step prediction  $p_\theta(\mathbf{x}_{t+1} | \hat{\mathbf{z}}_{t+1,k})$  and an additional mask loss term, which encourages the predicted VAE mask distribution  $p_\theta(\hat{\mathbf{m}}_{t+1,k} | \hat{\mathbf{z}}_{t+1,k})$  to be close to the attention mask distribution  $q_\psi(\mathbf{m}_{t+1,k} | \mathbf{x}_{t+1})$  of the next time step for each slot  $k$ :

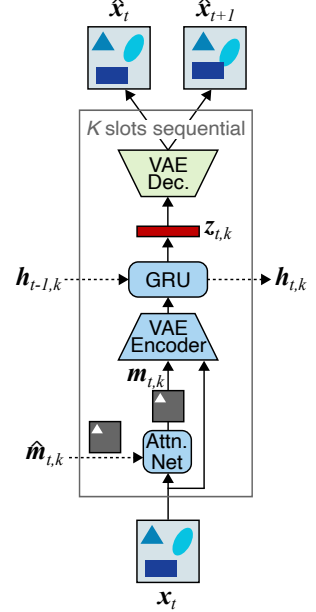


Figure C.1: VIMON. Attention network followed by VAE encoder and GRU computes latent  $\mathbf{z}_{t,k}$ .

$$\begin{aligned}
 L(\phi; \theta; \psi; \mathbf{x}) &= \sum_{t=1}^T L_{\text{negLL}} + \beta L_{\text{prior}} + \gamma L_{\text{mask}} \\
 L_{\text{negLL}} &= -(\log \sum_{k=1}^K \mathbf{m}_{t,k} p_{\theta}(\mathbf{x}_t | \mathbf{z}_{t,k}) + \log \sum_{k=1}^K \mathbf{m}_{t+1,k} p_{\theta}(\mathbf{x}_{t+1} | \widehat{\mathbf{z}}_{t+1,k})) \\
 L_{\text{prior}} &= D_{\text{KL}}(\prod_{k=1}^K q_{\phi}(\mathbf{z}_{t,k} | \mathbf{x}_t, \mathbf{m}_{t,k}) || p(\mathbf{z})) \\
 L_{\text{mask}} &= \sum_{k=1}^K D_{\text{KL}}(q_{\psi}(\mathbf{m}_{t,k} | \mathbf{x}_t) || p_{\theta}(\mathbf{m}_{t,k} | \mathbf{z}_{t,k})) + D_{\text{KL}}(q_{\psi}(\mathbf{m}_{t+1,k} | \mathbf{x}_{t+1}) || p_{\theta}(\mathbf{m}_{t+1,k} | \widehat{\mathbf{z}}_{t+1,k}))
 \end{aligned}$$

### C.3 Tracking-by-Animation

Tracking-by-Animation (TBA) (He et al., 2019) is a spatial transformer-based attention model which uses a simple 2D rendering pipeline as the decoder. Objects are assigned tracking templates and pose parameters by a tracker array, such that they can be reconstructed in parallel using a renderer based on affine spatial transformation (Fig. C.2). In contrast to VIMON, TBA uses explicit parameters to encode the position, size, aspect ratio and occlusion properties for each slot. Importantly, TBA is designed for scenes with static backgrounds, and pre-processes sequences using background subtraction (Bloisi and Iocchi, 2012) before they are input to the tracker array.

**Tracker Array:** TBA uses a tracker array to output a latent representation  $\mathbf{z}_t \in \mathbb{R}^{L \times K}$  at time  $t$  using a feature extractor  $f(\mathbf{x}_t; \psi)$  and a recurrent 'state update', where  $\mathbf{c}_t \in \mathbb{R}^{M \times N \times C}$  is a convolutional feature representation. The convolutional feature and latent representation have far fewer elements than  $\mathbf{x}_t$ , acting as a bottleneck:

$$\mathbf{c}_t = f(\mathbf{x}_t; \psi), \quad (10)$$

$$\mathbf{h}_{t,k} = \text{RAT}(\mathbf{h}_{t-1,k}, \mathbf{c}_t; \pi), \quad (11)$$

$$\mathbf{z}_t = g(\mathbf{h}_t; \phi). \quad (12)$$

Though the state update could be implemented as any generic recurrent neural network block, such as an LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014), TBA introduces a Reprioritized Attentive Tracking (RAT) block that uses attention to achieve explicit association of slots with similar features over time. Firstly, the previous tracker state  $\mathbf{h}_{t-1,k}$  is used to generate key variables  $\mathbf{k}_{t,k}$  and  $\beta_{t,k}$ :

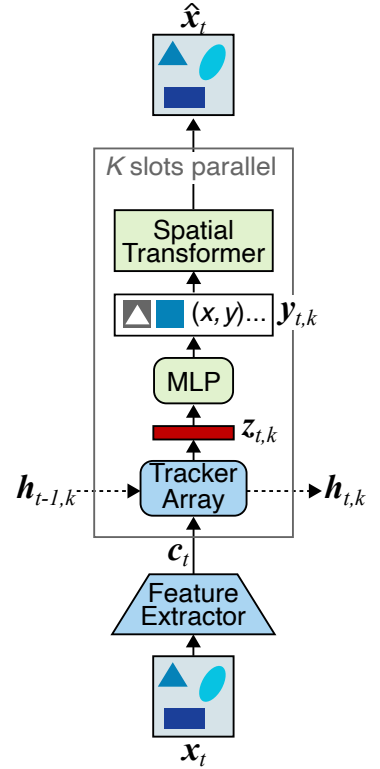


Figure C.2: **TBA.** Feature extractor CNN  $f$  and tracker array  $g$  to get latent  $\mathbf{z}_{t,k}$ . MLP  $h$  outputs mid-level representation  $\mathbf{y}_{t,k}$ , and Spatial Transformer renders reconstruction.

$$\{\mathbf{k}_{t,k}, \widehat{\beta}_{t,k}\} = \mathbf{T}\mathbf{h}_{t-1,k}, \quad (13)$$

$$\beta_{t,k} = 1 + \ln(1 + \exp(\widehat{\beta}_{t,k})), \quad (14)$$

where  $\mathbf{T}$  is a learned linear transformation,  $\mathbf{k}_{t,k} \in \mathbb{R}^S$  is the addressing key, and  $\widehat{\beta}_{t,k} \in \mathbb{R}$  is an un-normalized version of a key strength variable  $\beta_{t,k} \in (1, +\infty)$ . This key strength acts like a temperature parameter to modulate the feature re-weighting, which is described in the following. Each feature vector in  $\mathbf{c}_t$ , denoted by  $\mathbf{c}_{t,m,n} \in \mathbb{R}^S$ , where  $m \in \{1, 2, \dots, M\}$  and  $n \in \{1, 2, \dots, N\}$  are the convolutional feature dimensions, is first used to get attention weights:

$$W_{t,k,m,n} = \frac{\exp(\beta_{t,k} \text{Sim}(\mathbf{k}_{t,k}, \mathbf{c}_{t,m,n}))}{\sum_{m',n'} \exp(\beta_{t,k} \text{Sim}(\mathbf{k}_{t,k}, \mathbf{c}_{t,m',n'}))}. \quad (15)$$

Here,  $\text{Sim}$  is the cosine similarity defined as  $\text{Sim}(\mathbf{p}, \mathbf{q}) = \mathbf{p}\mathbf{q}^\top / (\|\mathbf{p}\| \|\mathbf{q}\|)$ , and  $W_{t,k,m,n}$  is an element of the attention weight  $\mathbf{W}_{t,k} \in [0, 1]^{M \times N}$ , satisfying  $\sum_{m,n} W_{t,k,m,n} = 1$ . Next, a read operation is defined as a weighted combination of all feature vectors of  $\mathbf{c}_t$ :

$$\mathbf{r}_{t,k} = \sum_{m,n} W_{t,k,m,n} \mathbf{c}_{t,m,n}, \quad (16)$$

where  $\mathbf{r}_{t,k} \in \mathbb{R}^S$  is the read vector, representing the associated input feature for slot  $k$ . Intuitively, for slots in which objects are present in the previous frame, the model can suppress the features in  $\mathbf{r}_{t,k}$  that are not similar to the features of that object, helping achieve better object-slot consistency. On the other hand, if there are slots which so far do not contain any object, the key strength parameter allows  $\mathbf{r}_{t,k}$  to remain similar to  $\mathbf{c}_t$ , facilitating the discovery of new objects.

The tracker state  $\mathbf{h}_{t,k}$  of the RAT block is updated with an RNN parameterized by  $\pi$ , taking  $\mathbf{r}_{t,k}$  instead of  $\mathbf{c}_t$  as its input feature:

$$\mathbf{h}_{t,k} = \text{RNN}(\mathbf{h}_{t-1,k}, \mathbf{r}_{t,k}; \pi). \quad (17)$$

The RAT block additionally allows for sequential prioritization of trackers, which in turn allows only a subset of trackers to update their state at a given time step, improving efficiency. For full details on the reprioritization and adaptive computation time elements of the RAT block, please refer to the original paper (He et al., 2019).

**Mid-Level Representation:** The key feature of TBA is that each latent vector  $\mathbf{z}_{t,k}$  is further decoded into a mid-level representation  $\mathbf{y}_{t,k} = \{y_{t,k}^c, \mathbf{y}_{t,k}^l, \mathbf{y}_{t,k}^p, \mathbf{Y}_{t,k}^s, \mathbf{Y}_{t,k}^a\}$  corresponding to interpretable, explicit object properties, via a fully-connected neural network  $h(\mathbf{z}_{t,k}; \theta)$  as follows:

$$\mathbf{y}_{t,k} = h(\mathbf{z}_{t,k}; \theta). \quad (18)$$

$h_\theta$  is shared by all slots, improving parameter efficiency. The different components of the mid-level representation are:

- Confidence  $y_{t,k}^c \in [0, 1]$ : Probability of existence of an object in that slot.
- Layer  $\mathbf{y}_{t,k}^l \in \{0, 1\}^O$ : One-hot encoding of the discretized pseudo-depth of the object relative to other objects in the frame. Each image is considered to be composed of  $O$  object layers, where higher layer objects occlude lower layer objects and the background is the zeroth (lowest) layer. E.g., when  $O=4$ ,  $\mathbf{y}_{t,k}^l = [0, 0, 1, 0]$  denotes the third layer. For simplicity and without loss of generality, we can also denote the same layer with its integer representation  $y_{t,k}^l = 3$ .
- Pose  $\mathbf{y}_{t,k}^p = [\hat{s}_{t,k}^x, \hat{s}_{t,k}^y, \hat{t}_{t,k}^x, \hat{t}_{t,k}^y] \in [-1, 1]^4$ : Normalized object pose for calculating the scale  $[s_{t,k}^x, s_{t,k}^y] = [1 + \eta^x \hat{s}_{t,k}^x, 1 + \eta^y \hat{s}_{t,k}^y]$  and the translation  $[t_{t,k}^x, t_{t,k}^y] = [\frac{W}{2} \hat{t}_{t,k}^x, \frac{H}{2} \hat{t}_{t,k}^y]$ , where  $\eta^x, \eta^y > 0$  are constants.
- Shape  $\mathbf{Y}_{t,k}^s \in \{0, 1\}^{U \times V}$  and Appearance  $\mathbf{Y}_{t,k}^a \in [0, 1]^{U \times V \times 3}$ : Object template, with hyperparameters  $U$  and  $V$  typically set much smaller than the image dimensions  $H$  and  $W$ . Note that the shape is discrete (for details, see below) whereas the appearance is continuous.

In the output layer of  $h_\theta$ ,  $y_{t,k}^c$  and  $\mathbf{Y}_{t,k}^a$  are generated by the sigmoid function,  $\mathbf{y}_{t,k}^p$  is generated by the tanh function, and  $\mathbf{y}_{t,k}^l$  as well as  $\mathbf{Y}_{t,k}^s$  are sampled from the Categorical and Bernoulli distributions, respectively. As sampling is non-differentiable, the Straight-Through Gumbel-Softmax estimator (Jang et al., 2017) is used to reparameterize both distributions so that backpropagation can still be applied.

**Renderer:** To obtain a frame reconstruction, the renderer scales and shifts  $\mathbf{Y}_{t,k}^s$  and  $\mathbf{Y}_{t,k}^a$  according to  $\mathbf{y}_{t,k}^p$  via a Spatial Transformer Network (STN) (Jaderberg et al., 2015):

$$\mathbf{m}_{t,k} = STN(\mathbf{Y}_{t,k}^s, \mathbf{y}_{t,k}^p), \quad (19)$$

$$\hat{\mathbf{x}}_{t,k} = STN(\mathbf{Y}_{t,k}^a, \mathbf{y}_{t,k}^p). \quad (20)$$

where  $\mathbf{m}_{t,k} \in \{0, 1\}^D$  and  $\hat{\mathbf{x}}_{t,k} \in [0, 1]^{D \times 3}$  are the spatially transformed shape and appearance respectively. To obtain the final object masks  $\hat{\mathbf{m}}_{t,k}$ , an occlusion check is performed by initializing  $\hat{\mathbf{m}}_{t,k} = y_{t,k}^c \mathbf{m}_{t,k}$ , then removing the elements of  $\hat{\mathbf{m}}_{t,k}$  for which there exists an object in a higher layer. That is, for  $k=1, 2, \dots, K$  and  $\forall j \neq k$  where  $y_{t,j}^l > y_{t,k}^l$ :

$$\hat{\mathbf{m}}_{t,k} = (\mathbf{1} - \mathbf{m}_{t,j}) \odot \hat{\mathbf{m}}_{t,k}. \quad (21)$$

In practice, the occlusion check is sped up by creating intermediate ‘layer masks’, partially parallelizing the operation. Please see the original paper for more details (He et al., 2019). The final reconstruction is obtained by summing over the  $K$  slots,  $\hat{\mathbf{x}}_t = \sum_{k=1}^K \hat{\mathbf{m}}_{t,k} \odot \hat{\mathbf{x}}_{t,k}$ .

**Loss:** Learning is driven by a pixel-level reconstruction objective, defined as:

$$L(\phi; \psi; \pi; \theta; \mathbf{x}) = \sum_{t=1}^T \left( MSE(\hat{\mathbf{x}}_t, \mathbf{x}_t) + \lambda \cdot \frac{1}{K} \sum_{k=1}^K s_{t,k}^x s_{t,k}^y \right), \quad (22)$$

where  $MSE$  refers to the mean squared error and the second term penalizes large scales  $[s_{t,k}^x, s_{t,k}^y]$  in order to make object bounding boxes more compact.

## C.4 IODINE

The Iterative Object Decomposition Inference Network (IODINE) (Greff et al., 2019), similar to MONET (Burgess et al., 2019), learns to decompose a static scene into a multi-slot representation, in which each slot represents an object in the scene and the slots share the underlying format of the independent representations. In contrast to MONET, it does not recurrently segment the image using spatial attention, rather it starts from an initial guess of the segmentation of the whole image and iteratively refines it. Thus, the inference component of both models differ, while the generative component is the same.

**Iterative Inference:** As with MONET, IODINE models the latent posterior  $q(\mathbf{z}_k|\mathbf{x})$  per slot  $k$  as a Gaussian parameterized by  $(\boldsymbol{\mu}_{m,k}, \boldsymbol{\sigma}_{m,k}) \in \mathbb{R}^{L \times 2}$ . To obtain latent representations for independent regions of the input image, IODINE starts from initial learned posterior parameters  $(\boldsymbol{\mu}_{1,k}, \boldsymbol{\sigma}_{1,k})$  and iteratively refines them using the refinement network  $f_\phi$  for a fixed number of refinement steps  $M$ .  $f_\phi$  consists of a convolutional neural network (CNN) in combination with an LSTM cell (Hochreiter and Schmidhuber, 1997) parameterized by  $\phi$ . In each processing step,  $f_\phi$  receives as input the image  $\mathbf{x} \in [0, 1]^{H \times W \times 3}$ , a sample from the current posterior estimate  $\mathbf{z}_{m,k} \in \mathbb{R}^L$  and various auxiliary inputs  $\mathbf{a}_k$ , which are listed in the original paper (Greff et al., 2019). The posterior parameters are concatenated with the output of the convolutional part of the refinement network and together form the input to the refinement LSTM. The posterior parameters are additively updated in each step  $m$  in parallel for all  $K$  slots:

$$(\boldsymbol{\mu}_{m+1,k}, \boldsymbol{\sigma}_{m+1,k}) = (\boldsymbol{\mu}_{m,k}, \boldsymbol{\sigma}_{m,k}) + f_\phi(\mathbf{z}_{m,k}, \mathbf{x}, \mathbf{a}_k). \quad (23)$$

**Decoder:** In each refinement step  $m$ , the image is represented by  $K$  latent representations  $\mathbf{z}_{m,k}$ . Similar to MONET, each  $\mathbf{z}_{m,k}$  is independently decoded into a reconstruction of the image  $\hat{\mathbf{x}}_{m,k} \in [0, 1]^{H \times W \times 3}$  and mask logits  $\tilde{\mathbf{m}}_{m,k}$ , which are subsequently normalized by applying the softmax across slots to obtain the masks  $\mathbf{m}_{m,k} \in [0, 1]^{H \times W}$ . The reconstruction of the whole image at each refinement step  $m$  is composed by summing over the  $K$  masked reconstructions of the decoder:  $\hat{\mathbf{x}} = \sum_{k=1}^K \mathbf{m}_{m,k} \odot \hat{\mathbf{x}}_{m,k}$ .

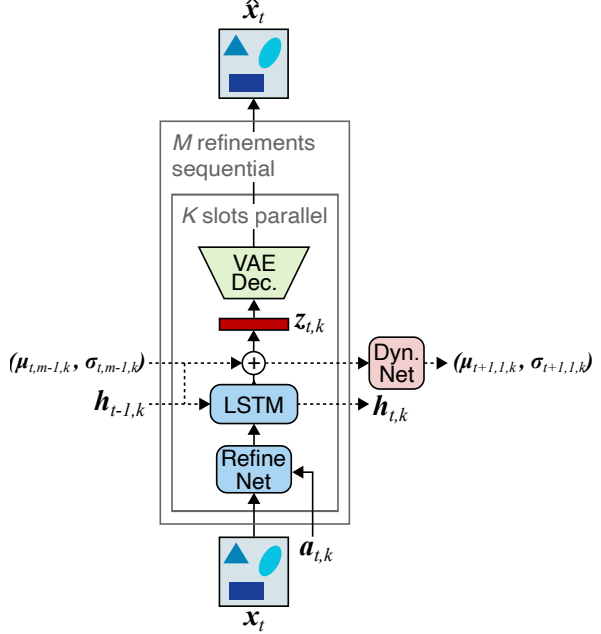


Figure C.3: **OP3**. Refinement network  $f$  followed by LSTM and dynamics network  $d$  compute latent  $\mathbf{z}_{t,k}$ .

**Training:** IODINE is trained by minimizing the following loss function that consists of the Evidence Lower BOund (ELBO) (Kingma and Welling, 2014) unrolled through  $N$  iterations:

$$L(\theta, \phi, (\boldsymbol{\mu}_{1,k}, \boldsymbol{\sigma}_{1,k}); \mathbf{x}) = \sum_{m=1}^M \frac{m}{M} \left[ -\log \sum_{k=1}^K \mathbf{m}_{m,k} p_{\theta}(\mathbf{x}|\mathbf{z}_{m,k}) + D_{\text{KL}} \left( \prod_{k=1}^K q_{\phi}(\mathbf{z}_{m,k}|\mathbf{x}) \| p(\mathbf{z}) \right) \right], \quad (24)$$

where  $p_{\theta}(\mathbf{x}|\mathbf{z}_{m,k})$  is the decoder log-likelihood weighted by the mask  $\mathbf{m}_k$  and  $D_{\text{KL}}$  is the Kullback-Leibler divergence between the unit Gaussian prior  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$  and the latent posterior distribution  $q(\mathbf{z}_{m,k}|\mathbf{x})$  factorized across slots.

### C.5 Object-centric Perception, Prediction, and Planning (OP3)

Object-centric Perception, Prediction, and Planning (OP3) (Veerapaneni et al., 2020) extends IODINE to work on videos and in a reinforcement learning (RL) setting. It uses the above described IODINE as an observation model to decompose visual observations into objects and represent them independently. These representations are subsequently processed by a dynamics model that models the individual dynamics of the objects, the pairwise interaction between the objects, as well as the action’s effect on the object’s dynamics, predicting the next frame in latent space (Fig. C.3). By modeling the action’s influence on individual objects, OP3 can be applied to RL tasks.

OP3 performs  $M$  refinement steps after each dynamics step.

**Refinement network:** The refinement steps proceed as in the description for IODINE in Section C.4. The input image  $\mathbf{x}_t \in [0, 1]^{H \times W \times 3}$ , which is the frame from a video at time  $t$ , is processed by the refinement network  $f_{\phi}$  conditioned on a sample from the current posterior estimate  $\mathbf{z}_{t,m,k} \in \mathbb{R}^L$ . The refinement network outputs an update of the posterior parameters  $(\boldsymbol{\mu}_{t,m,k}, \boldsymbol{\sigma}_{t,m,k})$  (see Eq. (23)). The posterior parameters  $(\mu_{1,1,k}, \sigma_{1,1,k})$  are randomly initialized.

**Dynamics model:** After refinement, samples from the current posterior estimate  $\mathbf{z}_{t,M,k}$  for each slot  $k$  are used as input to the dynamics network. The dynamics model  $d_{\psi}$  consists of a series of linear layers and nonlinearities parameterized by  $\psi$ . It models the individual dynamics of the objects per slot  $k$ , the pairwise interaction between all combinations of objects, aggregating them into a prediction of the posterior parameters for the next time step  $t + 1$  for each object  $k$ . The full dynamics model additionally contains an action component that models the influence of a given action on each object, which we do not use in our tracking setting. The predicted posterior parameters are then used in the next time step as initial parameters for the refinement network.

$$(\boldsymbol{\mu}_{t,1,k}, \boldsymbol{\sigma}_{t,1,k}) = d_{\psi}(\mathbf{z}_{t-1,M,k}, \mathbf{z}_{t-1,M,[\neq k]}). \quad (25)$$

**Training.** OP3 is trained end-to-end with the ELBO used at every refinement and dynamics step, with the loss  $L(\theta, \phi; \mathbf{x})$  given by:

$$\sum_{t=1}^T \frac{1}{T} \sum_{m=1}^{M+1} \frac{\min(m, M)}{M} \left( -\log \sum_{k=1}^K \mathbf{m}_{t,m,k} p_{\theta}(\mathbf{x}_t | \mathbf{z}_{t,m,k}) + D_{\text{KL}} \left( \prod_{k=1}^K q_{\phi}(\mathbf{z}_{t,m,k} | \mathbf{x}_t) \parallel q(\mathbf{z}_{t,1,k} | \mathbf{x}_t) \right) \right), \quad (26)$$

where for time step 1,  $q(\mathbf{z}_{1,1,k} | \mathbf{x}_1) = \mathcal{N}(\mathbf{0}, I)$ .

## C.6 SCALable Object-oriented Representation (SCALOR)

SCALable Object-oriented Representation (SCALOR) (Jiang et al., 2020) is a spatial transformer-based model that extends SQAIR (Kosiorrek et al., 2018) to scale to cluttered scenes. Similar to TBA, it factors the latent representations in pose, depth and appearance per object and uses spatial transformers (Jaderberg et al., 2015) to render objects in parallel. In contrast to TBA, it can handle dynamic backgrounds by integrating a background RNN that models background transitions.

**Proposal-Rejection Module:** SCALOR uses a proposal-rejection module  $g$  to discover new objects. All frames up to the current time step  $x_{1:t}$  are first encoded using a convolutional LSTM  $f$ . The resulting features are then aggregated with an encoding of propagated object masks and divided into  $H \times W$  grid cells.

$$\mathbf{c}_t^{img} = f(\mathbf{x}_{1:t}; \psi), \quad (27)$$

$$\mathbf{c}_t^{mask} = \text{MaskEncoder}(M_t^P), \quad (28)$$

$$\mathbf{c}_t^{agg} = \text{Concat}([\mathbf{c}_t^{img}, \mathbf{c}_t^{mask}]). \quad (29)$$

Per grid cell a latent variable  $\mathbf{z}_{t,h,w}$  is proposed. Proposal generation is done in parallel. Each  $\mathbf{z}_{t,h,w}$  consists of existence, pose, depth and appearance parameters  $(\mathbf{z}_{t,h,w}^{pres}, \mathbf{z}_{t,h,w}^{pose}, \mathbf{z}_{t,h,w}^{depth}, \mathbf{z}_{t,h,w}^{what})$ .

$$\mathbf{z}_{t,h,w}^{pres} \sim \text{Bern}(\cdot | g_1(\mathbf{c}_t^{agg})), \quad (30)$$

$$\mathbf{z}_{t,h,w}^{depth} \sim \mathcal{N}(\cdot | g_2(\mathbf{c}_t^{agg})), \quad (31)$$

$$\mathbf{z}_{t,h,w}^{pose} \sim \mathcal{N}(\cdot | g_3(\mathbf{c}_t^{agg})), \quad (32)$$

where  $g_1, g_2$  and  $g_3$  are convolutional layers.

The appearance parameters  $\mathbf{z}_{t,h,w}^{what}$  are obtained by first taking a glimpse from frame  $x_t$  of the area specified by  $\mathbf{z}_{t,h,w}^{pose}$  via a Spatial Transformer Network (STN) (Jaderberg et al., 2015) and subsequently extracting features from it via a convolutional neural network:

$$\mathbf{c}_{t,h,w}^{att} = \text{STN}(x_t, \mathbf{z}_{t,h,w}^{pose}), \quad (33)$$

$$\mathbf{z}_{t,h,w}^{what} \sim \mathcal{N}(\cdot | \text{GlimpseEnc}(\mathbf{c}_{t,h,w}^{att})), \quad (34)$$

$$\mathbf{o}_{t,h,w}, \mathbf{m}_{t,h,w} = \text{STN}^{-1}(\text{GlimpseDec}(\mathbf{z}_{t,h,w}^{what}), \mathbf{z}_{t,h,w}^{pose}), \quad (35)$$

where  $\mathbf{o}_{t,h,w}$  is the object RGB glimpse and  $\mathbf{m}_{t,h,w}$  is the object mask glimpse.



In the rejection phase, objects that overlap more than a threshold  $\tau$  in pixel space with a propagated object from the previous time step are rejected.

**Propagation Module:** During propagation, for each object  $k$  from the previous time step  $t - 1$  a feature attention map  $\mathbf{a}_{t,k}$  from the encoded frame features  $\mathbf{c}_t^{img}$  is extracted centered on the position of the object in the previous time step and used to update the hidden state  $\mathbf{h}_{t,k}$  of the tracker RNN for object  $k$ .

$$\mathbf{a}_{t,k} = att(STN(\mathbf{c}_t^{img}, \mathbf{z}_{t-1,k}^{pose}), \quad (36)$$

$$\mathbf{h}_{t,k} = GRU([\mathbf{a}_{t,k}, \mathbf{z}_{t-1,k}], \mathbf{h}_{t-1,k}), \quad (37)$$

$$\mathbf{z}_{t,k} = update(\mathbf{a}_{t,k}, \mathbf{h}_{t,k}, \mathbf{z}_{t-1,k}), \quad (38)$$

where  $STN$  is a spatial transformer module (Jaderberg et al., 2015). If  $\mathbf{z}_{t,k}^{pres} = 1$  the latent representation  $\mathbf{z}_{t,k}$  of the respective object  $k$  will be propagated to the next time step.

**Background:** The background of each frame  $x_t$  is encoded using a convolutional neural network conditioned on the masks  $M_t$  of the objects present at time step  $t$  and decoded using a convolutional neural network.

$$(\boldsymbol{\mu}^{bg}, \boldsymbol{\sigma}^{bg}) = BgEncoder(x_t, (1 - M_t)), \quad (39)$$

$$\mathbf{z}_t^{bg} \sim \mathcal{N}(\boldsymbol{\mu}^{bg}, \boldsymbol{\sigma}^{bg}), \quad (40)$$

$$\hat{\mathbf{x}}_t^{bg} = BgDecoder(\mathbf{z}_t^{bg}). \quad (41)$$

**Rendering:** To obtain frame reconstructions  $\hat{\mathbf{x}}_t$  foreground object appearances and masks are scaled and shifted using via a Spatial Transformer Network (STN):

$$\hat{\mathbf{x}}_{t,k}^{fg} = STN^{-1}(\mathbf{o}_{t,k}, \mathbf{z}_{t,k}^{pose}), \quad (42)$$

$$\gamma_{t,k} = STN^{-1}(\mathbf{m}_{t,k} \cdot \mathbf{z}_{t,k}^{pres} \sigma(-\mathbf{z}_{t,k}^{depth}), \mathbf{z}_{t,k}^{pose}), \quad (43)$$

$$\hat{\mathbf{x}}_t^{fg} = \sum_K \hat{\mathbf{x}}_{t,k}^{fg} \gamma_{t,k}. \quad (44)$$

Subsequently, foreground objects and background reconstruction are combined as follows to obtain the final reconstruction:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^{fg} + (1 - M_t) \odot \hat{\mathbf{x}}_t^{bg}. \quad (45)$$

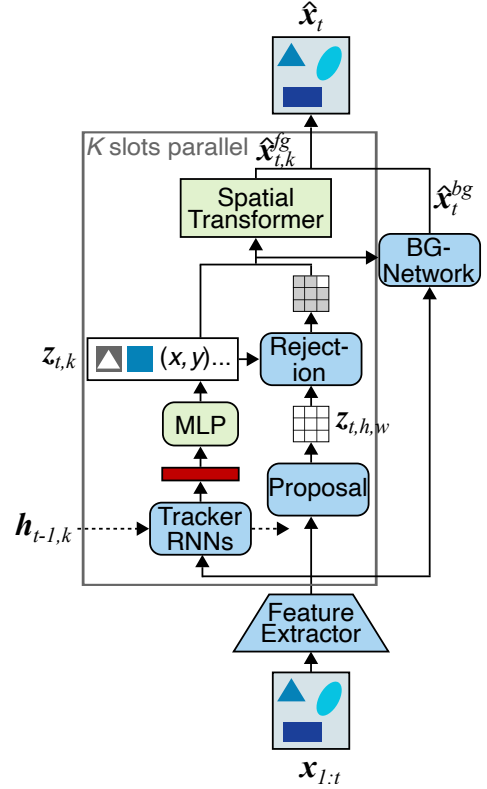


Figure C.4: **SCALOR** Feature extractor CNN  $f$  followed by tracker RNNs or proposal-rejection module to compute latent  $\mathbf{z}_{t,k}$ . Spatial Transformer in addition to background module renders reconstruction.

**Training:** SCALOR is trained on frame reconstruction using the evidence lower bound (ELBO):

$$\sum_{t=1}^T -\log p_{\theta}(\mathbf{x}_t|\mathbf{z}_t) + D_{\text{KL}}(q_{\phi}(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t})||q(\mathbf{z}_t|\mathbf{z}_{<t})). \quad (46)$$

## Appendix D. Model Implementation Details

### D.1 Baseline: k-Means

As a baseline, we implement a simple color segmentation algorithm to evaluate if this is sufficient to solve the benchmark data sets. To this end, we cluster the videos using the k-Means algorithm as implemented by sklearn (Pedregosa et al., 2011). The number of clusters are chosen equal to the number of slots in the other models, i.e.  $K = 5$  for SpMOT,  $K = 6$  for VMDS and  $K = 8$  for VOR. We run the clustering three times per video with different random seeds. Each pixel is treated as one sample with 3 features (RGB values). k-Means clustering is performed for each frame in the video separately. To obtain the tracking performance, we match objects in consecutive frames using IoU of the extracted segmentation masks with the Hungarian matching algorithm (Kuhn, 1955).

### D.2 MONET and Video MONET

**VAE:** Following Burgess et al. (2019), the VAE encoder is a CNN with 3x3 kernels, stride 2, and ReLU activations (Table D.1). It receives the input image and mask from the attention network as input and outputs  $(\mu, \log \sigma)$  of a 16-dimensional Gaussian latent posterior. The GRU has 128 latent dimensions and one hidden state per slot followed by a linear layer with 32 output dimensions. The VAE decoder is a Broadcast decoder as published by Watters et al. (2019b) with no padding, 3x3 kernels, stride 1 and ReLU activations (Table D.2). The output distribution is an independent pixel-wise Gaussian with a fixed scale of  $\sigma = 0.09$  for the background slot and  $\sigma = 0.11$  for the foreground slots.

**Attention Network:** The attention network is a U-Net (Ronneberger et al., 2015) and follows the architecture proposed by Burgess et al. (2019). The down- and up-sampling components consist each of five blocks with 3x3 kernels, 32 channels, instance normalization, ReLU activations and down- or up-sampling by a factor of two. The convolutional layers are bias-free and use stride 1 and padding 1. A three-layer MLP with hidden layers of size 128 connect the down- and the up-sampling part of the U-Net.

**Training:** MONET and VIMON are implemented in PyTorch (Paszke et al., 2019) and trained with the Adam optimizer (Kingma and Ba, 2015) with a batch size of 64 for MONET and 32 for VIMON, using an initial learning rate of 0.0001. Reconstruction performance is evaluated after each epoch on the validation set and the learning rate is decreased by a factor of 3 after the validation loss has not improved in 25 consecutive epochs for MONET and 100 epochs for VIMON, respectively. MONET and VIMON are trained for 600 and 1000 epochs, respectively. The checkpoint with the lowest reconstruction error is selected for the final MOT evaluation. MONET is trained with  $\beta = 0.5$  and  $\gamma = 1$  and VIMON is trained with  $\beta = 1$  and  $\gamma = 2$ .  $K = 5$  for SpMOT,  $K = 6$  for VMDS and  $K = 8$  for VOR. Due to the increased slot number for VOR, batch size for VIMON had to be decreased to

Type	Size/Ch.	Act. Func.	Comment
Input	4		RGB + Mask
Conv 3x3	32	ReLU	
Conv 3x3	32	ReLU	
Conv 3x3	64	ReLU	
Conv 3x3	64	ReLU	
MLP	256	ReLU	
MLP	32	Linear	

Table D.1: Architecture of ViMON **VAE Encoder**.

Type	Size/Ch.	Act. Func.	Comment
Input	16		
Broadcast	18		+ coordinates
Conv 3x3	32	ReLU	
Conv 3x3	32	ReLU	
Conv 3x3	32	ReLU	
Conv 3x3	32	ReLU	
Conv 1x1	4	Linear	RGB + Mask

Table D.2: Architecture of ViMON **VAE Decoder**.

24 to fit into the GPU memory. Respectively, the initial learning rate is set to 0.000075 for ViMON on VOR. We initialize the attention network and the VAE in ViMON with the pre-trained weights from MONET to facilitate learning and speed up the training. Note that for all evaluations, the reconstructed masks  $\hat{\mathbf{m}}$  from the VAE were used.

**Sprites-MOT Initialization:** When training MONET and Video MONET on Sprites-MOT from scratch, MONET struggles to learn the extreme color values of the objects that Sprites-MOT features. Instead it completely focuses on learning the shapes. To circumvent that, we initialized the weights of the models with MONET weights that were trained for 100 epochs on Multi-dSprites.

**Hyperparameter tuning:** Hyperparameters for MONET and ViMON were selected using the best MSE on the validation set of VMDS. Hyperparameter tuning for MONET included  $\beta$  from  $\{0.5, 1.0\}$  and  $\gamma$  from  $\{0.5, 1.0, 2.0\}$  and initial learning rate from  $\{1e-05, 1e-04\}$ .

Hyperparameter tuning for ViMON included  $\beta$  from  $\{0.5, 1.0\}$  and  $\gamma$  from  $\{1.0, 2.0, 5.0, 10.0, 50.0\}$  and initial learning rate from  $\{1e-05, 1e-04, 1e-03\}$ .

Hyperparameters for **texVMDS** were kept fixed to the best selected hyperparameters for VMDS, except for the VAE latent dimension, which is tuned to account for the higher visual complexity. The VAE latent dimension 16 is selected from  $\{16, 32, 64, 128\}$  as the one with the best MOTA for MONET on texVMDS validation set — analogous to the SCALOR hyperparameter tuning paradigm — and subsequently also used for training of ViMON on texVMDS.

### D.3 Tracking by Animation

**Preprocessing:** TBA expects its input frames to contain only foreground objects. In He et al. (2019), the authors use Independent Multimodal Background Subtraction (IMBS) (Bloisi and Iocchi, 2012) to remove the background from data sets consisting of natural videos with static backgrounds. Background subtraction algorithms maintain a spatio-temporal window around each pixel in the sequence, and remove the dominant mode based on a histogram of color values. Since the default implementation of IMBS has several hand-tuned thresholds corresponding to natural videos (e.g., for shadow suppression), it cannot be directly applied to synthetic data sets like VMDS without significant hyperparameter tuning. We instead re-generate all of the VMDS data sets with identical objects and motion but a black background for our experiments with TBA, to mimic a well-tuned background subtraction algorithm.

**Architecture:** For SpMOT, we follow the same architecture as in (He et al., 2019), while we increase the number of slots from  $K = 4$  to  $K = 5$  and number of layers from  $O = 3$  to  $O = 4$  for VMDS. Since TBA does not model the background, this makes the number of foreground slots equal to the other models in our study.

**Hyperparameter tuning:** Further, we increase the size prior parameters  $U \times V$  used for the shape and appearance templates from  $21 \times 21$  which is used for SpMOT, to  $64 \times 64$  for VMDS, which we empirically found gave the best validation loss among  $48 \times 48$ ,  $56 \times 56$ ,  $64 \times 64$  and  $72 \times 72$ . All other architectural choices are kept fixed for both data sets, and follow He et al. (2019). Note that due to this, we trained the TBA models at its default resolution of  $128 \times 128$  unlike the  $64 \times 64$  resolution used by MONET and OP3.

**Training and Evaluation:** We train for 1000 epochs using the same training schedule as He et al. (2019). The checkpoint with the lowest validation loss is selected for the final MOT evaluation. Further, we observed that the discrete nature of the shape code used in TBA’s mid-level representation leads to salt-and-pepper noise in the reconstructed masks. We therefore use a  $2 \times 2$  minimum pooling operation on the final output masks to remove isolated, single pixel foreground predictions and generate  $64 \times 64$  resolution outputs, similar to MONET and OP3 before evaluation.

**Deviation of SpMOT results compared to original publication:** Our results were generated with 100k training frames, while the original TBA paper (He et al., 2019) uses 2M training frames for the simple SpMOT task. Further, we report the mean of three training runs, while the original paper reports one run (presumably the best). Our best run achieves MOTA of 90.5 (Table E.1). Third, we evaluate using intersection over union (IoU) of segmentation masks instead of bounding boxes.

### D.4 OP3

**Training:** The OP3 loss is a weighted sum over all refinement and dynamics steps (Eq. (26)). For our evaluation on multi-object tracking, we weight all time steps equally. In contrast to the original training loss, in which the weight value is linearly increased indiscriminately, thus weighting later predictions more highly, we perform the linear increase only for the refinement steps between dynamics steps, thus weighting all predictions equally.

OP3, as published by Veerapaneni et al. (2020), uses curriculum learning. For the first 100 epochs,  $M$  refinement steps are taken, followed by a single dynamics step, with a final

refinement step afterwards. Starting after 100 epochs, the number of dynamics steps is incremented by 1 every 10 epochs, until five dynamics steps are reached. Thus, only 5 frames of the sequence are used during training at maximum.

We chose to use an alternating schedule for training, where after each dynamics step,  $M = 2$  refinement steps are taken, and this is continued for the entire sequence. Thus, the entire available sequence is used, and error is not propagated needlessly, since the model is enabled to refine previous predictions on the reconstruction before predicting again. Note that this is the schedule OP3 uses by default at test-time, when it is used for model predictive control. Note that we still use 4 refinement steps on the initial observation to update the randomly initialized posterior parameters, as in the released implementation. We split all 10-step sequences into 5-step sequences to avoid premature divergence.

**Hyperparameter tuning:** Note that OP3 by default uses a batch size of 80 with the default learning rate of 0.0003. As we found training OP3 to be very unstable, leading to eventual divergence in almost all experiments that have been performed for this study, we tested batch sizes [64, 32, 16, 8] and found reducing the batch size significantly, to 16, improved performance. We found the default learning rate to be too large for SpMOT in particular, as the model diverged significantly earlier than on VMDS and VOR. We tested learning rates [0.0002, 0.0001, 0.00006, 0.00003], and found decreasing the learning rate to 0.0001 rectified the observed premature divergence.

The checkpoint prior to divergence with the lowest KL loss is selected for the final MOT evaluation, as the KL loss enforces consistency in the latents over the sequence. Interestingly, the checkpoint almost always corresponded to the epochs right before divergence.

## D.5 SCALOR

**Architecture:** We follow the same architecture as in Jiang et al. (2020). We use a grid of  $4 \times 4$  for object discovery with a maximum number of objects of 10. The standard deviation of the image distribution is set to 0.1. Size anchor and variance are set to 0.2 and 0.1, respectively.

For SpMOT, background modeling is disabled and the dimensionality of the latent object appearance is set to 8.

For VMDS, the dimensionality of background is set to 3 and the dimensionality of the latent object appearance is set to 16. For object discovery, a grid of  $3 \times 3$  cells with a maximum number of objects of 8 is used.

For VOR, the dimensionality of background is set to 8 and the dimensionality of the latent object appearance is set to 16.

For texVMDS, the dimensionality of background is set to 32 and the dimensionality of the latent object appearance is set to 16. For object discovery, a grid of  $3 \times 3$  cells with a maximum number of objects of 8 is used.

Best model checkpoint according to the validation loss was chosen for MOT evaluation.

**Hyperparameter tuning:** For VMDS, we run hyperparameter search over number of grid cells  $\{3 \times 3, 4 \times 4\}$ , background dimension  $\{1, 3, 5\}$ , maximum number of objects  $\{5, 8, 10\}$  (dependent on number of grid cells), size anchor  $\{0.2, 0.25, 0.3, 0.4\}$ ,  $z^{\text{what}}$  dimensionality  $\{8, 16, 24\}$  and end value of  $\tau$   $\{0.3, 0.5\}$ .

For SpMOT, we run hyperparameter search over maximum number of objects  $\{4, 10\}$ , size anchor  $\{0.1, 0.2, 0.3\}$ ,  $z^{\text{what}}$  dimensionality  $\{8, 16\}$  and whether to model background (with background dimensionality 1) or not.

For VOR, we run hyperparameter search over size anchor  $\{0.2, 0.3\}$  and background dimensionality  $\{8, 12\}$ .

For texVMDS, we keep hyperparameters fixed as for VMDS and only tune background dimension  $\{3, 8, 16, 32, 64, 128\}$  and  $z^{\text{what}}$  dimensionality  $\{16, 32, 64, 128\}$  to account for the higher visual complexity of the data set.

We picked best hyperparameters according to the validation loss with the exception of the data set texVMDS, for which the best validation loss was given by the model using a latent dimensionality of 128 for both background and object. This led to the model using only the background slot to reconstruct the whole frame without using the object slots (object matches  $\sim 3\%$ ). As a sanity check, we picked the model which had the best MOTA on the validation set (shown in Table 3), but it also did not perform well on the more complex texVMDS data set.

We additionally tested if down-weighting the log-likelihood during training by a factor of 10, while keeping the high latent dimensionality (128 for both  $z^{\text{what}}$  and background) would help, by decreasing the focus on reconstructing the high frequency textures, but it still relied on the background to reconstruct the image without using object slots.

**Training:** We train SCALOR with a batch size of 16 for 300 epochs using a learning rate of 0.0001 for SpMOT and VOR and for 400 epochs for VMDS. For the final MOT evaluation, the checkpoint with the lowest loss on the validation set is chosen.

## Appendix E. Additional Results

Table E.1 lists the individual results for the three training runs with different random seeds per model and data set. The results of ViMON and SCALOR are coherent between the three runs with different random seed, while TBA has one run on SpMOT with significantly lower performance than the other two and shows variation in the three training runs on VMDS. OP3 exhibits one training run on SpMOT with lower performance than the other two.

Fig. E.2 shows the fraction of failure cases dependent on the number of objects over all types of failures (misses, ID. switches and FPs). Fig. E.1 shows the fraction of failure cases dependent on the number of objects present in the video for the three different failure cases separately; ID switches, FPs and misses. For ViMON, TBA and SCALOR, the number of failures increase with the number of objects present regardless of the type of failure. In contrast, OP3 shows this pattern for ID switches and misses, while it accumulates a higher number of false positives (FPs) in videos with fewer (only one or two) objects.

Fig. E.4 shows a comparison between MONET and ViMON on VMDS. MONET correctly finds and segments objects, but it does not assign them to consistent slots over time, while ViMON maintains a consistent slot assignment throughout the video.

Fig. E.5 shows failures cases of OP3 on VOR.

Table E.3 and Table E.4 list the results for the four models, ViMON, TBA, OP3 and SCALOR, on the VMDS challenge sets and out-of-distribution (o.o.d.) sets respectively.

BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

Model	Run	MOTA $\uparrow$	MOTP $\uparrow$	MD $\uparrow$	MT $\uparrow$	Match $\uparrow$	Miss $\downarrow$	ID S. $\downarrow$	FPs $\downarrow$	MSE $\downarrow$
<b>SpMOT</b>										
K-MEANS	1	36.2	77.6	76.8	76.3	80.5	19.2	0.2	44.3	-
	2	36.2	77.6	76.8	76.3	80.5	19.2	0.2	44.3	-
	3	36.3	77.6	76.8	76.2	80.5	19.3	0.2	44.2	-
MONET	1	70.0	90.6	92.8	49.4	74.7	4.1	21.2	4.7	10.4
	2	69.4	90.0	92.7	48.1	74.2	4.1	21.6	4.8	13.4
	3	71.3	88.1	91.6	53.8	77.1	4.9	18.0	5.8	15.2
ViMON	1	92.7	92.0	87.5	87.0	94.9	4.9	0.2	2.2	10.5
	2	92.8	92.0	86.9	86.3	94.8	5.0	0.2	2.0	11.8
	3	93.2	91.6	88.8	88.3	95.2	4.6	0.2	2.0	10.9
TBA	1	90.5	71.4	90.2	89.8	94.4	5.3	0.3	3.9	10.3
	2	58.4	70.7	69.6	60.8	75.0	18.1	6.9	16.6	14.6
	3	90.1	71.5	90.3	89.4	94.0	5.5	0.5	3.9	10.9
OP3	1	92.4	80.0	94.5	93.7	97.3	2.4	0.4	4.8	4.3
	2	81.9	74.9	86.9	86.5	92.8	6.8	0.3	10.9	30.1
	3	92.9	80.1	95.9	95.2	97.6	2.0	0.4	4.7	5.6
SCALOR	1	94.4	80.1	96.5	92.3	95.4	2.4	2.2	1.0	3.3
	2	94.7	80.2	96.4	93.1	95.8	2.4	1.8	1.1	3.4
	3	95.5	80.2	96.3	94.0	96.4	2.4	1.2	0.9	3.6
<b>VOR</b>										
K-MEANS	1	-38.0	76.5	69.8	62.8	72.6	22.1	5.3	110.7	-
	2	-38.1	76.5	69.8	63.0	72.7	22.1	5.2	110.8	-
	3	-37.9	76.5	70.0	62.9	72.8	22.0	5.2	110.7	-
MONET	1	28.0	81.3	73.8	26.7	57.4	18.0	24.6	29.4	14.1
	2	44.5	82.4	78.2	45.4	68.7	15.0	16.3	24.2	11.8
	3	38.5	81.6	78.7	39.8	67.0	14.4	18.5	28.5	10.8
ViMON	1	89.0	88.9	90.2	89.8	92.9	6.8	0.3	3.9	7.1
	2	89.0	89.8	89.9	89.6	93.0	6.8	0.2	4.0	6.2
	3	89.0	89.9	91.0	90.6	93.8	6.0	0.2	4.8	5.9
OP3	1	64.8	89.5	87.2	85.1	90.3	8.8	0.9	25.5	3.1
	2	66.2	88.1	88.6	85.1	90.7	7.9	1.4	24.5	2.9
	3	65.3	89.3	88.2	86.1	91.1	8.0	0.9	25.8	3.0
SCALOR	1	74.1	85.8	75.6	75.5	77.4	22.6	0.0	3.3	6.4
	2	74.6	86.0	75.9	75.9	78.1	21.9	0.1	3.5	6.4
	3	75.1	86.1	76.5	76.4	78.2	21.7	0.0	3.1	6.3

Table E.1: Analysis of SOTA object-centric representation learning models for MOT. Results for three runs with different random training seeds.

Model	Run	MOTA $\uparrow$	MOTP $\uparrow$	MD $\uparrow$	MT $\uparrow$	Match $\uparrow$	Miss $\downarrow$	ID S. $\downarrow$	FPS $\downarrow$	MSE $\downarrow$
<b>VMDS</b>										
K-MEANS	1	-3.3	89.8	98.3	93.4	96.4	1.0	2.6	99.7	-
	2	-3.3	89.8	98.2	93.2	96.4	1.1	2.6	99.7	-
	3	-3.2	89.8	98.2	93.4	96.4	1.0	2.6	99.6	-
MONET	1	51.7	79.6	75.1	36.7	67.6	12.9	19.5	15.9	20.8
	2	44.3	76.1	71.8	34.8	65.9	15.0	19.1	21.5	25.3
	3	52.2	80.2	75.6	35.5	66.5	13.0	20.5	14.2	20.4
ViMON	1	87.0	86.8	86.7	85.4	92.4	6.8	0.7	5.5	10.6
	2	87.1	86.8	86.1	85.1	92.3	7.1	0.6	5.3	10.8
	3	86.5	86.7	86.0	84.6	92.1	7.2	0.7	5.6	10.6
TBA	1	68.5	76.1	69.3	65.3	80.7	16.5	2.8	12.2	26.0
	2	38.9	73.8	55.1	50.5	70.2	26.6	3.2	31.3	30.8
	3	56.0	75.0	64.3	59.2	76.7	19.8	3.5	20.8	27.5
OP3	1	93.1	94.2	97.2	96.7	98.0	1.9	0.2	4.9	4.0
	2	92.7	93.4	96.9	96.3	97.8	2.0	0.2	5.1	4.3
	3	89.4	93.3	96.2	95.8	97.6	2.2	0.2	8.3	4.6
SCALOR	1	75.7	88.1	69.4	68.3	79.8	19.4	0.8	4.0	13.9
	2	72.7	87.2	66.7	65.6	77.6	21.6	0.8	4.9	14.2
	3	73.7	87.6	67.5	66.2	77.9	21.2	0.9	4.2	14.0
<b>texVMDS</b>										
K-MEANS	1	-99.5	76.4	25.4	24.3	30.3	69.2	0.5	129.8	-
	2	-99.5	76.4	25.3	24.4	30.4	69.2	0.5	129.9	-
	3	-99.5	76.4	25.3	24.3	30.3	69.2	0.5	129.8	-
MONET	1	-70.8	67.1	15.5	13.4	24.9	73.9	1.2	95.7	203.0
	2	-68.1	69.2	20.4	15.4	30.4	66.5	3.1	98.6	192.5
	3	-80.9	66.7	12.1	8.1	18.9	78.9	2.2	99.8	205.9
ViMON	1	-88.2	68.3	24.2	23.8	35.5	64.2	0.3	123.7	174.6
	2	-86.6	69.0	22.6	22.0	32.4	67.4	0.3	119.0	172.7
	3	-81.6	69.8	25.8	25.4	36.3	63.4	0.3	117.9	166.9
OP3	1	-105.6	71.5	20.6	20.4	28.3	71.6	0.2	133.9	113.6
	2	-116.0	70.1	9.4	9.2	13.6	86.3	0.1	129.6	153.2
	3	-109.7	70.3	19.5	19.0	26.9	72.8	0.3	136.6	131.8
SCALOR	1	-84.8	73.3	6.4	6.3	12.5	87.3	0.2	97.3	149.5
	2	-99.5	74.4	7.3	7.0	12.7	87.1	0.3	112.1	125.0
	3	-113.4	74.4	5.8	5.6	11.8	88.0	0.3	125.2	126.7

Table E.2: Analysis of SOTA object-centric representation learning models for MOT. Results for three runs with different random training seeds.



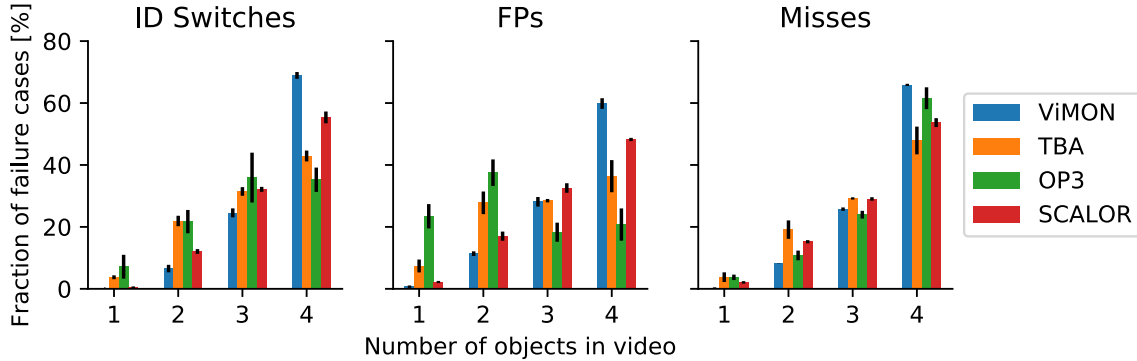


Figure E.1: Distribution of failure cases dependent on number of objects in VMDS videos split by failure class. Mean of three training runs. Error bars: SD.

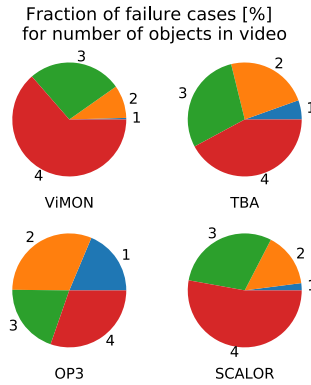


Figure E.2: Distribution of failure cases dependent on number of objects in VMDS videos for each model. Mean of three training runs.

Results are shown as the mean and standard deviation of three training runs with different random seed per model.

### E.1 Out-of-distribution test sets

To test whether the models can in principle learn additional object transformations as featured in the VMDS o.o.d. sets, we additionally train the models on a new training set that includes size and color changes as well as rotation of objects.

ViMON, OP3 and SCALOR are able to learn additional property changes of the objects when they are part of the training data while TBA fails to learn tracking on this more challenging data set (Fig. E.3; for absolute values Table E.4).

### E.2 Stability of training and runtime

To assess runtime in a fair way despite the models being trained on different hardware, we report the training progress of all models after one hour of training on a single GPU

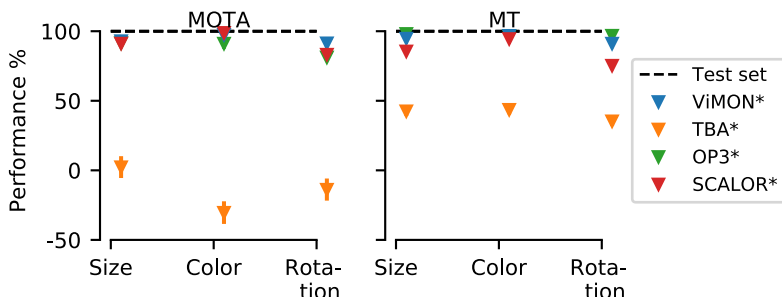


Figure E.3: Performance on out-of-distribution sets relative to VMDS test set (100%). \* indicates that models were trained on a data set that included color, size and orientation changes of objects.

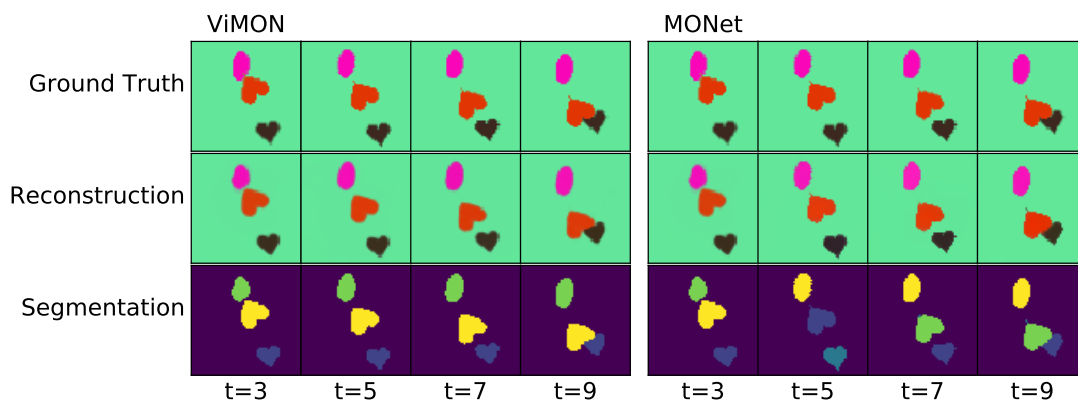


Figure E.4: Comparison of MONET and ViMON on VMDS. Example sequence of data set shown with corresponding outputs of the model. Reconstruction shows sum of components from all slots, weighted by the attention masks. Color-coded segmentation maps in third row signify slot-assignment. Note how the object-slot assignment changes for consecutive frames (3rd row) for MONET, while ViMON maintains a consistent slot assignment throughout the video.

(Table E.5). In addition, we quantify inference time on the full VMDS test set using a batch size of one.

### E.3 ViMON Ablations

Removing the GRU or the mask conditioning of the attention network reduces tracking performance (MOTA on VMDS from 86.8% to 70.6% and 81.4%, respectively; Table E.6)

### E.4 Impact of IoU Threshold for Matching

To examine whether the IoU threshold used for matching object predictions to ground truth impacts the ordering of the models, we compute all metrics with regard to the thresholds [0.1 .. 0.9] in steps of 0.1 on the VMDS test set. The ordering of the models with regard

## BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

Model	Occlusion			Same Color			Small Objects			Large Objects		
	MOTA	MOTP	MT	MOTA	MOTP	MT	MOTA	MOTP	MT	MOTA	MOTP	MT
ViMON	67.1 ± 0.4	82.5 ± 0.0	63.0 ± 0.1	72.2 ± 0.1	83.6 ± 0.1	70.4 ± 0.3	86.3 ± 0.2	83.3 ± 0.2	83.4 ± 0.4	70.7 ± 0.5	85.1 ± 0.1	76.1 ± 0.7
TBA	37.5 ± 10.4	72.8 ± 0.8	38.3 ± 4.6	47.2 ± 9.4	73.0 ± 0.7	45.2 ± 3.9	74.3 ± 0.7	71.9 ± 0.4	65.3 ± 1.6	25.6 ± 15.0	73.4 ± 0.9	44.7 ± 6.7
OP3	85.3 ± 1.0	91.6 ± 0.4	89.6 ± 0.9	51.5 ± 1.3	86.5 ± 0.3	66.3 ± 1.3	93.3 ± 1.6	93.0 ± 0.4	97.0 ± 0.2	83.8 ± 2.0	92.2 ± 0.4	93.5 ± 0.4
SCALOR	58.8 ± 1.0	86.6 ± 0.4	46.8 ± 1.2	53.7 ± 1.1	83.4 ± 0.3	46.2 ± 1.1	74.4 ± 0.7	86.1 ± 0.4	67.6 ± 1.3	66.1 ± 1.9	86.6 ± 0.5	62.4 ± 1.4

Table E.3: Performance on **VMDS challenge sets**. Results shown as mean  $\pm$  standard deviation for three runs with different random training seeds. Examples sequences for each challenge set shown below.

Model	Size				Color				Rotation			
	MOTA	MOTP	MD	MT	MOTA	MOTP	MD	MT	MOTA	MOTP	MD	MT
ViMON	61.4 ± 2.5	78.0 ± 0.3	71.3 ± 2.1	66.8 ± 1.9	87.4 ± 0.4	86.2 ± 0.2	86.4 ± 0.1	85.0 ± 0.2	-10.4 ± 4.0	70.5 ± 0.4	39.5 ± 2.6	29.8 ± 1.0
ViMON*	80.3 ± 0.9	82.1 ± 0.5	82.5 ± 0.4	79.8 ± 0.5	84.5 ± 0.6	84.6 ± 0.5	83.4 ± 0.5	81.8 ± 0.3	78.7 ± 1.6	82.0 ± 0.6	79.2 ± 0.4	76.4 ± 0.6
TBA	52.3 ± 8.7	73.3 ± 0.7	59.8 ± 4.9	51.8 ± 4.9	56.1 ± 11.4	75.1 ± 0.9	63.7 ± 5.4	59.0 ± 5.2	52.4 ± 9.9	73.6 ± 0.8	59.3 ± 6.2	49.8 ± 5.5
TBA*	1.3 ± 7.8	68.4 ± 1.9	30.6 ± 4.5	24.8 ± 3.4	-16.5 ± 8.1	69.6 ± 1.5	29.1 ± 3.8	25.4 ± 3.3	-7.5 ± 7.9	69.4 ± 1.4	26.6 ± 4.0	20.6 ± 3.4
OP3	87.0 ± 1.9	90.8 ± 0.4	96.4 ± 0.1	95.3 ± 0.1	90.8 ± 1.2	93.5 ± 0.5	97.3 ± 0.1	95.8 ± 0.1	54.7 ± 5.7	84.2 ± 0.7	87.1 ± 1.7	80.5 ± 2.5
OP3*	84.0 ± 2.8	91.2 ± 1.0	95.9 ± 0.8	94.5 ± 1.2	83.6 ± 3.7	91.6 ± 1.3	95.5 ± 0.5	92.9 ± 1.6	74.5 ± 2.2	89.8 ± 0.7	94.8 ± 0.6	93.3 ± 0.8
SCALOR	68.1 ± 1.7	84.9 ± 0.4	63.3 ± 1.7	60.0 ± 2.0	75.5 ± 1.1	89.9 ± 0.5	67.0 ± 1.4	65.7 ± 1.6	46.5 ± 1.8	82.1 ± 0.5	41.9 ± 1.7	37.1 ± 1.3
SCALOR*	67.5 ± 1.2	85.2 ± 0.6	61.2 ± 1.2	57.1 ± 0.7	73.3 ± 0.7	89.8 ± 0.5	64.8 ± 1.1	63.0 ± 0.9	61.6 ± 1.4	83.5 ± 0.4	53.4 ± 1.5	50.2 ± 1.1

\* Models trained on a data set that featured color, size and orientation changes of objects during the sequence.

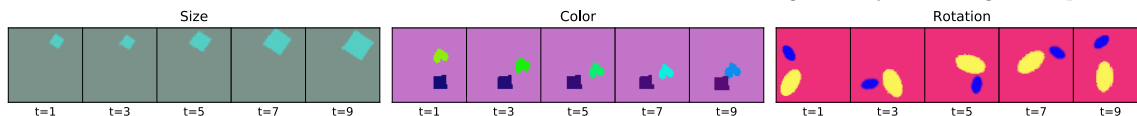


Table E.4: Performance on **VMDS OOD test sets**. Results shown as mean  $\pm$  standard deviation for three runs with different random training seeds. Examples sequences for each o.o.d. set shown below.

to their MOTA is consistent over all matching thresholds, but the performance difference between the models increases with higher thresholds (Fig. E.6). For low thresholds (0.1 – 0.3) TBA outperforms SCALOR in MD, MT, Matches and Misses but has a significantly lower segmentation performance as measured by MOTP.

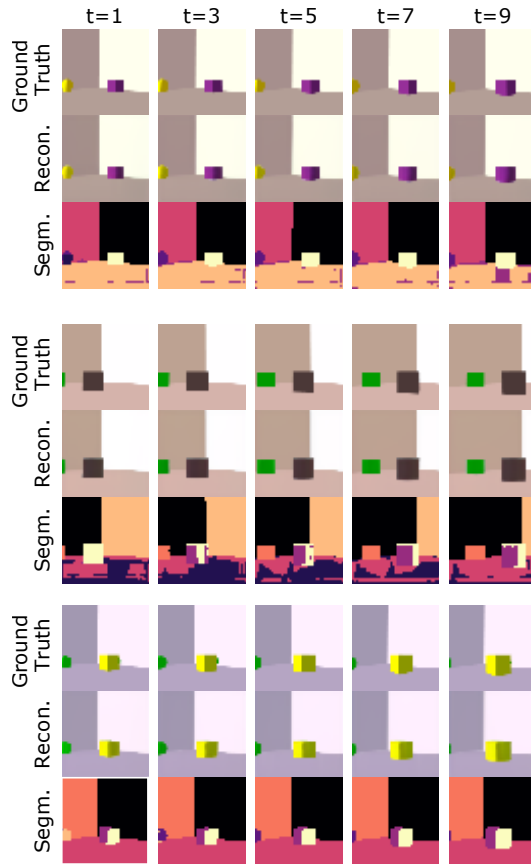


Figure E.5: Failure cases of OP3 on VOR. Example sequences of VOR test set shown with corresponding outputs of the model after final refinement step. Binarized color-coded segmentation maps in third row signify slot-assignment.

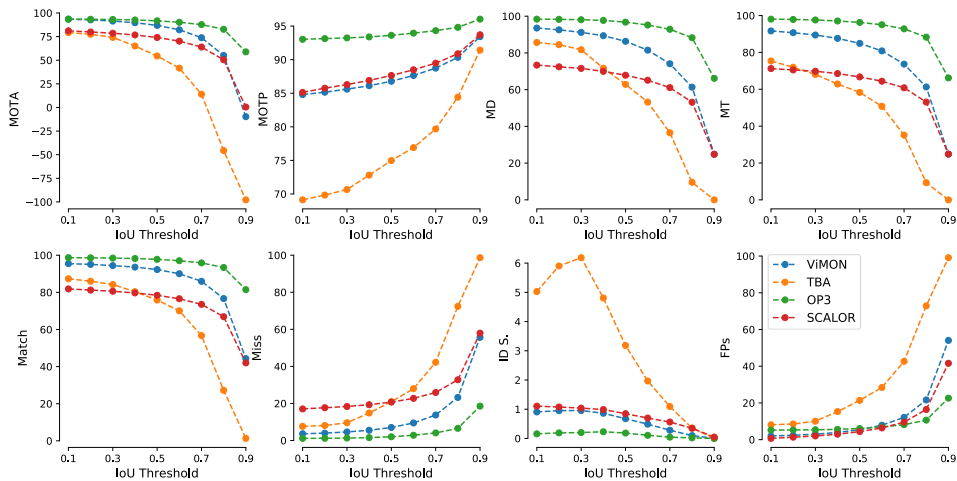


Figure E.6: Results of ViMON, TBA, OP3 and SCALOR on the VMDS test set using different IoU thresholds for matching ground truth objects to model predictions.

## BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

Model	Resolution	No. Param.	Training				Inference		
			Batch Size	Memory [MiB]	No. Iters	Epochs	Memory [MiB]	Avg. runtime / batch	Total runtime
ViMON	64×64	714,900	18	10,860	3687	6.63	910	0.28 s/it	4min 39s
TBA	128×128	3,884,644*	64	10,564	4421	28.29	972	0.24 s/it	4min 05s
OP3	64×64	876,305	10	10,874	2204	2.20	4092	0.54 s/it	9min 04s
SCALOR	64×64	2,763,526	48	10,942	2547	12.23	930	0.29 s/it	4min 48s

\* The TBA parameter count scales with the feature resolution, which is kept fixed using adaptive pooling. This makes the parameter count independent of input resolution.

Table E.5: Runtime analysis (using a single RTX 2080 Ti GPU). Training: models trained on VMDS for one hour. Inference: models evaluated on VMDS test set with batch size=1 (10 frames).

Model	MOTA ↑	MOTP ↑	MD ↑	MT ↑	Match ↑	Miss ↓	ID S. ↓	FPs ↓	MSE ↓
ViMON w/o MASK CONDITIONING	70.6	87.8	75.7	66.0	81.4	13.4	5.2	10.8	16.9
ViMON w/o GRU	81.4	86.9	79.8	77.3	88.2	10.3	1.4	6.8	18.9

Table E.6: Ablation experiments for ViMON on VMDS.

## Appendix F. Supplementary Figures

See figures F.5 – F.13 for additional, randomly picked examples of reconstruction and segmentation for K-MEANS, ViMON, TBA, OP3 and SCALOR on the four data sets (VMDS, SpMOT, VOR and texVMDS).

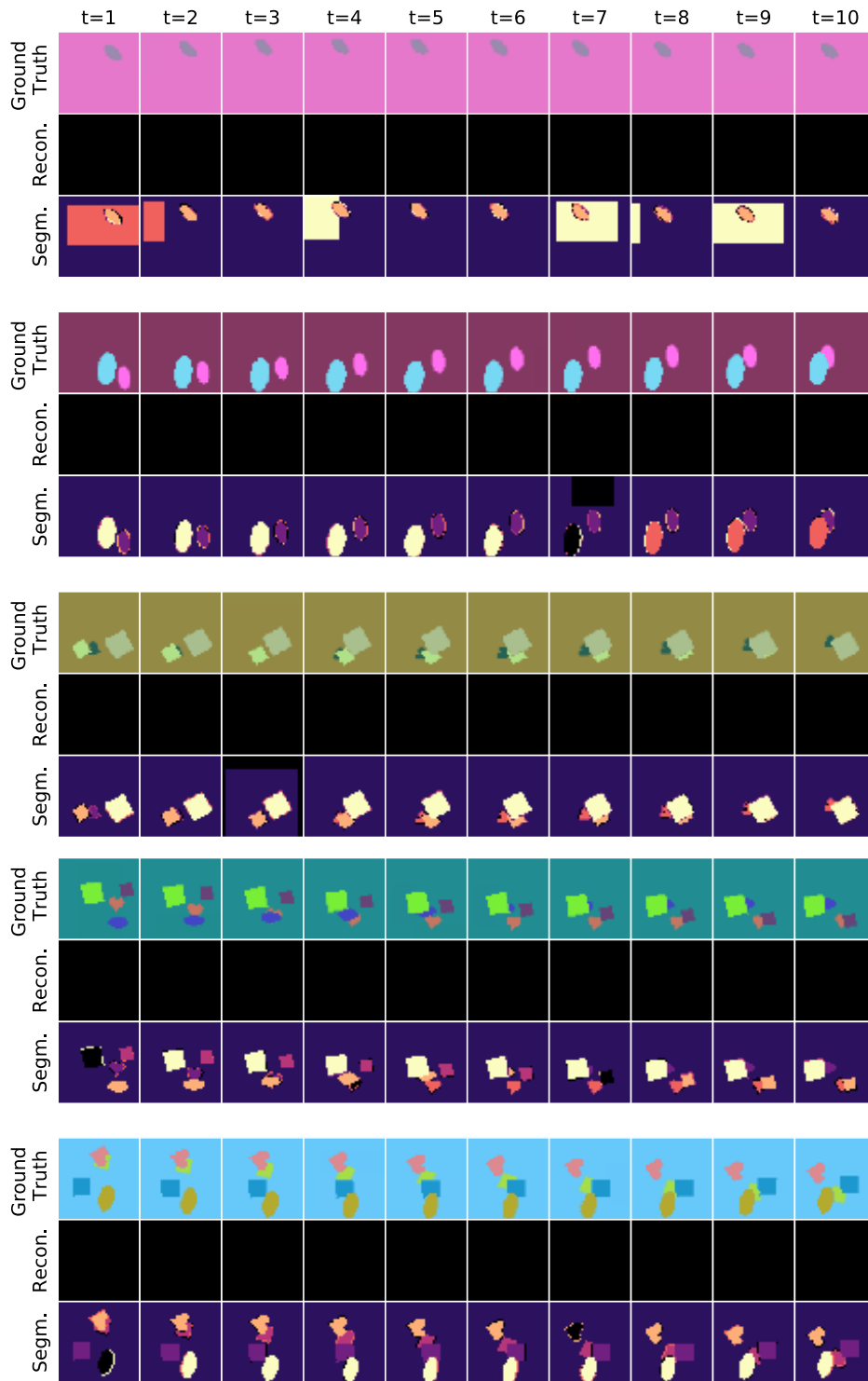


Figure F.1: Results of  $\kappa$ -MEANS on VMDS. Random example sequences of VMDS test set shown with corresponding outputs of the model. Note that  $\kappa$ -Means does not give a reconstruction of the input.

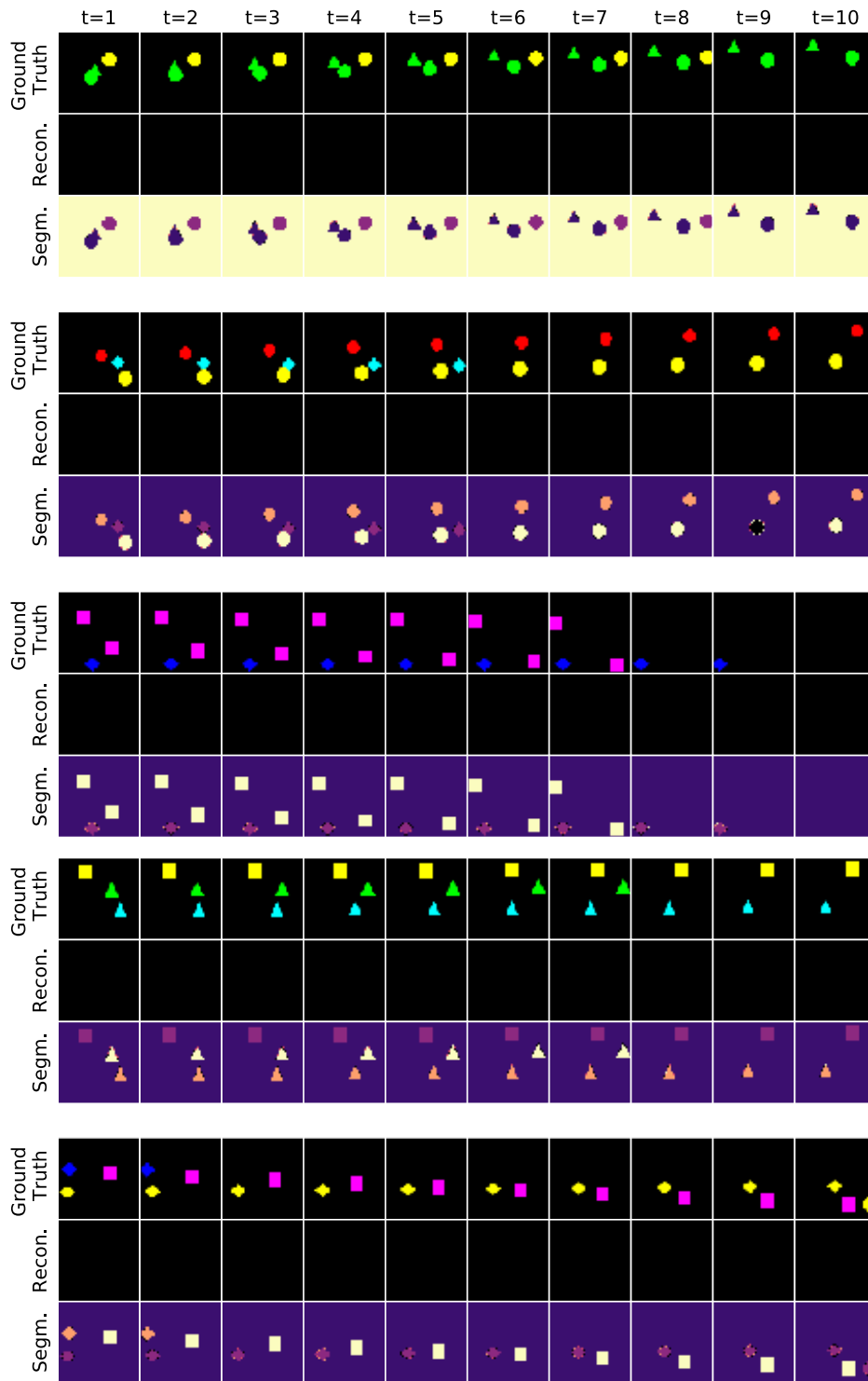


Figure F.2: Results of K-MEANS on SpMOT. Random example sequences of SpMOT test set shown with corresponding outputs of the model. Note that k-Means does not give a reconstruction of the input.

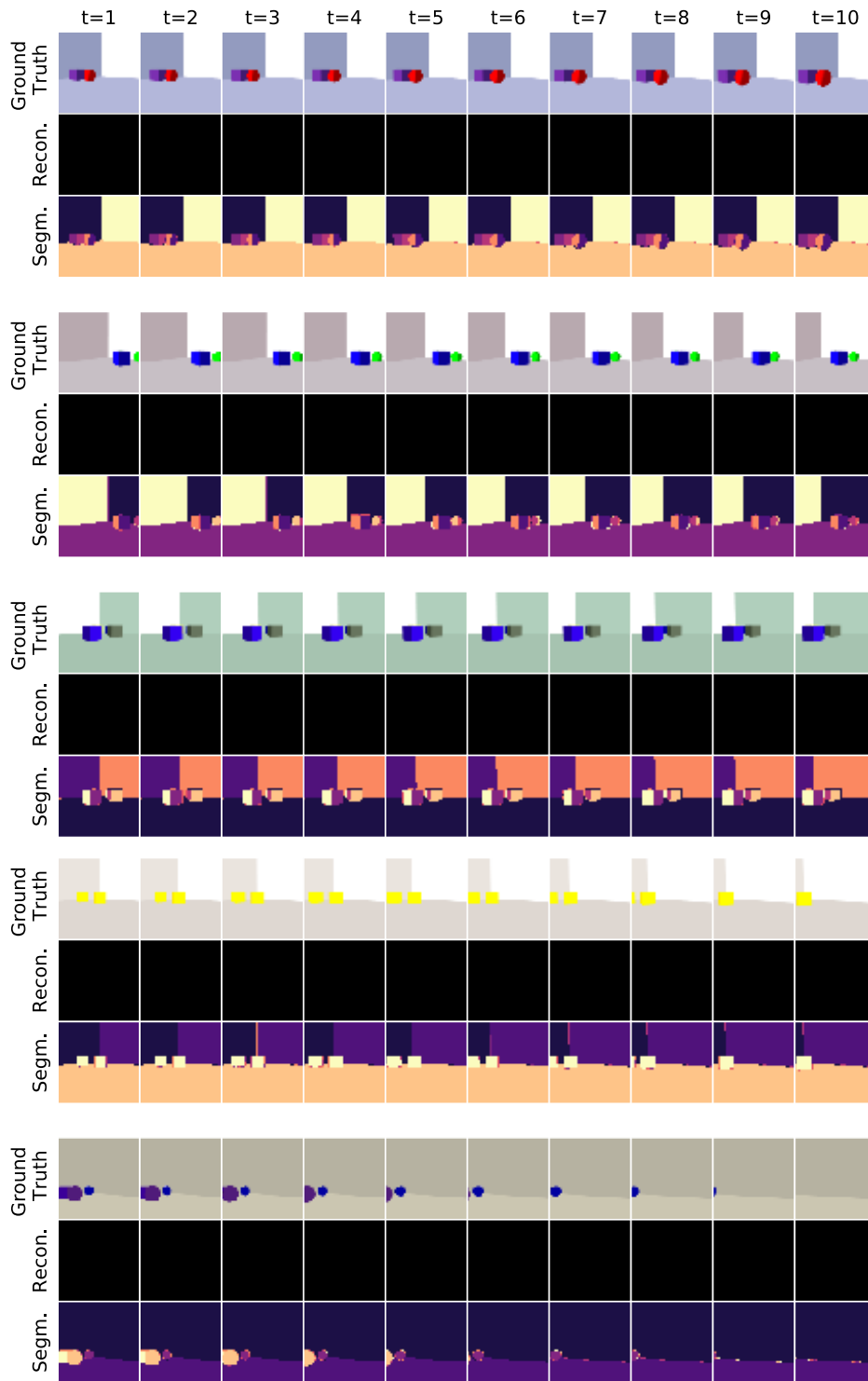


Figure F.3: Results of K-MEANS on VOR. Random example sequences of VOR test set shown with corresponding outputs of the model. Note that k-Means does not give a reconstruction of the input.



BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

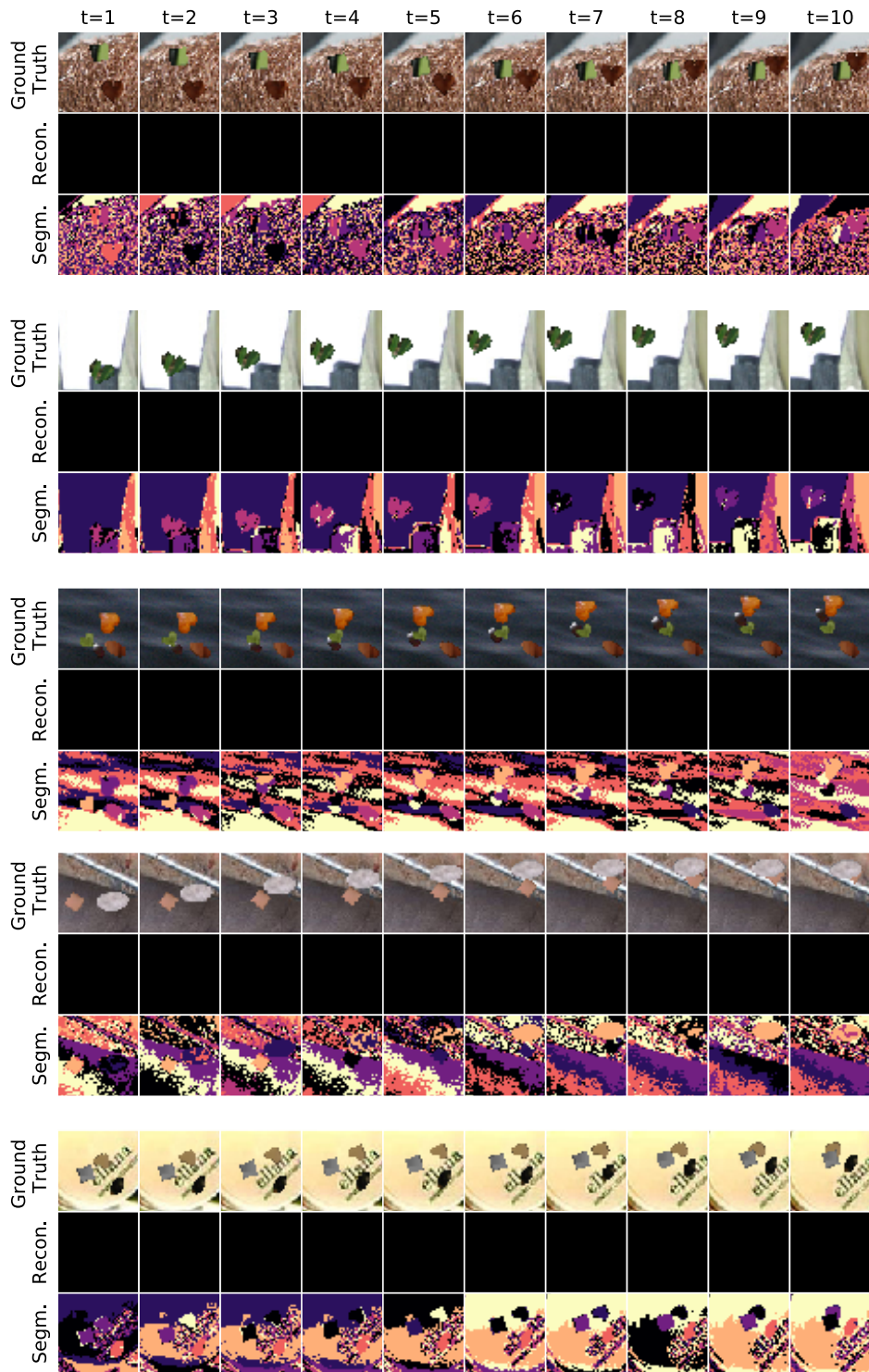


Figure F.4: Results of k-MEANS on texVMDS. Random example sequences of texVMDS test set shown with corresponding outputs of the model. Note that k-Means does not give a reconstruction of the input.

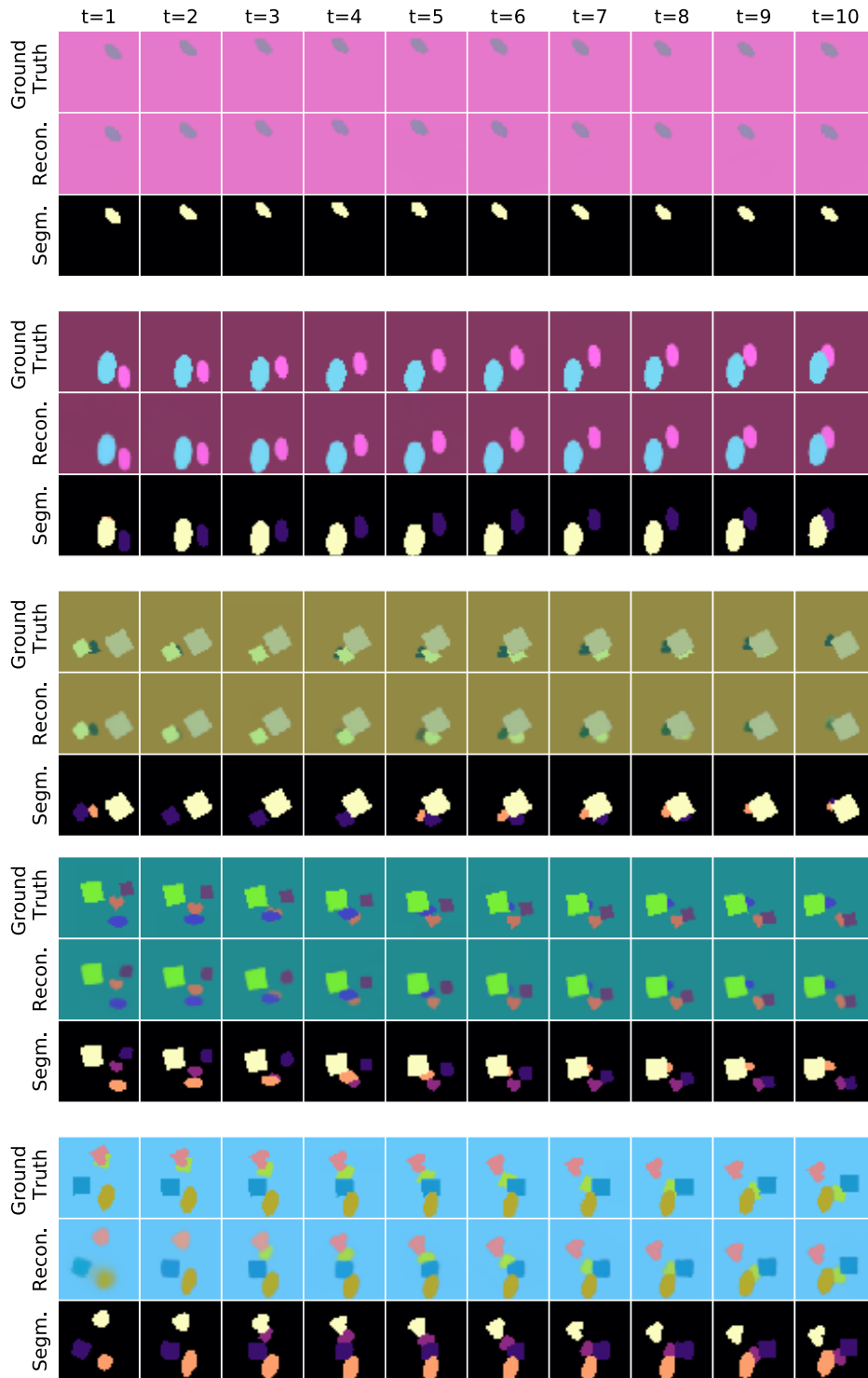


Figure F.5: Results of ViMON on VMDS. Random example sequences of VMDS test set shown with corresponding outputs of the model. Reconstruction shows sum of components from all slots, weighted by the reconstructed masks from the VAE. Binarized color-coded segmentation maps in third row signify slot-assignment.

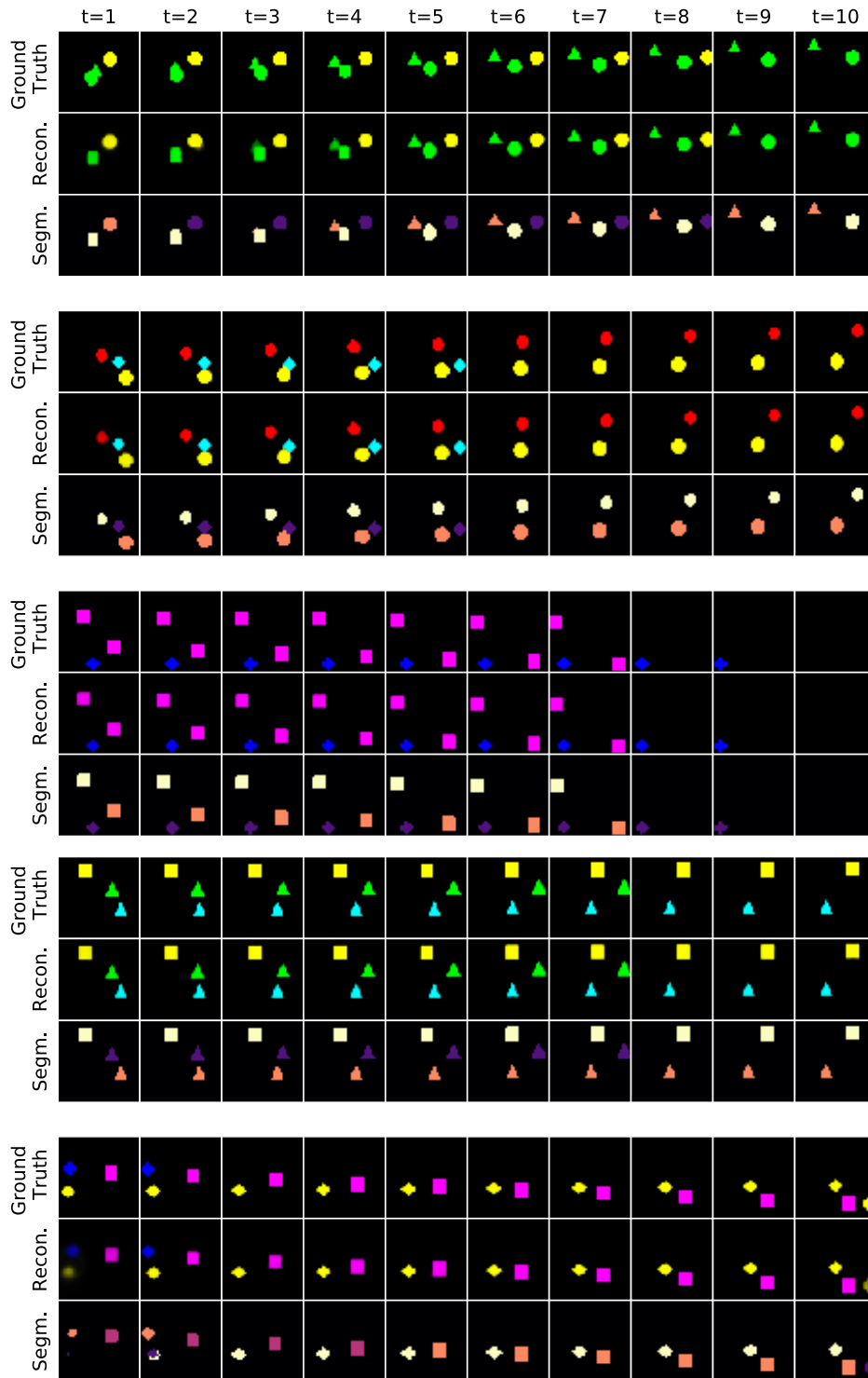


Figure F.6: Results of VIMON on SpMOT. Random example sequences of SpMOT test set shown with corresponding outputs of the model. Reconstruction shows sum of components from all slots, weighted by the reconstructed masks from the VAE. Binarized color-coded segmentation maps in third row signify slot-assignment.

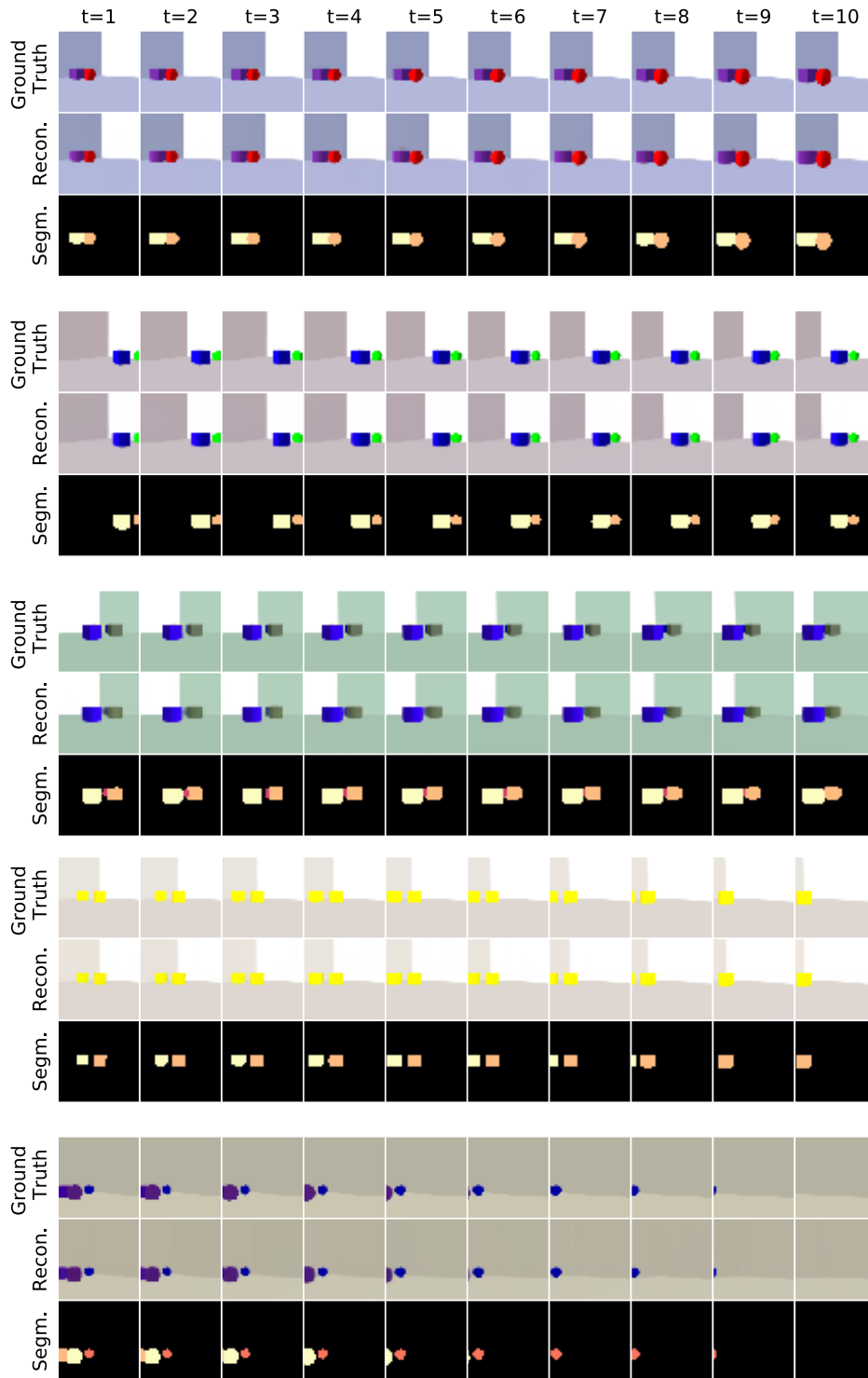


Figure F.7: Results of ViMON on VOR. Random example sequences of VOR test set shown with corresponding outputs of the model. Reconstruction shows sum of components from all slots, weighted by the reconstructed masks from the VAE. Binarized color-coded segmentation maps in third row signify slot-assignment.

BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

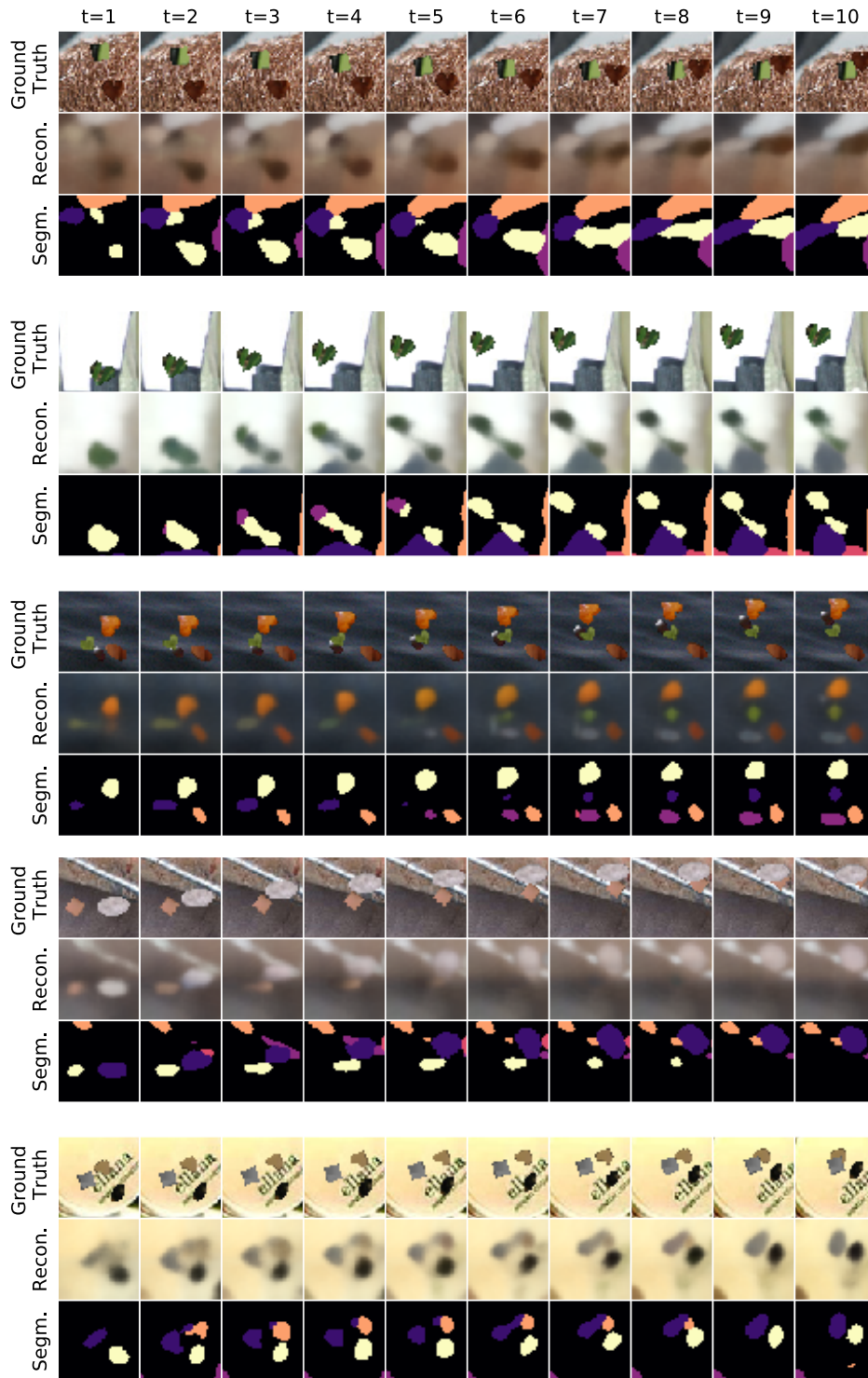


Figure F.8: Results of VIMON on texVMDS. Random example sequences of texVMDS test set shown with corresponding outputs of the model. Reconstruction shows sum of components from all slots, weighted by the reconstructed masks from the VAE. Binarized color-coded segmentation maps in third row signify slot-assignment.

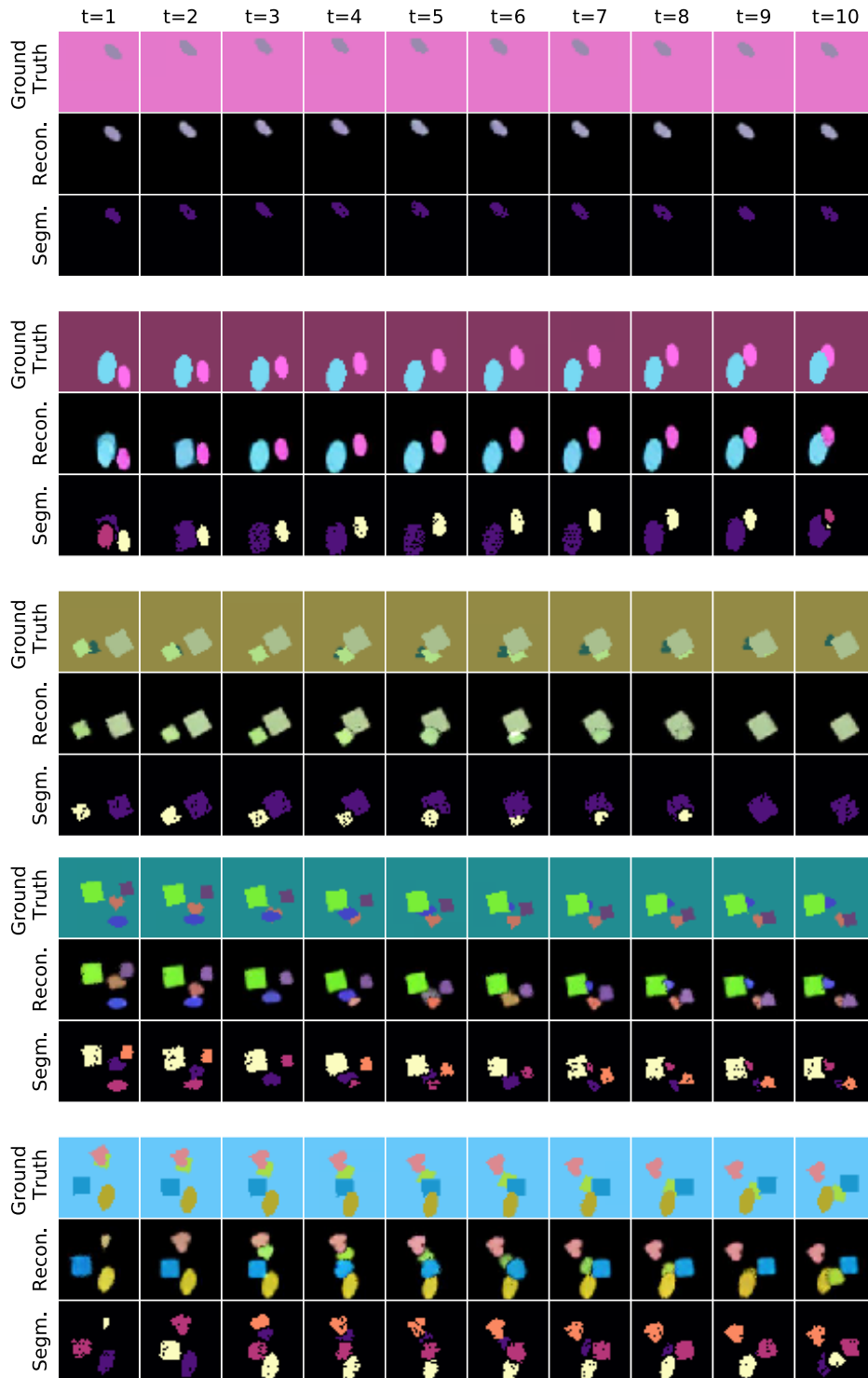


Figure F.9: Results of TBA on VMDS. Random example sequences of VMDS test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment. Note that background subtraction is performed in the preprocessing of TBA, hence the black background in the reconstructions.

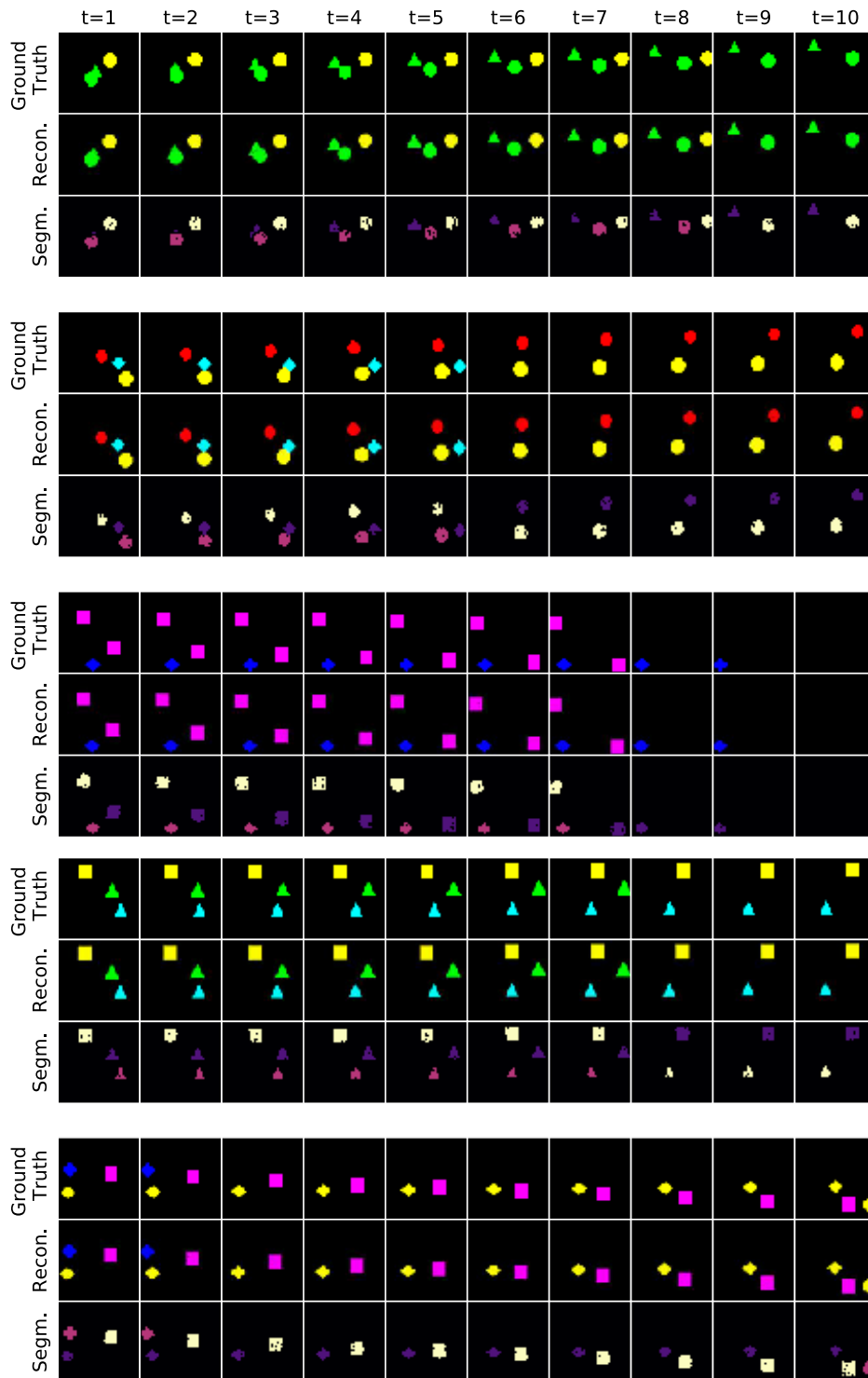


Figure F.10: Results of TBA on SpMOT. Random example sequences of SpMOT test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment.

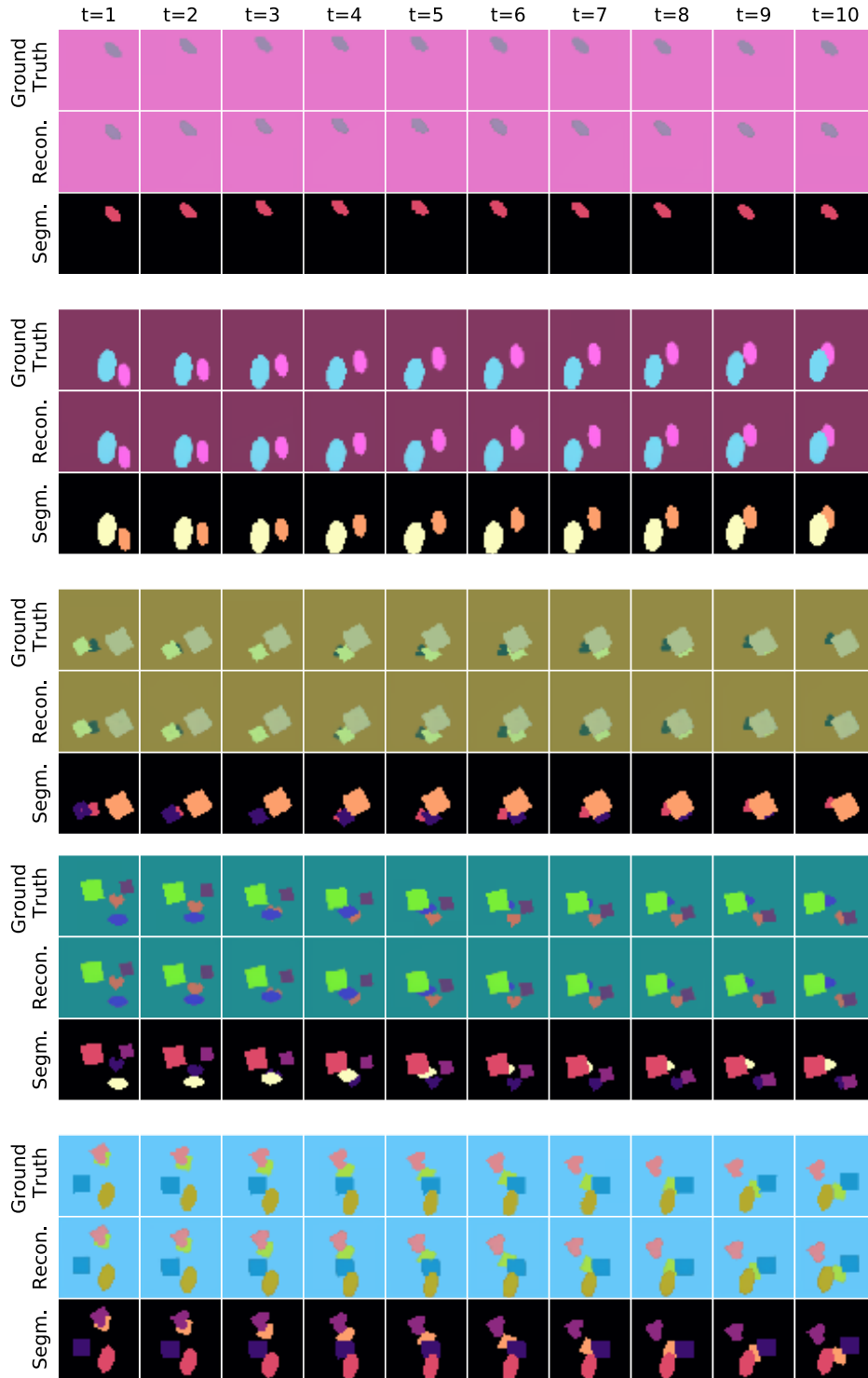


Figure F.11: Results of OP3 on VMDS. Random example sequences of VMDS test set shown with corresponding outputs of the model after final refinement step. Binarized color-coded segmentation maps in third row signify slot-assignment.



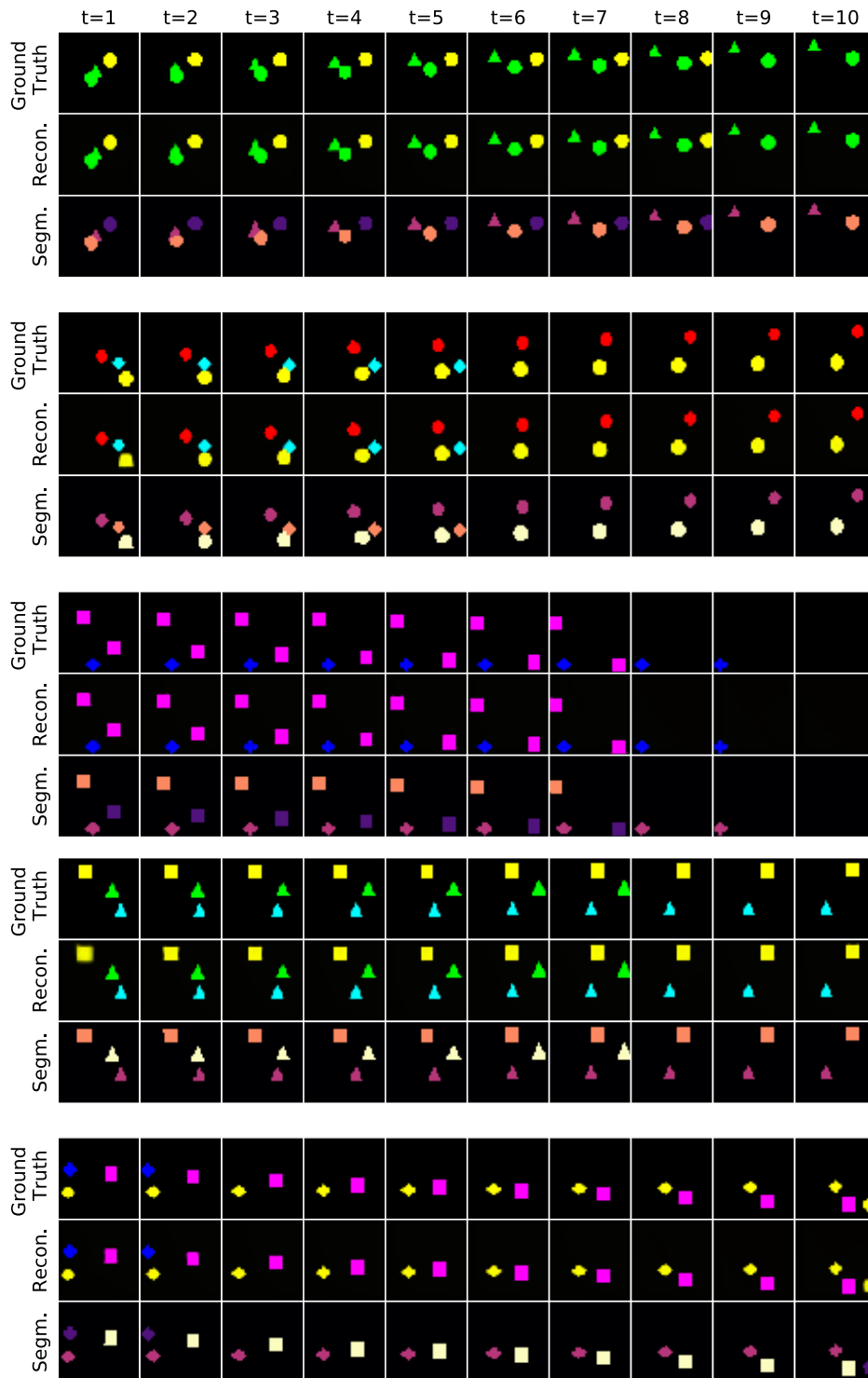


Figure F.12: Results of OP3 on SpMOT. Random example sequences of SpMOT test set shown with corresponding outputs of the model after final refinement step. Binarized color-coded segmentation maps in third row signify slot-assignment.

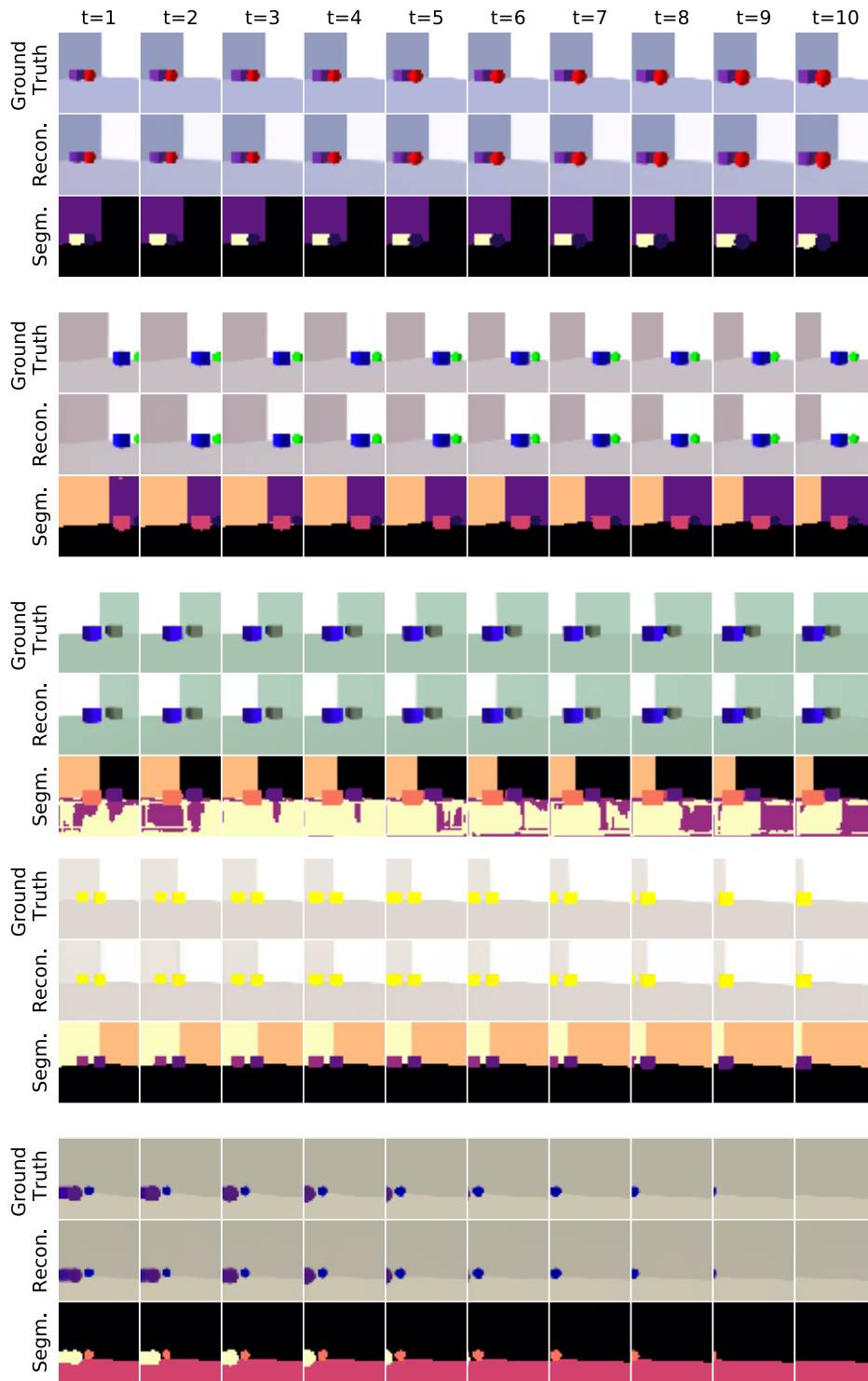


Figure F.13: Results of OP3 on VOR. Random example sequences of VOR test set shown with corresponding outputs of the model after final refinement step. Binarized color-coded segmentation maps in third row signify slot-assignment.

BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

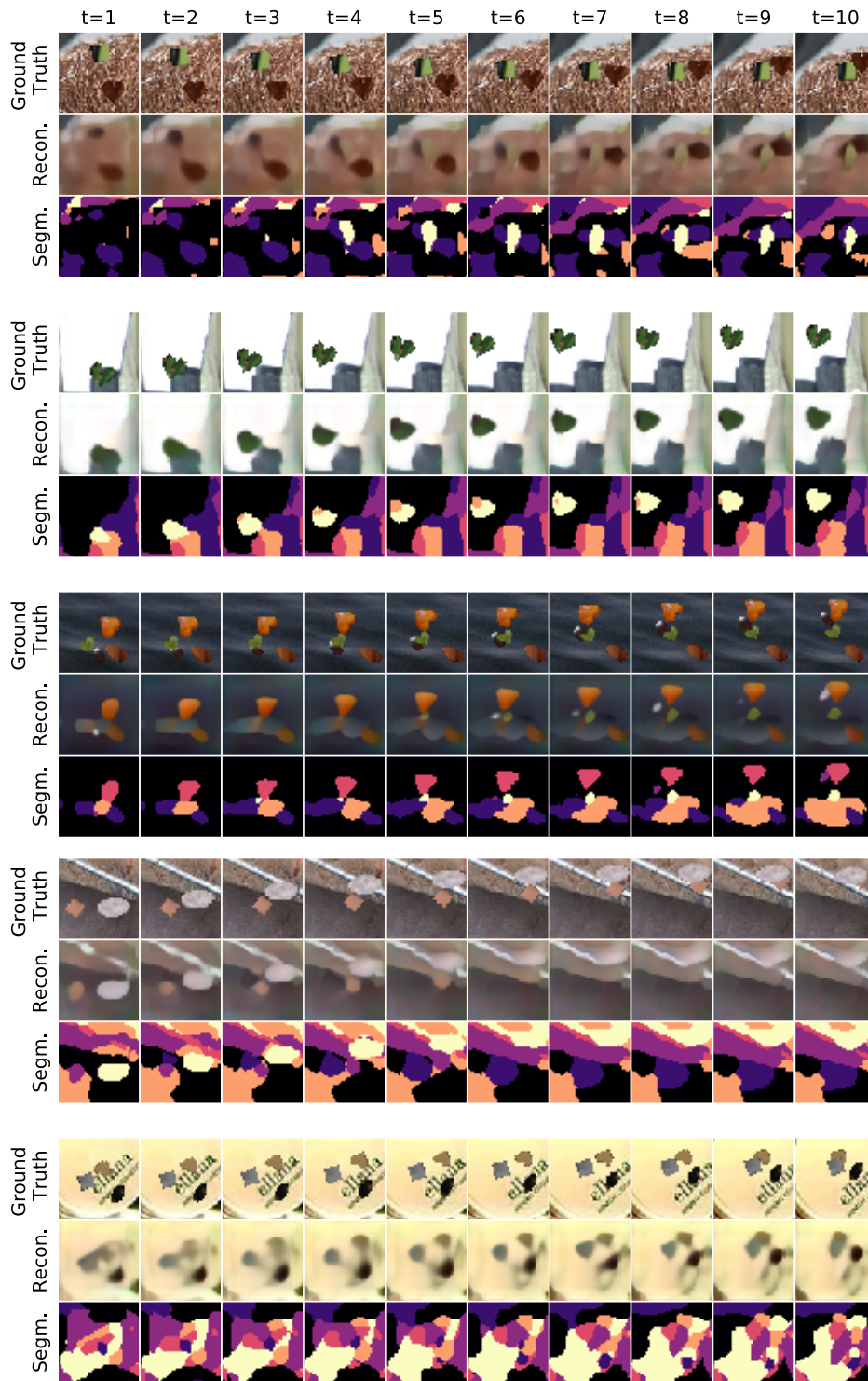


Figure F.14: Results of OP3 on texVMDS. Random example sequences of texVMDS test set shown with corresponding outputs of the model after final refinement step. Binarized color-coded segmentation maps in third row signify slot-assignment.

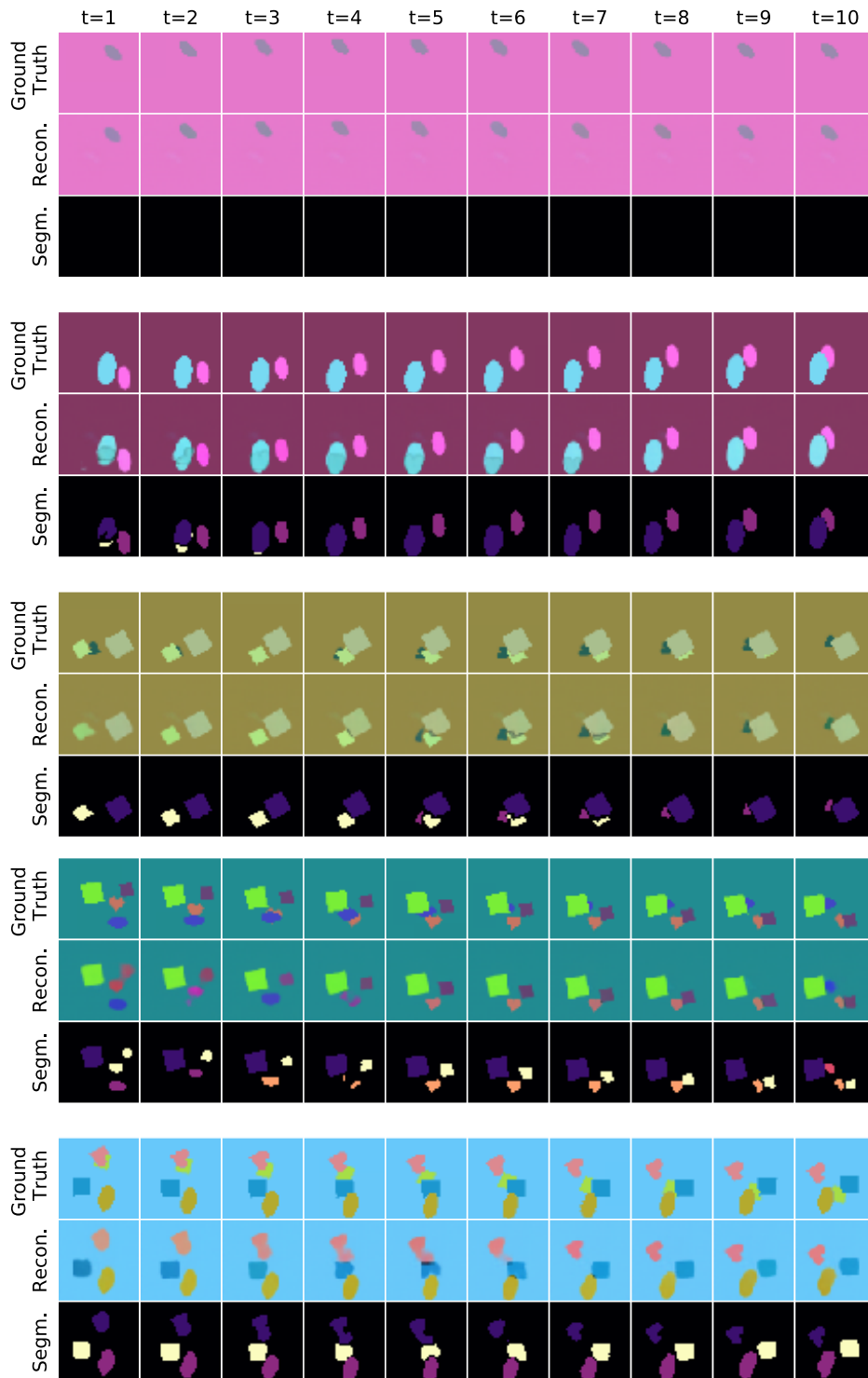


Figure F.15: Results of SCALOR on VMDS. Random example sequences of VMDS test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment.

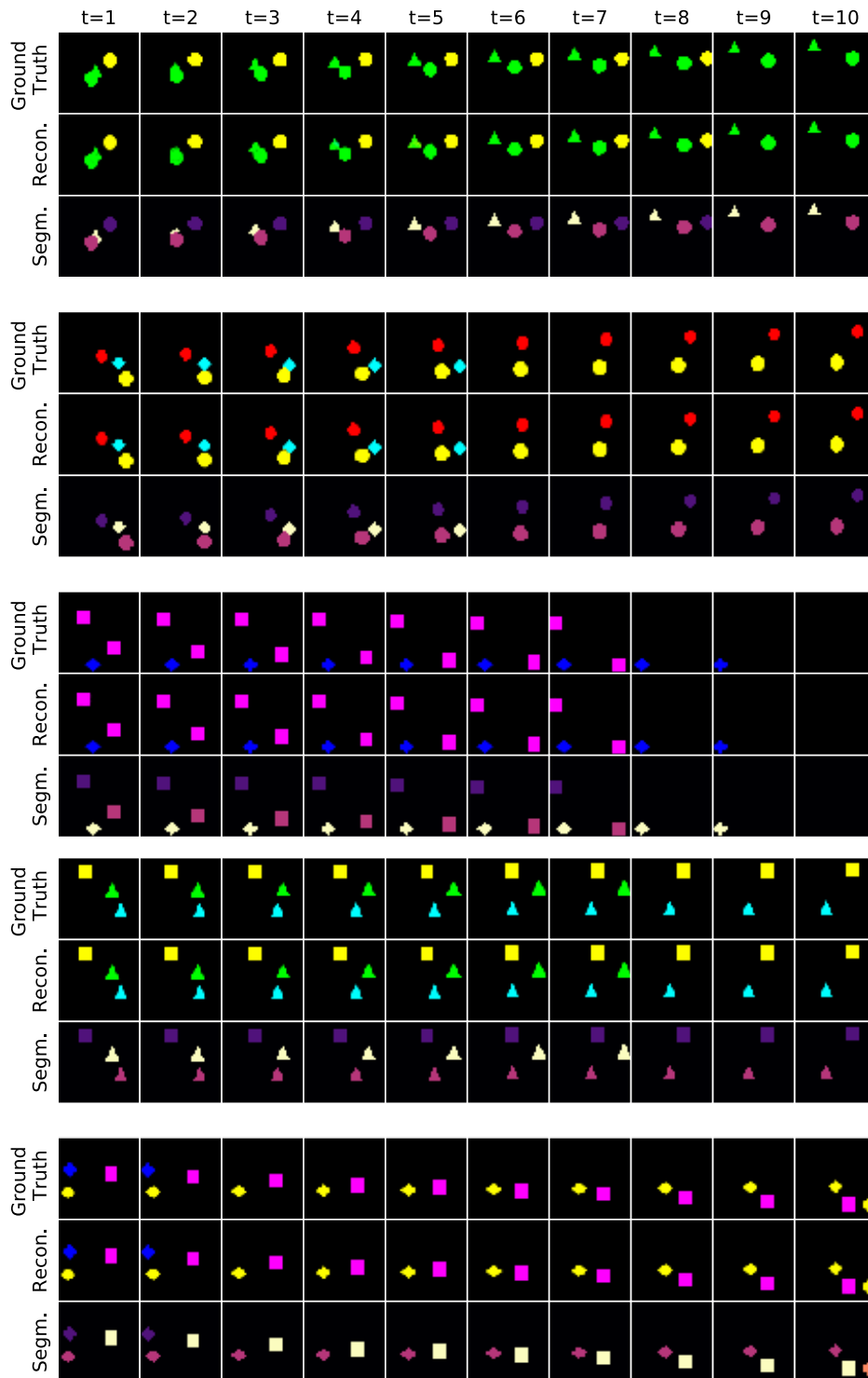


Figure F.16: Results of SCALOR on SpMOT. Random example sequences of SpMOT test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment.

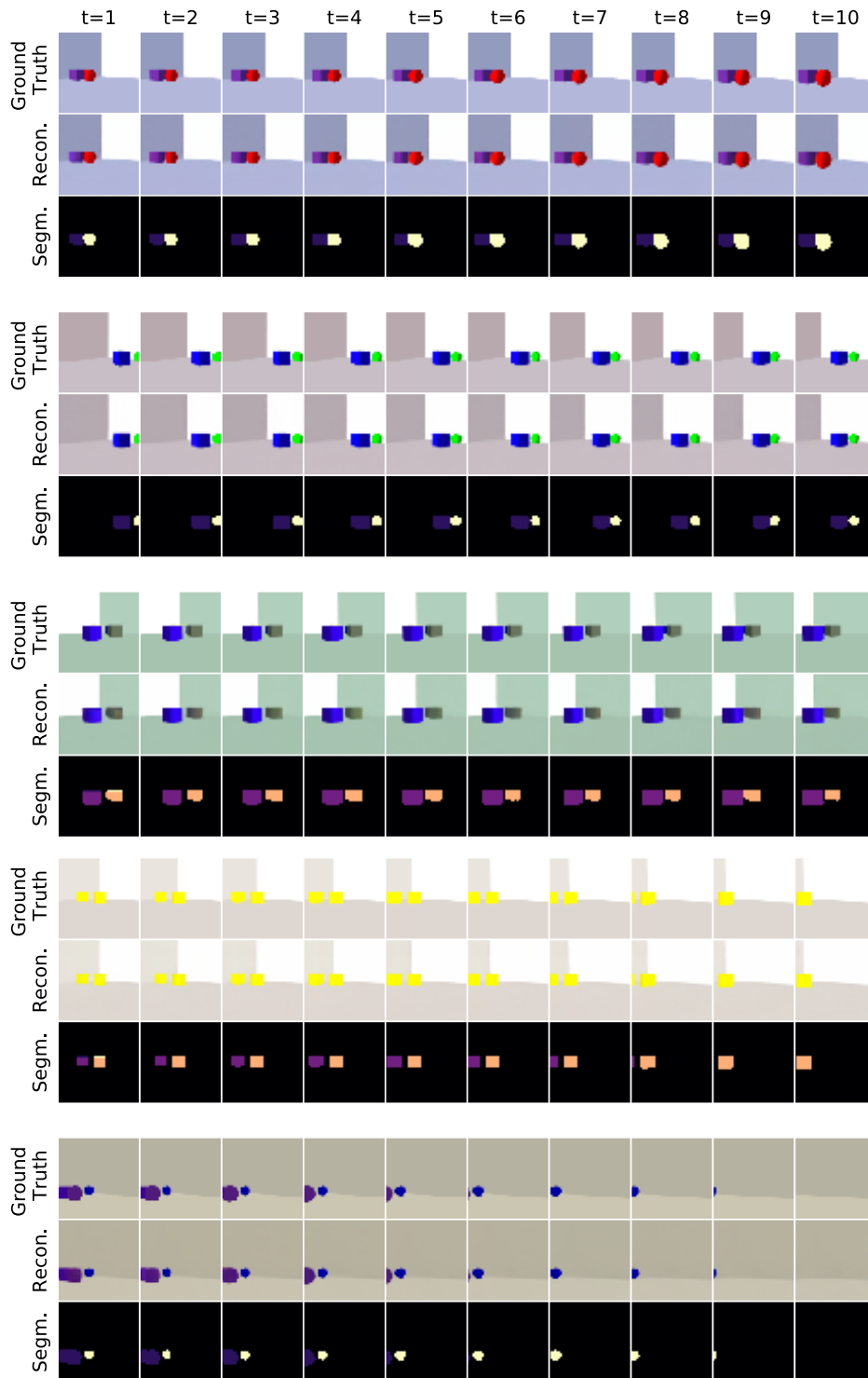


Figure F.17: Results of SCALOR on VOR. Random example sequences of VOR test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment.

BENCHMARKING UNSUPERVISED OBJECT REPRESENTATIONS

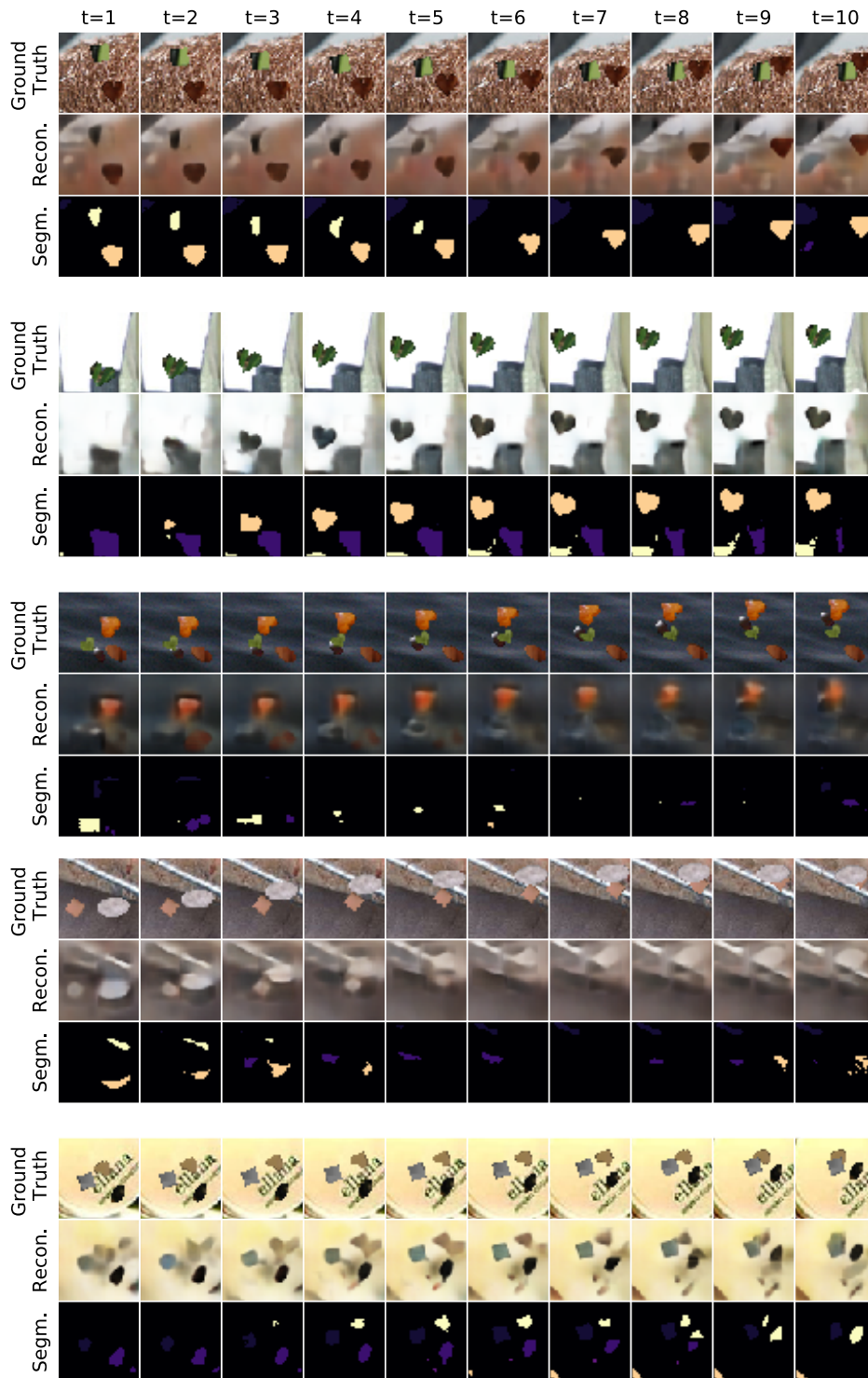


Figure F.18: Results of SCALOR on texVMDS. Random example sequences of texVMDS test set shown with corresponding outputs of the model. Binarized color-coded segmentation maps in third row signify slot-assignment.

## References

- Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008.
- Domenico Daniele Bloisi and Luca Iocchi. Independent multimodal background subtraction. In *CompIMAGE*, 2012.
- Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv.org*, 1901.11390, 2019.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014. doi: 10.3115/v1/D14-1179.
- Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2019.
- Antonia Creswell, Kyriacos Nikiforou, Oriol Vinyals, Andre Saraiva, Rishabh Kabra, Loic Matthey, Chris Burgess, Malcolm Reynolds, Richard Tanburn, Marta Garnelo, and Murray Shanahan. Alignnet: Unsupervised entity alignment. *arXiv.org*, 2007.08973, 2020.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- Sebastien Ehrhardt, Oliver Groth, Aron Monszpart, Martin Engelcke, Ingmar Posner, Niloy Mitra, and Andrea Vedaldi. Relate: Physically plausible multi-object scene synthesis using structured latent spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.
- Martin Engelcke, Adam R. Kosior, Oivi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv.org*, 1508.06576, 2015.
- Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2016.



- Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.
- Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019.
- Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv.org*, 2012.05208, 2020.
- Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3D. In *Advances in Neural Information Processing Systems (NeurIPS)* 33, 2020.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.
- X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141, 2017. doi: 10.1109/WACV.2017.131.
- Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2015.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017.
- Jindong Jiang and Sungjin Ahn. Generative neurosymbolic machines. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Scott P Johnson. Object perception. In *Oxford Research Encyclopedia of Psychology*. Oxford University Press, 2018.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2014.

- Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Adam Kosior, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- J. v. Kügelgen, I. Ustyuzhaninov, P. Gehler, M. Bethge, and B. Schölkopf. Towards causal generative scene models via competition of experts. In *ICLR 2020 Workshop "Causal Learning for Decision Making"*, 2020.
- H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017. doi: 10.1017/S0140525X16001837.
- Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *Proc. of the International Conf. on Machine Learning (ICML)*, 2018.
- David Marr. Vision: A computational investigation into the human representation and processing of visual information. *W. H. Freeman, San Francisco*, 1982.
- Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>, 2017.
- A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv.org*, 1603.00831, 2016.
- Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
- Elizabeth S. Spelke. Where perceiving ends and thinking begins: The apprehension of objects in infancy. *The Minnesota symposia on child psychology*, 20:197–234, 1988.
- Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007. doi: <https://doi.org/10.1111/j.1467-7687.2007.00569.x>.
- Tomer D. Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B. Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in Cognitive Sciences*, 21(9):649–665, 2017. ISSN 1364-6613. doi: <https://doi.org/10.1016/j.tics.2017.05.012>.
- Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018.
- Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Proc. Conf. on Robot Learning (CoRL)*, 2020.
- Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P. Burgess, and Alexander Lerchner. COBRA: Data-efficient model-based RL through unsupervised object discovery and curiosity-driven exploration. *arXiv.org*, 1905.09275, 2019a.
- Nicholas Watters, Loic Matthey, Christopher P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in VAEs. *arXiv.org*, 1901.07017, 2019b.
- Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Unsupervised discovery of parts, structure, and dynamics. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Jinyang Yuan, Bin Li, and Xiangyang Xue. Generative modeling of infinite occluded objects for compositional scene representation. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019.



# Self-supervised graph representation learning for neuronal morphologies

*The following 26 pages have been published as:*

Marissa A. Weis, Laura Hansel, Timo Lüddecke, and Alexander S. Ecker. Self-supervised graph representation learning for neuronal morphologies. *Transactions on Machine Learning Research (TMLR)*, 2023.

*A summary of the motivation, results, and discussion can be found in Section 3.2 on page 30.*

## Abstract

Unsupervised graph representation learning has recently gained interest in several application domains such as neuroscience, where modeling the diverse morphology of cell types in the brain is one of the key challenges. It is currently unknown how many excitatory cortical cell types exist and what their defining morphological features are. Here we present GRAPHDINO, a purely data-driven approach to learn low-dimensional representations of 3D neuronal morphologies from unlabeled large-scale datasets. GRAPHDINO is a novel transformer-based representation learning method for spatially-embedded graphs. To enable self-supervised learning on transformers, we (1) developed data augmentation strategies for spatially-embedded graphs, (2) adapted the positional encoding and (3) introduced a novel attention mechanism, AC-ATTENTION, which combines attention-based global interaction between nodes and classic graph convolutional processing. We show, in two different species and across multiple brain areas, that this method yields morphological cell type clusterings that are on par with manual feature-based classification by experts, but without using prior knowledge about the structural features of neurons. Moreover, it outperforms previous approaches on quantitative benchmarks predicting expert labels. Our method could potentially enable data-driven discovery of novel morphological features and cell types in large-scale datasets. It is applicable beyond neuroscience in settings where samples in a dataset are graphs and graph-level embeddings are desired.



# Self-Supervised Graph Representation Learning for Neuronal Morphologies

Marissa A. Weis<sup>1,2,\*</sup>, Laura Hansel<sup>1</sup>, Timo Lüddecke<sup>1</sup>, and Alexander S. Ecker<sup>1,3</sup>

<sup>1</sup>Institute of Computer Science and Campus Institute Data Science, University of Göttingen, Germany

<sup>2</sup>Institute for Theoretical Physics, University of Tübingen, Germany

<sup>3</sup>Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany

\*Correspondence: [marissa.weis@uni-goettingen.de](mailto:marissa.weis@uni-goettingen.de)

Reviewed on OpenReview: <https://openreview.net/forum?id=ThhMzfrd6r>

## Abstract

Unsupervised graph representation learning has recently gained interest in several application domains such as neuroscience, where modeling the diverse morphology of cell types in the brain is one of the key challenges. It is currently unknown how many excitatory cortical cell types exist and what their defining morphological features are. Here we present GRAPHDINO, a purely data-driven approach to learn low-dimensional representations of 3D neuronal morphologies from unlabeled large-scale datasets. GRAPHDINO is a novel transformer-based representation learning method for spatially-embedded graphs. To enable self-supervised learning on transformers, we (1) developed data augmentation strategies for spatially-embedded graphs, (2) adapted the positional encoding and (3) introduced a novel attention mechanism, AC-ATTENTION, which combines attention-based global interaction between nodes and classic graph convolutional processing. We show, in two different species and across multiple brain areas, that this method yields morphological cell type clusterings that are on par with manual feature-based classification by experts, but without using prior knowledge about the structural features of neurons. Moreover, it outperforms previous approaches on quantitative benchmarks predicting expert labels. Our method could potentially enable data-driven discovery of novel morphological features and cell types in large-scale datasets. It is applicable beyond neuroscience in settings where samples in a dataset are graphs and graph-level embeddings are desired.

## 1 Introduction

The brain is structured into different areas that contain diverse types of neurons (Ascoli et al., 2008). The morphology of cortical neurons is highly complex with widely varying shapes. Cell morphology has long been used to classify neurons into cell types (Ramón y Cajal, 1911), but characterizing neuronal morphologies is still a challenging open research question. Morphological analysis has traditionally been carried out by visual inspection (Ascoli et al., 2008; Defelipe et al., 2013) or by computing a set of predefined, quantitatively measurable features such as number of branching points (Uylings & Van Pelt, 2002; Scorcioni et al., 2008; Oberlaender et al., 2012; Polavaram et al., 2014; Markram et al., 2015; Lu et al., 2015; Gouwens et al., 2019). However, both approaches have deficits: expert assessments have a high variance (Defelipe et al., 2013) and the manual definition of morphological features introduces biases (Wang, 2018), thus calling for more unbiased, data-driven approaches to characterize the morphology of neurons.

Recent advances in recording technologies have greatly accelerated data collection and therefore the amount of data available (MICrONS Consortium et al., 2023; Ramaswamy et al., 2015; Scala et al., 2021; Allen Institute, 2016; Peng et al., 2021; Winnubst et al., 2019). These developments have opened the floor for data-driven approaches based on unsupervised machine learning methods (Schubert et al., 2019,

Elabbady et al., 2022). One form of data representation that is particularly suitable for neurons is representing the skeleton of a neuron as a tree. In such a tree, the root node represents the neuron’s cell body and the node features are their 3D locations. The availability of a number of such skeleton datasets has recently sparked some work on graph-level representation learning of neuronal morphologies (Laternus & Berens, 2021; Zhao et al., 2022; Chen et al., 2022a). Following this line of research and work from the graph learning community (Sun et al., 2020; You et al., 2020), we present an unsupervised graph-level representation learning approach.

Our contributions in this paper are fourfold:

1. We propose a new self-supervised model to learn graph-level embeddings for spatial graphs. Unlike previous methods, our approach does not require human annotation or manual feature definition.
2. We introduce a novel attention module that combines transformer-style attention and message passing between neighboring nodes as in graph neural networks.
3. We apply this approach to the classification of excitatory neuronal morphologies and show that it produces clusters that are comparable with known excitatory cell types obtained by manual feature-based classification and expert-labeling.
4. We outperform existing approaches based on manual feature engineering and auto-encoding in predicting expert labels.

Our code is available at <https://eckerlab.org/code/weis2023/>.

## 2 Related Work

### 2.1 Representation learning for neuronal morphologies

Morphology has been used for a long time to classify neurons by either letting experts visually inspect the cells (Ramón y Cajal, 1911; Defelipe et al., 2013) or by specifying expert-defined features that can be extracted and used as input to a classifier (Oberlaender et al., 2012; Markram et al., 2015; Kanari et al., 2017; Wang, 2018; Kanari et al., 2019; Gouwens et al., 2019) (see Armañanzas & Ascoli (2015) for review). Ascoli et al. (2008) made an effort to unify the used expert-defined features.

With the advent of new technologies for microscopic imaging, electrical recording, and molecular analysis such as Patch-seq (Cadwell et al., 2015) that allow the simultaneous recording of transcriptomy, electrophysiology and morphology of whole cells, several works have explored the prediction of cell types from multiple modalities (Gala et al., 2021) or one modality from the other (Cadwell et al., 2015; Scala et al., 2021; Gouwens et al., 2020).

Multiple previous works try to either hand-engineer or learn a representation of neuronal morphologies. Laternus & Berens (2021) propose a generative approach involving random walks in graphs to model neuronal morphologies. Schubert et al. (2019) process 2D projections of morphologies with a convolutional neural network (CNN) to learn low-dimensional representations. Seshamani et al. (2020) extract local mesh features around spines and combine them with traditional Sholl analysis (Sholl, 1953). Gouwens et al. (2019) define a set of morphological features based on graphs and perform hierarchical clustering on them. We use the latter as a baseline for a classical approach with pre-defined features and Laternus & Berens (2021) as a baseline of a model with learned features.

Concurrent work (Zhao et al., 2022) proposes a contrastive graph neural network to learn neuronal embeddings with a focus on retrieval efficiency from large-scale databases. Elabbady et al. (2022) learn representations of neurons based on subcellular features of the somatic region of the neurons and show that those features are sufficient for classifying cell types on large-scale EM datasets. Chen et al. (2022b) propose a combination of graph-based processing and manually-defined features to learn embeddings of neuronal morphologies using a LSTM-based network and contrastive learning. We compare to the latter in Section 5.8.



## 2.2 Graph Neural Networks (GNNs)

Graph neural networks learn node representations by recursively aggregating information from adjacent nodes as defined by the graph’s structure. While early approaches date back over a decade (Scarselli et al., 2009), recently numerous new variants were introduced for (semi-) supervised settings: relying on convolution over nodes (Duvenaud et al., 2015; Hamilton et al., 2017; Kipf & Welling, 2017), using recurrence (Li et al., 2016), or making use of attention mechanisms (Veličković et al., 2018). A representation for the whole graph is often derived by a readout operation on the node representations, for instance averaging. See Dwivedi et al. (2020) for a recent benchmark on graph neural network architectures.

**Transformer-based GNNs.** Similar to us, Zhang et al. (2020) and Dwivedi & Bresson (2021) use transformer attention to work with graphs. However, Zhang et al. (2020) compute transformer attention over the nodes of sampled subgraphs, while Dwivedi & Bresson (2021) compute the attention only over local neighbors of nodes, which boils down to a weighted message passing that is conditioned on node feature similarity, and trains with supervision. Unlike these previous approaches, we compute the attention between nodes of the global graph and adapt the transformer attention to consider the adjacency matrix of the graph, which allows the model to take into account both the direct neighbors of a node as well as all other nodes in the graph. Mialon et al. (2021) consider encoding local sub-structures into their node features and leverage kernels on graphs in their attention as relative positional encodings. Their 1-step random walk (RW) kernel is similar to our AC-ATTENTION mechanism, except that the influence of the adjacency in their attention is not learnable. Ying et al. (2021) propose strategies to adapt positional encodings to graphs in order to leverage the structural information of the graphs with transformer attention. Specifically, they propose to use three different structural encodings: (1) a centrality encoding based on the node degree; (2) edge encodings based on the edge features and (3) a spatial encoding based on the shortest path between two nodes. For neural skeletons, the centrality encoding is not effective as all the nodes besides the soma have a node degree of two or three. Furthermore, the edge encoding is not applicable since in the neuronal graphs do not have edge features. We use Laplacian positional encodings instead as it was shown that they are beneficial to capture structural and positional information (Dwivedi & Bresson, 2021) and outperform previously proposed positional encodings (Zhang et al., 2020). We did not use any additional positional encodings such as shortest-path encodings (Ying et al., 2021), but they could be easily integrated into our model. Concurrent to our work, Rampášek et al. (2022) proposed a two-stream architecture, in which transformer attention and message passing are computed in parallel and then combined after each block. In contrast, we propose one combined attention mechanism that subsumes transformer attention and message passing with a learned trade-off per node between the two settings. Chen et al. (2022a) incorporate structural information into the transformer attention by extracting a subgraph representation around each node before computing attention over nodes.

**Self-supervised learning on graphs.** Self-supervised learning has proven to be a useful technique for training image feature extractors (Oord et al., 2018; Chen et al., 2020; Chen & He, 2021; Caron et al., 2021) and has been investigated for learning graph (Li et al., 2016; Hassani & Khasahmadi, 2020; Qiu et al., 2020; You et al., 2020; Xu et al., 2021) and node (Veličković et al., 2019) representations. Narayanan et al. (2017) learn graph representations through skip-gram with negative sampling by predicting present sub-graphs. You et al. (2020) propose four data augmentations for contrastive learning of graph-level embeddings. Sun et al. (2020) learn graph-level representations in a contrastive way, by predicting if a subgraph and a graph representation originate from the same graph. Similarly, Hassani & Khasahmadi (2020) put node features of one view in contrast with the graph encoding of a second view and vice versa. They build on graph diffusion networks (Klicpera et al., 2019) and only augment the structure of the graph but not the initial node features. We use Sun et al. (2020) and You et al. (2020) as a baseline for graph-level unsupervised representation learning. Qiu et al. (2020) propose a generic pre-training method which uses an InfoNCE objective (Oord et al., 2018) to learn features by telling augmented versions of one subgraph from other subgraphs with random walks as augmentations. Xu et al. (2021) aim to capture local and global structures for whole-graph representation learning. They rely on an EM-like algorithm to jointly train the assignment of graphs to hierarchical prototypes, the GNN parameters and the prototypes. Zhu et al. (2021) propose adaptive augmentation, which considers node centrality and importance to generate graph views in a contrastive

framework. Similar to our approach, Thakoor et al. (2022) use two encoders of which only one is trained and the other is an exponential moving average of the first. In contrast to our approach, though, their training objective encourages the *node embeddings* of two augmented versions of the same graph to be similar – not the *graph-level* embedding. Moreover, they use node feature and edge masking as graph augmentations.

Unlike most prior work, we contrast two global views of a graph in order to learn a whole-graph representation. Our method operates on spatially embedded graphs, in which nodes correspond to points in 3D space. We make use of this knowledge in the choice of augmentations.

### 3 GraphDINO

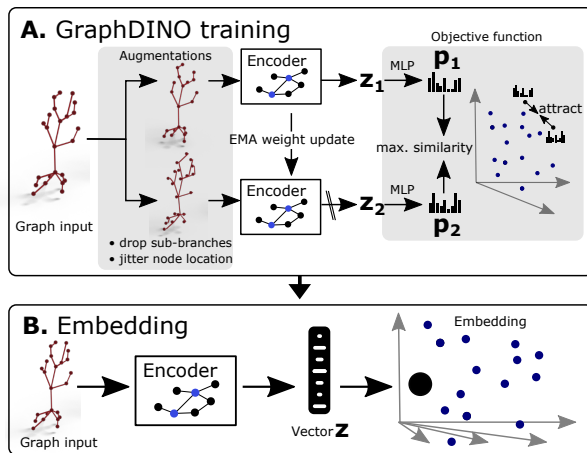


Figure 1: **A.** Self-supervised learning of low dimensional vector embeddings  $z_1, z_2$  that capture the essence of the 3D morphology of individual neurons using GraphDINO. Two augmented “views” of the neuron are input into the network, where the weights of one encoder (bottom) are an exponential moving average (EMA) of the other encoder (top). The resulting latent embeddings  $z$  are projected to probability distributions  $p$  by a MLP. The objective is to maximize the similarity between  $p_1$  and  $p_2$ . **B.** An individual neuron is represented by its vector embedding as a point in the  $D$ -dimensional vector space.

More specifically, we introduce the following modifications: (1) we incorporate the graph’s adjacency matrix into the attention computation; (2) we use the graph Laplacian as positional encoding; (3) we define augmentations suitable for spatial graphs.

**Input.** Input to the network is the 3D shape of a neuron which is represented as an undirected graph  $G = (V, E)$ .  $V$  is the set of nodes  $\{v_i\}_{i=1}^N$  and  $E$  the set of undirected edges  $E = \{e_{ij} = (v_i, v_j)\}$  that connect two nodes  $v_i, v_j$ . The features of each node  $v_i$  in the graph are encoded into a token using a linear transformation. These tokens are then used as input to the transformer model, which consists of  $l$  multi-head attention modules with  $h$  heads each.

**Attention bias.** Key-value query attention became popular in natural language modelling (Vaswani et al., 2017) and is now used routinely also in image models (Dosovitskiy et al., 2020).

We propose GRAPHDINO, a method for self-supervised representation learning of graphs. It is inspired by recent progress in self-supervised representation learning of images that has been shown to be competitive to supervised learning without relying on labels. The core idea is to enforce that the representations of two augmented versions of the same image are close to each other in latent space.

DINO (Caron et al., 2021) is an implementation of this self-supervised learning framework consisting of two encoders with a transformer backbone. To avoid mode collapse, only one encoder is directly trained through backpropagation (student) while the weights of the other encoder (teacher) are an exponential moving average (ema) of the student’s weights. The latent representations  $z \in \mathbb{R}^{D_1}$  given by the encoders are mapped to probability distributions  $p \in \mathbb{R}^{D_2}$  by a multi-layer perceptron (MLP) and subsequent softmax operator over which the cross-entropy loss is computed (Fig. C.4). For further explanation of DINO see Appendix C.1.

GRAPHDINO adapts this self-supervised framework to the data domain of graphs (Fig. 1). In order to use information given by the connectivity of the graph, we modify the computation of the transformer attention to take the graph adjacency matrix into account and use the graph Laplacian as positional encoding.

To make use of the information given by the adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  of the input graph — i.e. the neighborhood of nodes —, we bias the attention towards  $\mathbf{A}$  by adding a learned bias to the attention matrix that is conditioned on the input token values:

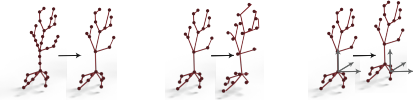

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{A}) = \sigma \left( \lambda \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \gamma \mathbf{A} \right) \mathbf{V}, \quad \text{with } [\lambda_i, \gamma_i] = \exp(\mathbf{W}x_i), \quad (1)$$

where  $\mathbf{K}, \mathbf{Q}, \mathbf{V}$  are the keys, queries and values which are computed as a learned linear projection of the tokens.  $\sigma(\cdot)$  denotes the softmax function.  $x_i \in \mathbb{R}^D$  is the token of node  $v_i$ ,  $\mathbf{W} \in \mathbb{R}^{2 \times D}$  is a learned weight matrix,  $\lambda, \gamma \in \mathbb{R}^N$  are two factors per node that trade off how much weight is assigned to neighboring nodes versus all other nodes in the graph, and  $N$  is the number of nodes.

When  $\gamma = 0$  and  $\lambda = 1$ , the adjacency-conditioned attention (AC-ATTENTION) reduces to regular transformer attention. In the other extreme case ( $\lambda = 0$ ), the attention matrix is dominated by  $\mathbf{A}$  and the transformer attention computation is akin to the message passing algorithm that is commonly used when working with graphs (Scarselli et al., 2009; Duvenaud et al., 2015; Gilmer et al., 2017). GRAPHDINO is more flexible than both regular message passing and point-cloud attention since it can decide how much weight is given to the neighbors of a node while maintaining the flexibility to attend to all other nodes in the graph as well.

**Positional encoding.** Following Dwivedi et al. (2020), we use the normalized graph Laplacian matrix  $\mathbf{L}$  as positional encoding, which is computed by  $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U}$ , where  $\mathbf{I}$  is the identity matrix,  $\mathbf{D}$  the  $N \times N$  degree matrix,  $\mathbf{A}$  the adjacency matrix, and  $\mathbf{U}$  and  $\mathbf{\Lambda}$  are the matrices of eigenvectors and eigenvalues, respectively. The positional encodings are the first 32 eigenvectors with largest eigenvalues. Positional encodings are added to the nodes features after tokenization.

Table 1: Overview of data augmentations for spatially-embedded graphs such as neuronal skeletons.

Level	Augmentations
Graph	(1) Subsampling, (2) Rotation, (5) Translation
	
Subgraph	(3) Jittering, (4) Branch deletion
	

**Rotation:** we perform random 3D rotation around the y-axis, that is orthogonal to the pia. **(3) Jittering:** we randomly translate individual node positions by adding Gaussian noise with  $\mathcal{N}(0, \sigma_1)$ . **(4) Subgraph deletion:** We identify branches that connect leaf nodes to the last upstream parent node in the graph, i.e. terminal branches that do not split into further branches, and randomly delete  $n$  of them starting at a random location along the branch, while maintaining the overall graph structure. **(5) Graph position:** we randomly translate the graph as a whole by adding Gaussian noise with  $\mathcal{N}(0, \sigma_2)$  to all nodes. Unlike Caron et al. (2021), we do not differentiate between the augmentations seen by the student and the teacher network.

**Data augmentation.** Data augmentation plays an important role in self-supervised learning and needs to be adapted to the data, since it expresses which invariances should be imposed. Given the spatial neuronal data, we apply the following augmentations: **(1) Sub-sampling:** We subsample the original graph to a fixed number of  $n$  nodes by randomly removing nodes that are not branching points (i.e. nodes connected to more than two other nodes), and connecting the two neighbors of the removed node. This facilitates batch processing. Furthermore, this augmentation retains the global structure of the neuron, while altering local structure in the two views. **(2)**

## 4 Data and Experiments

### 4.1 Synthetic graphs

To demonstrate that our novel attention mechanism is strictly more powerful than simple all-to-all attention on a graph, we generate a synthetic graph dataset. In this dataset, the five classes share similar node locations but differ in how the nodes are connected. See Appendix A for the detailed generation process. We use this dataset to test the efficacy of our novel attention mechanism, AC-ATTENTION, and the positional encoding.

### 4.2 Neuronal and tree graphs

We apply GRAPHDINO to five publicly available neuronal datasets and one non-neuronal dataset.

**Blue Brain Project (BBP): Rat somatosensory cortex.** Available from the Neocortical Microcircuit Collaboration Portal of the Blue Brain Project<sup>1</sup> (Ramaswamy et al., 2015), the dataset contains 1,389 neurons from juvenile rat somatosensory cortex. We train GRAPHDINO without supervision on the 3D dendritic morphologies of all neurons. For evaluation, we use the subset of 616 neurons which have been labeled by experts into cell types and cortical layer. Of these 616 neurons 286 are excitatory that have been assigned to 14 cell types (Markram et al., 2015). See Appendix C.5.1 for more details on the dataset. We use this dataset to evaluate the capability of GRAPHDINO to learn useful representations of neuronal morphologies that align with known cell types, perform ablation experiments on the novel graph augmentation strategies and compare to previous work using manually-defined features.

**M1 PatchSeq: Mouse motor cortex.** The M1 PatchSeq dataset contains 275 excitatory and 371 inhibitory cells from M1 in adult mouse primary motor cortex (Scala et al., 2021).<sup>2</sup> The excitatory cells (M1 EXC) have been classified into tufted, untufted and other neurons based on their morphology in a previous study (Laternus & Berens, 2021). We use this dataset to compare to previous work that learns morphological embeddings in a data-driven way. We train GRAPHDINO without supervision on the 3D dendritic morphologies of the 646 neurons. For evaluation, we follow the evaluation protocol and use the same dataset split as Laternus & Berens (2021). We additionally report the 5-nearest neighbour accuracy of three additional dataset splits to estimate the variance due to the chosen split, since the test set is very small (60 neurons) and the balanced accuracy is strongly influenced by the morphologically heterogeneous “other” class that is only represented by six samples in the test set (Laternus & Berens, 2021).

**Allen Brain Atlas (ACT): Mouse visual cortex.** As part of the Allen Cell Types Database, the dataset contains 510 neurons from the mouse visual cortex with a broad coverage of types, layers and transgenic lines.<sup>3</sup> See Allen Institute (2016) for details on how the dataset was recorded. It comes with a classification of each neuron into spiny, aspiny, or sparsely spiny, where spiny are assumed to be excitatory neurons and all else are inhibitory (Gouwens et al., 2019). Additionally, the cortical layer of each neuron is provided.

**Brain Image Library (BIL): Whole mouse brain.** The Brain Image Library contains 1,741 reconstructed neurons from cortex, claustrum, thalamus, striatum and other brain areas in mice (Peng et al., 2021).<sup>4</sup>

**Janelia MouseLight (JML): Whole mouse brain.** The Janelia MouseLight platform contains 1,200 projection neurons from the motor cortex, thalamus, subiculum, and hypothalamus (Winnubst et al., 2019).<sup>5</sup>

**Joint training on ACT, BIL and JML.** Following Chen et al. (2022b), for joint training of the ACT, BIL and JML datasets, we rotate the neurons such that the first principal component is aligned with the

<sup>1</sup><http://microcircuits.epfl.ch/#/main>

<sup>2</sup><https://download.brainimagelibrary.org/3a/88/3a88a7687ab66069/>

<sup>3</sup><http://celltypes.brain-map.org/>

<sup>4</sup><https://download.brainimagelibrary.org/biccn/zeng/luo/fMOST/>

<sup>5</sup><http://mouselight.janelia.org/>

y-axis. Chen et al. (2022b) group the neurons of the three datasets ACT, BIL and JML into eleven classes based on the cortical layer or brain region they originate from. They then evaluate their learned embeddings on a subset of six (for BIL) or four classes (for ACT and JML) that have a broad coverage across the datasets. See Appendix C.5.4 for further details.

**Botanical Trees.** The Trees dataset (Seidel et al., 2021) is a highly diverse dataset comprised of 391 skeletons of trees stemming from 39 different genres and 152 species or breedings. The skeletons were extracted from LIDAR scans of the trees. Nodes of the skeletons have a 3D coordinate associated with them. We normalize the data such that the lowest point (start of the tree trunk) is normalized to  $(0, 0, 0)$ .

#### 4.2.1 Data Preprocessing

Since the objective of GRAPHDINO is to learn purely from the 3D dendritic morphology of neurons, we normalize each graph such that the soma location is centered at  $(0, 0, 0)$  (no cortical depth information is given to the model). Furthermore, axons are removed for all experiments in the paper, because the reconstruction of axonal arbors of excitatory neurons from light microscopy images is difficult due to their small thickness and long ranges that they cover (Kanari et al., 2019) and thus often unreliable. The input nodes  $V$  have features  $v_i = [x, y, z]$  where  $v_i \in \mathbb{R}^3$  are the spatial xyz-coordinates in micrometers [ $\mu\text{m}$ ].

#### 4.2.2 Training details.

GRAPHDINO is implemented in PyTorch (Paszke et al., 2019) and trained with the Adam optimizer (Kingma & Ba, 2015). The latent dimensionality of  $z$  is 16 for the synthetic graphs and 32 for the neuronal and the botanical tree datasets. For M1 PatchSeq we use a latent dimensionality of 64. See C.2 for an overview of the hyperparameters used for training on the different datasets. At inference time, the latent embeddings  $z$  are extracted from the student network for the unaugmented graphs. We use scipy for fitting Gaussian Mixture models (GMM) and k-nearest neighbor classifiers (kNN) (Pedregosa et al., 2011).

## 5 Results

We first establish that GraphDINO works on the synthetic graph dataset and show that our novel AC-ATTENTION is necessary for exploiting information from graph connectivity. Second, we show that our novel augmentation strategies are suitable for spatially-embedded graphs that are tree-structured and that classical GNN message-passing is not sufficient when graphs have long-ranging branches. Then, we move to the gradually more complex, biological questions of spiny-aspiny differentiation, cell type recovery and consistency with existing labels. To this end, we employ in total five neuronal datasets that encompass two species and range across multiple brain areas. Finally, we compare our model to several previous works based on manually-defined morphological features as well as approaches with learned features. See Appendix B for the application of GRAPHDINO to a non-neuronal dataset.

### 5.1 AC-Attention recovers information encoded by graph connectivity

We start by demonstrating the efficacy of our novel AC-ATTENTION module. For this experiment, we use the synthetic graph dataset where classes differ in how nodes are connected whereas the distribution of node positions does not vary across classes. Therefore, considering the graph structure is necessary to differentiate between the classes (more details in Appendix A). We train GRAPHDINO on the synthetic graph dataset without labels in three configurations: (1) with AC-ATTENTION, (2) with regular transformer attention and (3) with transformer attention and without positional encoding. We assess the quality of the learned

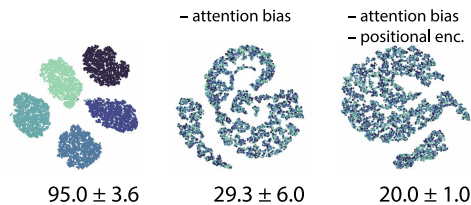


Figure 2: t-SNE embedding of latent representations of 3D synthetic graphs shown for one example run per model. Accuracy averaged over five random seeds and given as mean  $\pm$  standard deviation. “-” means removing one component from the full model.

embedding using the ground truth labels. A linear classifier on the learned embeddings achieves a test set accuracy of (1)  $95\% \pm 4$ , (2)  $29\% \pm 6$ , and (3)  $20\% \pm 1$ , showing that AC-ATTENTION allows us to capture the structure of spatially embedded graphs when the location of the nodes alone does not provide sufficient information. Removing both AC-ATTENTION and the positional encoding results in the classifier performing at chance level. Using only the positional encoding performs slightly better than chance, because the positional encodings contain some information about node connections through the graph Laplacian. To make full use of the information given by the connectivity of the graphs, using AC-ATTENTION is essential (Fig. 2).

## 5.2 Tailored graph augmentations are well-suited for spatially-embedded graphs

In self-supervised learning, data augmentation is used to obtain two views that define a positive input pair. The augmentations here are chosen to encode invariances that should not change the underlying sample identity. In previous contrastive learning for graphs, these augmentations were for example dropping random edges or masking node features (You et al., 2020). These augmentations are not appropriate for our spatially-embedded graphs that form a tree and whose only node features are their 3D location in space. Thus we designed five novel augmentation techniques specifically for spatially embedded graphs such as neural morphologies or botanical trees: subsampling, rotation, node jittering, branch deletion and graph translation (see Section 3).

To test the importance of the individual graph augmentations we perform a set of ablation experiments using the BBP dataset. We remove one augmentation from our model at a time and evaluate the leave-one-out 5-nearest neighbor accuracy when predicting the expert labels. For the subsampling augmentation we vary the number of retained nodes. Our full model achieves an average accuracy of 65.8% when classifying the excitatory cells into the 12 expert labels (Appendix C.5.1). When removing individual data augmentations the accuracy decreases (Tab. 2). Especially 3D rotation and graph translation are important augmentation strategies whose removal lead to substantial performance deterioration.

## 5.3 Message-passing is not sufficient for long-range graphs

Next, we investigate whether classical message-passing is sufficient to process graphs with long-ranging branches such as neuronal morphologies. Therefore, we train GRAPHDINO once when only using message passing while removing the global attention (setting  $\lambda = 0$  in Eq. 1). This decreases the performance to 59.8% (Tab. 2). Additionally, we train INFOGRAPH (Sun et al., 2020), as a baseline for an unsupervised method that learns graph-level representations and uses GNN message-passing. INFOGRAPH achieves accuracy of 48.2% (Tab. 3). Thus, we conclude that using global attention is beneficial in situations where graphs contain long-range branches. Global attention enables information flow between distant (in terms of graph connectivity) nodes that might be close in space or function.

## 5.4 Morphological embeddings differentiate between spiny/aspiny cells and layers

To evaluate the capability of GRAPHDINO to capture essential features of 3D neuronal shapes purely data-driven, we train GRAPHDINO on the BBP dataset and use t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) to map the learned embeddings of the BBP dataset into 2D (Fig. 3) for

Table 2: Ablation results on the BBP dataset. Cell-type classification accuracy [%] of our model and ablations averaged over three random seeds and given as mean  $\pm$  standard deviation. “-” indicates removal of an augmentation or model component.

Model	Accuracy
<b>GraphDINO</b>	<b>65.8 <math>\pm</math> 1</b>
- 3D rot.	55.4 $\pm$ 1
- node jitter	64.8 $\pm$ 2
- graph translation	55.6 $\pm$ 2
- drop branch	64.6 $\pm$ 1
subsampling: 50 nodes	60.0 $\pm$ 0
subsampling: 200 nodes	62.0 $\pm$ 3
- adjacency ( $\gamma = 0$ )	62.8 $\pm$ 1
- attention ( $\lambda = 0$ )	59.8 $\pm$ 2

Table 3: Cell-type classification accuracy [%] on the BBP dataset. Performance of our model and INFOGRAPH (Sun et al., 2020) averaged over three random seeds and given as mean  $\pm$  standard deviation.

Model	Accuracy
INFOGRAPH (Sun et al., 2020)	48.2 $\pm$ 0
<b>GraphDINO</b>	<b>65.8 <math>\pm</math> 1</b>

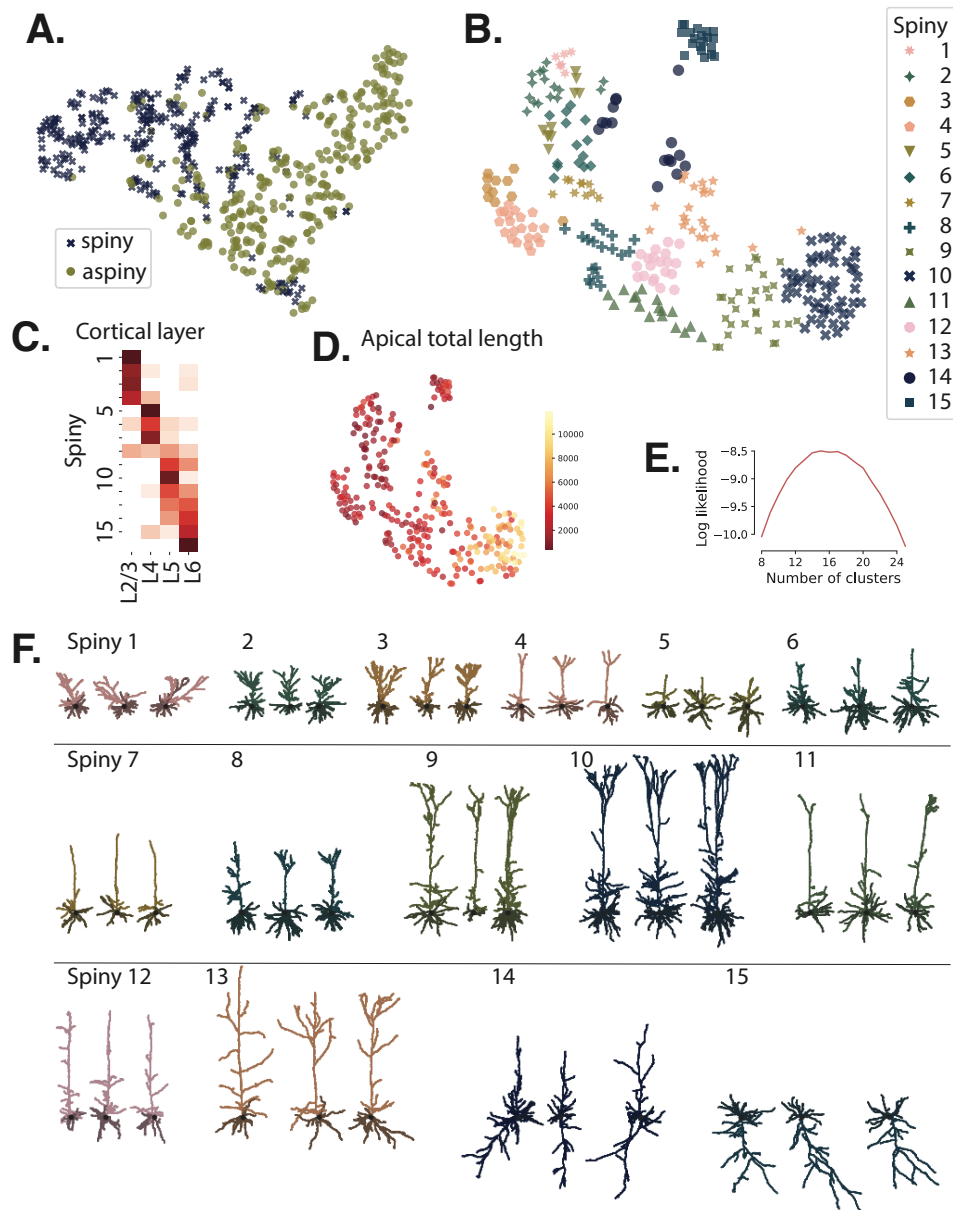


Figure 3: **A.** t-SNE embedding (perplexity=30) of latent representation of 3D neuronal morphologies of the BBP dataset showing a separation into spiny and aspiny neurons ( $n = 616$ ). **B.** t-SNE embedding (perplexity=30) of the latent representations of the morphologies of the spiny neurons colored by the cluster found by our model ( $n = 286$ ). **C.** Relative cortical layer distribution of neurons per cluster across L2/3—L6. Higher values are indicated by red. **D.** As **B.** but neurons colored by their total apical length revealing an organization of the latent space in terms of morphological properties. **E.** Log-likelihood of Gaussian Mixture Model on held-out test set for spiny neurons used to select the optimal number of clusters. **F.** Example neurons for each cluster are shown with apical dendrites in lighter color, while basal dendrites are colored darker. Soma is indicated by black circle.

visualization. A clear separation between spiny and aspiny neurons can be observed (see Fig. 3A), indicating that our learned representation captures meaningful biological differences of the neuronal morphologies.

Interestingly, some of the spiny neurons end up in the aspiny cluster (Fig. 3A bottom right). These are inverted L6 neurons (Fig. 3F Cluster 15), whose size and dendritic tree are morphologically similar to the surrounding aspiny neurons that also show a downwards bias in the dendritic tree.

### 5.5 Morphological embeddings recover known excitatory cell types

To identify cell types, we fit a Gaussian mixture model (GMM) with a diagonal covariance matrix to our learned representation of the spiny neurons. To determine the number of clusters, we fit 1,000 GMMs with different random seeds using five-fold cross-validation for 2–30 clusters. We average over the log-likelihood for each number of clusters over repetitions and folds. We find  $n = 15$  to be the optimal number of clusters (Fig. 3E).

Having identified the optimal number of clusters, we re-fit the GMM to the full dataset including all spiny neurons. To avoid picking a particularly good or bad random clustering, we fit 100 models and choose the one that has the highest average adjusted rand index (ARI) to all other clusterings.

The spiny neurons cluster nicely into different shapes and layers (Fig. 3F and Appendix Fig. D.6), retrieving known excitatory cell types. The first four spiny clusters contain mainly cells from layer 2/3 (L2/3) (Fig. 3C) and group them by morphology: Cluster 1 contains wide and short neurons from layer 2/3, while L2/3 neurons in cluster 4 are more elongated with a less pronounced apical tuft (Fig. 3F). Clusters 5–7 group cells from layer 4 (L4) (Fig. 3C), differentiating between spiny stellate cells (cluster 5) and atufted L4 neurons (cluster 7) (Fig. 3F). Within layer 5 and 6, neurons are grouped by their size, amount of apical tuft and obliques, as well as the direction of the apical-like dendrites: For instance, cluster 10 groups thick-tufted pyramidal cells from layer 5 and cluster 15 contains inverted L6 neurons (Fig. 3F).

Most clusters show a strong preference for grouping cells whose soma position is in a certain layer (Fig. 3C) even though the model — in contrast to the experts who labeled the cells — does not have access to anatomical knowledge such as cortical layer of origin. One exception are pyramidal L6 cells with upward-directed apicals that separate less well and get rather clustered with L4 and L5 neurons of the same size and similar morphological shape. This is to be expected, as the model only learns to differentiate between different morphologies but has no knowledge about anatomical features such as soma depth.

### 5.6 Data-driven clusters are consistent with expert labels

To compare our data-driven features to manually-designed features, we compute the adjusted rand index (ARI) between our clusters and the expert-identified cell types on the BBP dataset and compare the performance to the clusters based on morphometrics obtained by Gouwens et al. (2020). We achieve an ARI performance of 0.31 when clustering neurons across all cortical layers together while using significantly less prior information than Gouwens et al. (2019). In comparison, Gouwens et al. (2019) reached an ARI of 0.27 with a feature space specifically designed for spiny neurons and by splitting the neurons into their cortical layer of origin before performing the clustering. This approach reduces the complexity of the problem significantly, since misassignments across layers are excluded by construction. When performing the clustering like Gouwens et al. (2019) only within the layers, we achieve an ARI of 0.46 (Tab. 4).

### 5.7 Morphological embeddings encode distinct morphological features

Laternus & Berens (2021) classified the M1 EXC dataset (Scala et al., 2021) into three classes based on presence of an apical tuft (tufted, untufted and others). Following their work, we train a 5-nearest-neighbor classifier on our learned embeddings and show that GRAPHDINO learns meaningful features to differentiate between the three classes (Tab. 5). Our method outperforms their MORPHVAE method as well as a baseline using density maps of the neurons (Laternus & Berens, 2021). This dataset is rather small and Laternus & Berens (2021) used only a single train/test split. To estimate how reliable the reported accuracy metrics are, we compute the cross-validated accuracy across multiple different train/test splits, which show a vari-



Table 4: Adjusted rand index (ARI) between identified clusters and expert labels for the learned embeddings from GRAPHDINO and manually-defined features by Gouwens et al. (2019) and expert labels, when performing the clustering within cortical layers and across the whole cortex.

Clustering	Features	ARI
across layers	GRAPHDINO	0.31
within layers	Gouwens et al. (2019)	0.27
	<b>GraphDINO</b>	<b>0.46</b>

Table 5: Balanced accuracy [%] on M1 EXC test set using the learned embeddings from either GraphDINO or MorphVAE (Laturnus & Berens, 2021) (mean  $\pm$  SEM across three runs and across three data splits, respectively). Percentages in brackets indicate the amount of labels used during training for MorphVAE. GraphDINO is trained without labels.

Model	Accuracy over runs (mean $\pm$ SEM)	Accuracy over splits (mean $\pm$ SD)
MorphVAE (100 %)	70 $\pm$ 5	-
MorphVAE (0 %)	58 $\pm$ 7	-
Density Map (0 %)	60	-
<b>GraphDINO (0 %)</b>	<b>68 <math>\pm</math> 5</b>	71 $\pm$ 9

ability across splits of  $\pm 9\%$  (standard deviation; Tab. 5). We conclude that GraphDINO likely outperforms MorphVAE trained without supervision and performs approximately on par with MorphVAE trained fully supervised.

## 5.8 Morphological embeddings encode cortical regions

TREEMOCO (Chen et al., 2022b) is an LSTM-based model that was concurrently proposed to perform unsupervised representation learning on neuronal graphs. The model uses as input the simplified skeletons of neurons that only contain the branching points as nodes. They compute 26 manually-selected features in addition to the xyz-coordinates as node features to describe the morphology of the skeletons between branching points. TREEMOCO is trained on a combination of the datasets BIL, JML and ACT and quantitatively evaluated on the task of predicting the brain anatomical region or cortical layer of origin of the neurons on a subset of the neuronal classes. Chen et al. (2022b) remove 955 neurons from the dataset due to “reconstruction errors” and evaluate on a 80-20% training-test split. Since we did not have access to the exact neurons used for training and evaluation both in terms of split and which neurons were removed, we trained unsupervised on the joint dataset and evaluated using 5-fold cross-validation, i.e. splitting the data into five folds and evaluating each fold, given the other four folds as training data and reporting the average performance across folds. For further details regarding the evaluation, see Appendix C.5.4.

GRAPHDINO performs on par or better than TREEMOCO and GRAPHCL when predicting the origin of neurons (Tab. 6). Note that GRAPHDINO is fully data-driven while TREEMOCO and GRAPHCL additionally employ manually extracted node features.

Note that the evaluation reported by Chen et al. (2022b) uses excitatory and inhibitory neurons at the same time. With this approach, morphologies of neurons of the “same” class can look very different (Fig. C.5). A better proxy task to evaluate the encoding capabilities of the models would be to restrict the evaluation to only excitatory cells. For the ACT dataset this information is available. We therefore repeated the evaluation only on this subset (Tab. 6), which should provide a more meaningful baseline for future studies.

Table 6: Cell-type classification on the TreeMoCo dataset. Performance of our model (GRAPHDINO) averaged over three random seeds and given as mean  $\pm$  standard deviation. TREEMOCO and GRAPHCL performance given as the average accuracy over the last five epochs per dataset. \*Results taken Fig. C1 of Chen et al. (2022b).

Model	BIL-6	JML-4	ACT	ACT spiny
TreeMoCo*	76.9	59.7	53.9	-
GraphCL*	66.3	50.6	55.6	-
GRAPHDINO	79 $\pm$ 1	63 $\pm$ 6	54 $\pm$ 5	73 $\pm$ 6

## 6 Limitations

GRAPHDINO is designed to learn graph-level representations of spatially-embedded tree-structured graphs using self-supervised learning. As we focus on graphs where each node has a location in 3D space and design the data augmentations accordingly, the approach is not expected to work out-of-the-box on graphs that have different node features. AC-ATTENTION is likely to be beneficial in many other scenarios as well, since it can smoothly interpolate between message passing and global attention based on node similarity, but this hypothesis remains to be tested empirically. Data augmentations would need to be adapted to the respective data domain and the respective invariances that should be encoded or supervised learning to be used. The attention mechanism is not tied in any way to the self-supervised learning objective we use.

We encode the desired invariance for neuronal morphologies in GRAPHDINO via tailored data augmentations. Rotation and translation equivariance could alternatively be built into the architecture of the encoders explicitly. Recent works have proposed such architectures for GNNs (Satorras et al., 2021), as well as for transformers (Fuchs et al., 2020). Adapting these for AC-ATTENTION would be an interesting future research direction.

Computing the full transformer attention matrix has a quadratic complexity and might therefore be computationally infeasible for graphs with a large number of nodes. We solve this problem here by subsampling the neuronal skeletons to a smaller number of nodes, which has the added benefit of being a strong data augmentation that keeps the global morphology of the neuron intact while altering the local structure between the two views. However, this approach might not be suitable for all graph datasets. There has been some work in building attention mechanism that scale linearly with the number of input tokens (Wang et al., 2020; Kitaev et al., 2020; Choromanski et al., 2021), but integrating them with the message passing might not be straightforward.

Self-supervised learning has been shown to be most successful when training on large datasets (Bao et al., 2022; Oquab et al., 2023). We equipped GRAPHDINO with appropriate inductive biases to make it possible to learn on the smaller publicly available neuronal datasets that have been used in previous studies. Nevertheless, applying GRAPHDINO to neuronal datasets with more samples will likely improve its learning capabilities. With the continual development of better imaging techniques and initiatives like MICRONS (MICrONS Consortium et al., 2023) more large-scale datasets of neuronal morphologies will be available to test this hypothesis.

In terms of neuronal cell type classification, we did not take some features into account that have been previously used to differentiate cell types, such as the shape of the soma (as formerly used for GABAergic interneurons) or spine densities (Ascoli et al., 2008). Future work could focus on incorporating them into our framework. Depending on the type of feature, they could be easily integrated by adding them as features of the graph or as additional node features.

## 7 Conclusion

Increasingly large and complex datasets of neurons have given rise to the need for unbiased and quantitative approaches to cell type classification. We have demonstrated one such approach that is purely data-driven and self-supervised, and that learns a low-dimensional representation of the 3D shape of a neuron. By using self-supervised learning, we do not pre-specify which cell types to learn and which features to use, thereby reducing bias in the classification process and opening up the possibility to discover new cell types. A similar approach can also be useful in other domains beyond neuroscience, where samples of the dataset are spatial graphs and graph-level embeddings are desired, such as tree classification in forestry.

## Acknowledgments

We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS), Tübingen, for supporting Marissa A. Weis. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon Europe research and innovation program (Grant agreement No. 101041669).

## References

- Allen Institute. Allen cell types database technical white paper: Cell morphology and histology. 2016. URL [http://help.brain-map.org/download/attachments/8323525/CellTypes\\_Morph\\_Overview.pdf](http://help.brain-map.org/download/attachments/8323525/CellTypes_Morph_Overview.pdf).
- Rubén Armañanzas and Giorgio A. Ascoli. Towards the automatic classification of neurons. *Trends in Neurosciences*, 38(5):307–318, 2015.
- Giorgio Ascoli, Lidia Alonso-Nanclares, Stewart Anderson, Germán Barrionuevo, Ruth Benavides-Piccione, Andreas Burkhalter, Gyorgy Buzsáki, Bruno Cauli, Javier Defelipe, and Alfonso Fairen. Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nature reviews. Neuroscience*, 9:557–568, 2008.
- Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022.
- Cathryn Cadwell, Athanasia Palasantza, Xiaolong Jiang, Philipp Berens, Qiaolin Deng, Marlene Yilmaz, Jacob Reimer, Shan Shen, Matthias Bethge, Kimberley Tolia, Rickard Sandberg, and Andreas Tolia. Electrophysiological, transcriptomic and morphologic profiling of single neurons using patch-seq. *Nature Biotechnology*, 34, 2015.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.
- Dexiong Chen, Leslie O’Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning, 2022a.
- Hanbo Chen, Jiawei Yang, Daniel Maxim Iascone, Lijuan Liu, Lei He, Hanchuan Peng, and Jianhua Yao. Treemoco: Contrastive neuron morphology representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.
- Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- Krzysztof Marcin Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2021.
- Javier Defelipe, Pedro López-Cruz, Ruth Benavides-Piccione, Concha Bielza, Pedro Larranaga, Stewart Anderson, Andreas Burkhalter, Bruno Cauli, Alfonso Fairen, Dirk Feldmeyer, Gord Fishell, David Fitzpatrick, Tamás Freund, Guillermo Gonzalez Burgos, Shaul Hestrin, Sean Hill, Patrick Hof, Josh Huang, Edward Jones, and Giorgio Ascoli. New insights into the classification and nomenclature of cortical gabaergic interneurons. *Nature reviews. Neuroscience*, 14, 2013.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv.org*, 2020.
- David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

- Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv.org*, 2012.09699, 2021.
- Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv.org*, 2003.00982, 2020.
- Leila Elabbady, Sharmishta Seshamani, Shang Mu, Gayathri Mahalingam, Casey M Schneider-Mizell, Agnes Bodor, J Alexander Bae, Derrick Brittain, JoAnn Buchanan, Daniel J Bumbarger, et al. Quantitative census of local somatic features in mouse visual cortex. *bioRxiv*, 2022.
- Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1970–1981, 2020.
- Rohan Gala, Agata Budzillo, Fahimeh Baftizadeh, Jeremy Miller, Nathan Gouwens, Anton Arkhipov, Gabe Murphy, Bosiljka Tasic, Hongkui Zeng, Michael Hawrylycz, et al. Consistent cross-modal identification of cortical neurons with coupled autoencoders. *Nature Computational Science*, 1(2):120–127, 2021.
- Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proc. of the International Conf. on Machine learning (ICML)*, pp. 1263–1272, 2017.
- Nathan Gouwens, Staci Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan Sunkin, David Feng, Costas Anastassiou, Eliza Barkan, Kris Bickley, Nicole Blesie, Thomas Braun, Krissy Brouner, Agata Budzillo, Shiella Caldejon, Tamara Casper, Dan Castelli, Peter Chong, and Christof Koch. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature Neuroscience*, 22, 2019.
- Nathan W. Gouwens, Staci A. Sorensen, Fahimeh Baftizadeh, Agata Budzillo, Brian R. Lee, Tim Jarsky, Lauren Alfiler, Katherine Baker, Eliza Barkan, Kyla Berry, Darren Bertagnolli, Kris Bickley, Jasmine Bomben, Thomas Braun, Krissy Brouner, Tamara Casper, Kirsten Crichton, Tanya L. Daigle, Rachel Dalley, Rebecca A. de Frates, Nick Dee, Tsega Desta, Samuel Dingman Lee, Nadezhda Dotson, Tom Egdorf, Lauren Ellingwood, Rachel Enstrom, Luke Esposito, Colin Farrell, David Feng, Olivia Fong, Rohan Gala, Clare Gamlin, Amanda Gary, Alexandra Glandon, Jeff Goldy, Melissa Gorham, Lucas Graybuck, Hong Gu, Kristen Hadley, Michael J. Hawrylycz, Alex M. Henry, DiJon Hill, Madie Hupp, Sara Kebede, Tae Kyung Kim, Lisa Kim, Matthew Kroll, Changkyu Lee, Katherine E. Link, Matthew Mallory, Rusty Mann, Michelle Maxwell, Medea McGraw, Delissa McMillen, Alice Mukora, Lindsay Ng, Lydia Ng, Kiet Ngo, Philip R. Nicovich, Aaron Oldre, Daniel Park, Hanchuan Peng, Osnat Penn, Thanh Pham, Alice Pom, Zoran Popović, Lydia Potekhina, Ramkumar Rajanbabu, Shea Ransford, David Reid, Christine Rimorin, Miranda Robertson, Kara Ronellenfitch, Augustin Ruiz, David Sandman, Kimberly Smith, Josef Sulc, Susan M. Sunkin, Aaron Szafer, Michael Tieu, Amy Torkelson, Jessica Trinh, Herman Tung, Wayne Wakeman, Katelyn Ward, Grace Williams, Zhi Zhou, Jonathan T. Ting, Anton Arkhipov, Uygur Sümbül, Ed S. Lein, Christof Koch, Zizhen Yao, Bosiljka Tasic, Jim Berg, Gabe J. Murphy, and Hongkui Zeng. Integrated morphoelectric and transcriptomic classification of cortical gabaergic cells. *Cell*, 183(4):935–953.e19, 2020.
- William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1025–1035, 2017.
- Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *Proc. of the International Conf. on Machine learning (ICML)*, volume 119, pp. 4116–4126, 2020.
- Lida Kanari, Pawel Dlotko, Martina Scolamiero, Ran Levi, Julian C. Shillcock, Kathryn Hess, and Henry Markram. A topological representation of branching neuronal morphologies. *Neuroinformatics*, 16:3 – 13, 2017.
- Lida Kanari, Srikanth Ramaswamy, Ying Shi, Sebastien Morand, Julie Meystre, Rodrigo Perin, Marwan Abdellah, Yun Wang, Kathryn Hess, and Henry Markram. Objective morphological classification of neocortical pyramidal cells. *Cerebral Cortex*, 29(4):1719–1735, 2019.

- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015.
- Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Johannes Klicpera, Stefan Weiß enberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.
- Sophie C. Latusmus and Philipp Berens. Morphvae: Generating neural morphologies from 3d-walks using a variational autoencoder with spherical latent space. In *Proc. of the International Conf. on Machine learning (ICML)*, volume 139, pp. 6021–6031, 2021.
- Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *Proc. of the International Conf. on Learning Representations (ICLR)*, 2016.
- Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *arXiv.org*, 1608.03983, 2016.
- Yanbin Lu, Lawrence Carin, Ronald Coifman, William Shain, and Badrinath Roysam. Quantitative arbor analytics: unsupervised harmonic co-clustering of populations of brain cell arbors based on l-measure. *Neuroinformatics*, 13(1):47–63, 2015.
- Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael Reimann, Marwan Abdellah, Carlos Aguado, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, Atenekeng Kahou Guy Antoine, Thomas K Berger, Ahmet Bilgili, Nenad Buncic, Athanassia Chalimourda, Giuseppe Chindemi, Jean-Denis Courcol, Fabien Delalondre, Vincent Delattre, and Felix Schürmann. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163:456–492, 2015.
- Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers, 2021.
- The MICrONS Consortium, J. Alexander Bae, Mahaly Baptiste, Caitlyn A. Bishop, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Brendan Celii, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Paul G. Fahey, Tim Fliss, Emmanouil Froudarakis, Jay Gager, Clare Gamlin, William Gray-Roncal, Akhilesh Halageri, James Hebditch, Zhen Jia, Emily Joyce, Justin Joyce, Chris Jordan, Daniel Kapner, Nico Kemnitz, Sam Kinn, Lindsey M. Kitchell, Selden Koolman, Kai Kuehner, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Jordan Matelsky, Sarah McReynolds, Elanine Miranda, Eric Mitchell, Shanka Subhra Mondal, Merlin Moore, Shang Mu, Taliah Muhammad, Barak Nehoran, Oluwaseun Ogedengbe, Christos Papadopoulos, Stelios Papadopoulos, Saamil Patel, Xaq Pitkow, Sergiy Popovych, Anthony Ramos, R. Clay Reid, Jacob Reimer, Patricia K. Rivlin, Victoria Rose, Casey M. Schneider-Mizell, H. Sebastian Seung, Ben Silverman, William Silversmith, Amy Sterling, Fabian H. Sinz, Cameron L. Smith, Shelby Suckow, Marc Takeno, Zheng H. Tan, Andreas S. Tolia, Russel Torres, Nicholas L. Turner, Edgar Y. Walker, Tianyu Wang, Adrian Wanner, Brock A. Wester, Grace Williams, Sarah Williams, Kyle Willie, Ryan Willie, William Wong, Jingpeng Wu, Chris Xu, Runzhe Yang, Dimitri Yatsenko, Fei Ye, Wenjing Yin, Rob Young, Szi chieh Yu, Daniel Xenos, and Chi Zhang. Functional connectomics spanning multiple areas of mouse visual cortex. *bioRxiv*, 2023. doi: 10.1101/2021.07.28.454025.
- Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv.org*, 1707.05005, 2017.
- Marcel Oberlaender, Christiaan P. J. de Kock, Randy M. Bruno, Alejandro Ramirez, Hanno S. Meyer, Vincent J. Derksen, Moritz Helmstaedter, and Bert Sakmann. Cell Type-Specific Three-Dimensional Structure of Thalamocortical Circuits in a Column of Rat Vibrissal Cortex. *Cerebral Cortex*, 22(10): 2375–2391, 2012.

- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv.org*, 1807.03748, 2018.
- Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.
- Hanchuan Peng, Peng Xie, Lijuan Liu, Xiuli Kuang, Yimin Wang, Lei Qu, Hui Gong, Shengdian Jiang, Anan Li, Zongcai Ruan, Liya Ding, Zizhen Yao, Chao Chen, Mengya Chen, Tanya Daigle, Rachel Dalley, Zhangcan Ding, Yanjun Duan, Aaron Feiner, and Hongkui Zeng. Morphological diversity of single neurons in molecularly defined cell types. *Nature*, 598:174–181, 10 2021. doi: 10.1038/s41586-021-03941-1.
- Sridevi Polavaram, Todd A Gillette, Ruchi Parekh, and Giorgio A Ascoli. Statistical analysis and data mining of digital reconstructions of dendritic morphologies. *Frontiers in neuroanatomy*, 8:138, 2014.
- Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proc. of Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 1150–1160, 2020.
- Srikanth Ramaswamy, Jean-Denis Courcol, Marwan Abdellah, Stanislaw R. Adaszewski, Nicolas Antille, Selim Arsever, Guy Atenekeg, Ahmet Bilgili, Yury Brukau, Athanassia Chalimourda, Giuseppe Chindemi, Fabien Delalondre, Raphael Dumusc, Stefan Eilemann, Michael Emiel Gevaert, Pdraig Gleeson, Joe W. Graham, Juan B. Hernandez, Lida Kanari, Yury Katkov, Daniel Keller, James G. King, Rajnish Ranjan, Michael W. Reimann, Christian Rössert, Ying Shi, Julian C. Shillcock, Martin Telefont, Werner Van Geit, Jafet Villafranca Diaz, Richard Walker, Yun Wang, Stefano M. Zaninetta, Javier DeFelipe, Sean L. Hill, Jeffrey Muller, Idan Segev, Felix Schürmann, Eilif B. Muller, and Henry Markram. The neocortical microcircuit collaboration portal: a resource for rat somatosensory cortex. *Frontiers in Neural Circuits*, 9: 44, 2015.
- Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer, 2022.
- Santiago Ramón y Cajal. *Histologie du système nerveux de l’homme et des vertébrés*. 1911.
- Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *arXiv.org*, 2102.09844, 2021.
- Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn Cadwell, Jesus Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Latus, Elanine Miranda, Shalaka Mulherkar, Zheng Tan, Zizhen Yao, Hongkui Zeng, Rickard Sandberg, Philipp Berens, and Andreas Tolias. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598:1–7, 2021.
- Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

- Philipp Schubert, Sven Dorkenwald, Michal Januszewski, Viren Jain, and Joergen Kornfeld. Learning cellular morphology with neural networks. *Nature Communications*, 10:2736, 2019.
- Ruggero Scorcioni, Sridevi Polavaram, and Giorgio A Ascoli. L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature protocols*, 3(5):866–876, 2008.
- Dominik Seidel, Yonten Dorji, Bernhard Schuldt, Emilie Isasa, and Klaus Körber. Dataset: New insights into tree architecture from mobile laser scanning and geometry analysis. *Dryad*, 2021. doi: <https://doi.org/10.5061/dryad.2fqz612n6>.
- Sharmishta Seshamani, Leila Elabbady, Casey Schneider-Mizell, Gayathri Mahalingam, Sven Dorkenwald, Agnes Bodor, Thomas Macrina, Daniel Bumbarger, JoAnn Buchanan, Marc Takeno, Wenjing Yin, Derrick Brittain, Russel Torres, Daniel Kapner, Kisuk Lee, Ran Lu, Jingpeng Wu, Nuno daCosta, R. Clay Reid, and Forrest Collman. Automated neuron shape analysis from electron microscopy. *arXiv.org*, 2006.00100, 2020.
- D. A. Sholl. Dendritic organization in the neurons of the visual and motor cortices of the cat. *Journal of Anatomy*, 87:387–406, 1953.
- Fan-Yun Sun, Jordan Hoffmann, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.
- Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2022.
- Harry BM Uylings and Jaap Van Pelt. Measures for quantifying dendritic arborizations. *Network: computation in neural systems*, 13(3):397, 2002.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018.
- Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019.
- Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv.org*, 2020.
- Yun Wang. A simplified morphological classification scheme for pyramidal cells in six layers of primary somatosensory cortex of juvenile rats. *IBRO Reports*, 5, 2018.
- Johan Winnubst, Erhan Bas, Tiago A. Ferreira, Zhuhao Wu, Michael N. Economo, Patrick Edson, Ben J. Arthur, Christopher Bruns, Konrad Rokicki, David Schauder, Donald J. Olbris, Sean D. Murphy, David G. Ackerman, Cameron Arshadi, Perry Baldwin, Regina Blake, Ahmad Elsayed, Mashtura Hasan, Daniel Ramirez, Bruno Dos Santos, Monet Weldon, Amina Zafar, Joshua T. Dudman, Charles R. Gerfen, Adam W. Hantman, Wyatt Korff, Scott M. Sternson, Nelson Spruston, Karel Svoboda, and Jayaram Chandrashekar. Reconstruction of 1,000 projection neurons reveals new cell types and organization of long-range connectivity in the mouse brain. *Cell*, 179(1):268–281.e13, 2019. ISSN 0092-8674. doi: <https://doi.org/10.1016/j.cell.2019.07.042>.

- Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *Proc. of the International Conf. on Machine learning (ICML)*, volume 139, pp. 11548–11558, 2021.
- Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- Jiawei Zhang, Haopeng Zhang, Congying Xia, and Li Sun. Graph-bert: Only attention is needed for learning graph representations. *arXiv.org*, 2001.05140, 2020.
- Jie Zhao, Xuejin Chen, Zhiwei Xiong, Zheng-Jun Zha, and Feng Wu. Graph representation learning for large-scale neuronal morphological analysis. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022. doi: 10.1109/TNNLS.2022.3204686.
- Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pp. 2069–2080, 2021.



## Appendices

### A Synthetic graph dataset

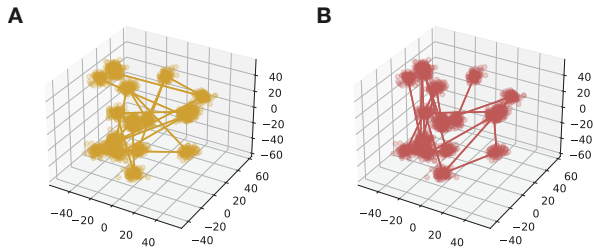


Figure A.1: Example classes 1 (A) and 2 (B) of synthetic graph dataset. Samples within one class share graph connectivity. Samples between classes share mean node locations. Node locations are drawn from  $\mathcal{N}(\mu, \sigma)$ .

node positions from  $\mathcal{N}(\mu, \sigma)$  with  $\mu$  equal to the above drawn means and  $\sigma = 10$ .

Tab. C.1, Tab. C.2 and Tab. C.3 list the hyperparameters used for experiments on the synthetic graphs.

**t-SNE of the learned latent spaces:** To visualize the learned latent space we perform t-SNE with a perplexity of 30 to reduce the embedding to two dimensions (Fig. A.2).

**Linear classifier:** We train a supervised linear classifier on the extracted embeddings of GRAPHDINO for 100 epochs and a learning rate of 0.01. To train the classifier, we use the test set that has not been used in training GRAPHDINO, and split it in 8,000 samples for training the classifier and 2,000 samples for evaluating held-out test set accuracy.

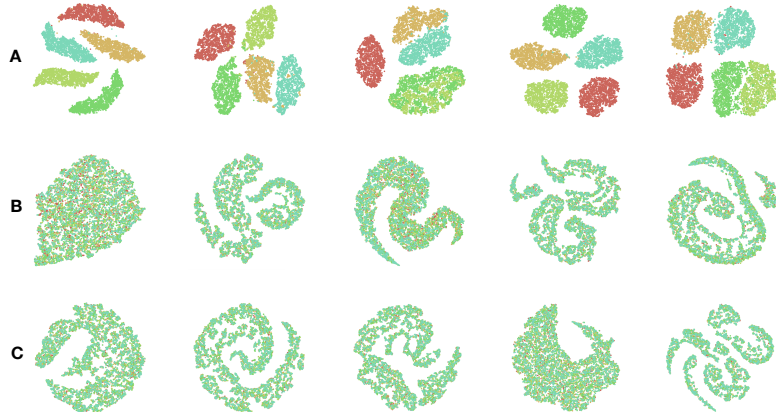


Figure A.2: t-SNE embedding of synthetic graph dataset colored by class membership for **A** five runs of GRAPHDINO with GRAPHATTENTION, **B** five runs of GRAPHDINO with regular transformer attention, and **C** five runs of GRAPHDINO with transformer attention and without positional encoding.

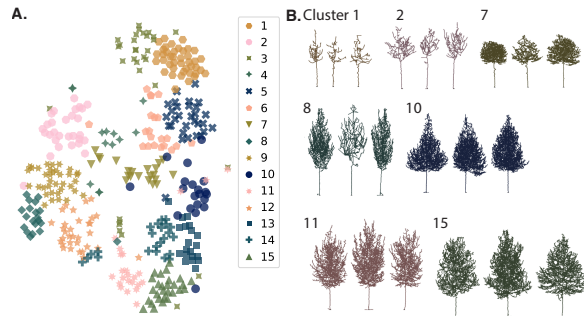


Figure B.3: **A.** t-SNE embedding of Trees dataset colored by cluster membership based on GMM clustering with 15 clusters. **B.** Three example tree morphologies are shown for different clusters.

## B Application to different domain: Tree Morphologies

We developed a model that is able to learn graph-level embeddings of spatially-embedded graphs. So far, we have shown that it yields meaningful cell types clusterings of neuronal morphologies. To show that GRAPHDINO is applicable to data domains beyond neuronal morphologies, we train our model on 3D skeletons of individual trees (from a forest).

The Trees dataset (Seidel et al., 2021) is a highly diverse dataset comprised of 391 skeletons of trees stemming from 39 different genera and 152 species or breedings. The skeletons were extracted from LIDAR scans of the trees. Nodes of the skeletons have a 3D coordinate. We normalize the data such that the lowest point (start of the tree trunk) is normalized to (0, 0, 0).

GRAPHDINO learns a latent space that orders tree morphologies with respect to their size, crown size and crown shape (Fig. B.3, Fig. D.7).

## C Extended Methods

### C.1 Background: DINO

DINO (Fig. C.4) (Caron et al., 2021) is a method for self-supervised image representation learning. Similar to previous approaches, it consists of two image encoders which process different views of an image. These views are obtained by image augmentation. The training objective is to enforce both encoders to generate the same output distribution when the same input image is shown. This can be implemented by the cross entropy loss function:  $\sum_i -q_i \log p_i$ . Both encoders are transformers that share the architecture but differ in their weights: One of the encoders is the student encoder which receives weight updates through gradients of the training objective while the other encoder’s (teacher) weights are an exponential moving average of the student’s weights. In contrast to some other self-supervised methods, DINO does not require contrastive (negative) samples. To prevent collapse, i.e. predicting the same distribution independent of the input image, two additional operations on the teacher’s predictions are crucial: sharpening by adjusting the softmax temperature, and centering using batch statistics. Besides competitive performance on downstream image classification tasks, another key finding of the paper is that object segmentations emerge in the self-attention when applying DINO training on visual transformer image encoders.

### C.2 Data preprocessing.

To speed up data loading during training, we reduce the number of nodes in the graph of each neuron to 1000 nodes in the same way as when subsampling and ensure that it contains only one connected component. If there are unconnected components, we connect them by adding an edge between two nodes of two unconnected components that have the least distance between their spatial coordinates.

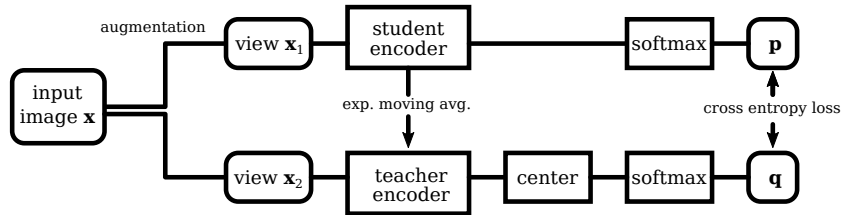


Figure C.4: The DINO method for self-supervised image representation learning (figure adapted from Caron et al. (2021)).

### C.3 Training details and hyperparameters

To select hyperparameters we run three grid searches and pick the best hyperparameters according to the lowest average loss over the BBP and M1 PatchSeq dataset.

For the optimization, we run a hyperparameter search over batch size  $\in \{32, 64, 128\}$ , learning rate  $\in \{10^{-3}, 10^{-4}, 10^{-5}\}$ , and number of training iterations  $\in \{20,000, 50,000, 100,000\}$ .

For the augmentation strength, we run a hyperparameter search over jitter variance  $\sigma_1 \in \{1.0, 0.1, 0.001\}$ , number of deleted branches  $n \in \{1, 5, 10\}$ , and graph position variance  $\sigma_2 \in \{0.1, 1.0, 10.0\}$ .

For the architecture, we run a hyperparameter search over latent dimension  $\in \{16, 32, 64\}$ , number of GRAPHATTENTION blocks (depth)  $\in \{5, 7, 10\}$ , and number of attention heads per block  $\in \{2, 4, 8\}$ .

#### C.3.1 Architecture Hyperparameters

Table C.1: Hyperparameters used for the architecture for the different datasets.  $\mathbf{D}_1$ : Dimensionality of latent embedding  $z$ .  $\mathbf{D}_2$ : Dimensionality of probability distribution  $p$ .  $\mathbf{PE}$ : Positional encoding.  $\mathbf{T temp}$ : Softmax temperature of teacher network.

Dataset	$\mathbf{D}_1$	$\mathbf{D}_2$	# layers	# heads	MLP dims	PE dims	T temp
Synthetic Graphs	16	300	4	4	16	16	0.04
BBP	32	1000	10	8	64	32	0.06
M1 PatchSeq	64	1000	7	8	64	32	0.06
Joint dataset (BIL, JML, ACT)	32	1000	7	4	64	32	0.06
Trees	32	1000	7	8	64	32	0.06

Tab. C.1 lists the hyperparameters used for the architecture for the different datasets. For the synthetic graph dataset, we downscale the network as it is a simpler dataset. DINO (Caron et al., 2021) uses an output dimensionality of 65,536 for  $p$  when training on ImageNet (Deng et al., 2009) (1,000 classes). The number of classes in the neuronal datasets is unknown, but previous literature described 14 – 19 cell types (Gouwens et al., 2019; Markram et al., 2015). Hence, we decrease the number of dimensions  $D_2$  of  $p$  proportionally to 1,000, approximately retaining the ratio between classes and number of dimensions.

#### C.3.2 Optimization Hyperparameters

The learning rate is linearly increased to the value given in Tab. C.2 during the first 2,000 iterations and then decayed using an exponential decay with rate 0.5 (Loshchilov & Hutter, 2016).

Table C.2: Hyperparameters used for optimization for the different datasets.

Dataset	Iterations	Batch size	Learning rate
Synthetic Graphs	100,000	512	$10^{-4}$
BBP	100,000	64	$10^{-4}$
M1 PatchSeq	50,000	128	$10^{-3}$
Joint dataset (BIL, JML, ACT)	100,000	128	$10^{-3}$
Trees	100,000	64	$10^{-4}$

### C.3.3 Augmentation Hyperparameters

Table C.3: Augmentation hyperparameters for the different datasets. **# nodes**: Number of nodes to subsample to.  $\sigma_1$ : Variance of node jittering. **# DB**: Number of deleted branches.  $\sigma_2$ : Variance of graph translation.

Dataset	# nodes	$\sigma_1$	# DB	$\sigma_2$
Synthetic Graphs	15	0.1	0	0
BBP	100	0.001	10	10.0
M1 PatchSeq	100	0.1	10	10.0
Joint dataset (BIL, JML, ACT)	200	1.0	5	10.0
Trees	200	0.1	5	10.0

### C.3.4 Computation

All trainings were performed on a NVIDIA Quadro RTX 5000 single GPU. Training on the neuronal BBP dataset ran for approximately 10 hours for 100,000 training iterations.

### C.4 Inference

To extract the latent representation per sample, we encode the unaugmented graphs subsampled to 200 nodes using the student encoder and extract the latent representation  $z$  using the weights of the last iteration of training (no early-stopping is used).

### C.5 Evaluation

#### C.5.1 Evaluation on BBP

For visualization of the latent space, we use t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) with PCA-initialization, Euclidean distance and a perplexity of 30.

For quantitative evaluation we use the subset of labeled excitatory neurons ( $n = 286$ ) with the following 14 expert labels: L23-PC, L4-PC, L4-SP, L4-SS, L5-STPC, L5-TTPC1, L5-TTPC2, L5-UTPC, L6-BPC, L6-IPC, L6-TPC-L1, L6-TPC-L4, L6-UTPC, L6-HPC (Markram et al., 2015).

For the ablation experiments and the comparison to INFOGRAPH Sun et al. (2020), we perform k-nearest neighbor (kNN) classification with  $k = 5$  in a leave-one-out setting predicting the above listed expert labels with two exceptions: We remove the L6-HPC cells, since there are only three samples in the dataset, and we group the L5-TTPC1 and L5-TTPC2 into one class L5-TTPC following previous work that found that they rather form a continuum than two separate classes (Gouwens et al., 2019; Kanari et al., 2019).

For the clustering analysis and the comparison to Gouwens et al. (2019), we follow Gouwens et al. (2019) and compute the adjusted rand index between our found clusters and the 14 expert labels. To determine the optimal number of clusters, we use cross-validation to compute the log-likelihood of held-out data of the Gaussian Mixture model and choose the number of clusters with the highest log-likelihood. The optimal

number of clusters is 15 for the BBP dataset. To perform clustering within cortical layers, we chose the number of clusters per layer based on the number of clusters with the majority of cells from the cortex-wide clustering (Fig. 3): four for layer 2/3, layer 5 and layer 6 and three for layer 4.

### C.5.2 Comparison to InfoGraph (Sun et al., 2020)

We use the official implementation<sup>6</sup> to train INFOGRAPH on the BBP dataset. We perform a hyperparameter search for INFOGRAPH as detailed in the original publication (Sun et al., 2020) and extend it to include more training epochs to train it for approximately the same number of iterations as GRAPHDINO. In detail, we run a grid search over learning rate ( $lr \in \{10^{-2}, 10^{-3}, 10^{-4}\}$ ), number of training epochs  $\in \{10, 20, 100, 200, 1,000, 2,000\}$  and GNN layers  $\in \{4, 8, 12\}$ . We select the hyperparameters based on the lowest unsupervised loss. The chosen hyperparameters are:  $lr = 0.001$ ,  $epochs = 1,000$  and four GNN layers with a hidden dimensionality of 32.

We evaluate the performance of INFOGRAPH (Sun et al., 2020) using a kNN classifier analogous to the ablation experiments (Appendix. C.5.1).

### C.5.3 Comparison to MorphVAE (Laternus & Berens, 2021)

We follow the evaluation protocol of Laternus & Berens (2021) and perform k-nearest neighbor (kNN) classification with  $k = 5$  on the learned latent embeddings of the excitatory neurons to predict whether they are untufted, tufted or “other” on the test set ( $n = 60$ ) and report the balanced accuracy. The “other” class only contains six examples in the test set. To get an estimate of the variance that is due to the chosen data split, we additionally evaluate three further data splits and report the average test set performance over the three splits. We report the performance of MORPHVAE as given in Tab. 3 of Laternus & Berens (2021).

### C.5.4 Comparison to TreeMoCo (Chen et al., 2022b)

A fair comparison to TreeMoCo proved difficult. We tried to replicate their setting as best as possible from the information given in the paper as well as by inferring it from their code base<sup>7</sup> while trying to set up a more fair benchmark for future works.

We downloaded the three datasets BIL, JML and ATC using the official code base of TREEMOCO and used it to assign the eleven class labels: L1, L2/3, L4, L5, L6, VPM, CP, VPL, SUB, PRE, MG and Others as used by Chen et al. (2022b). However, our cell counts slightly differ from those given in Chen et al. (2022b). More specifically, the JML dataset contained 1,200 neurons instead of 1,107.

Chen et al. (2022b) removed a substantial amount of the neurons (995 of 3,358 neurons) from the datasets due to reconstruction errors. Since we did not have access to the identities of these neurons, we trained GRAPHDINO unsupervised on all cells with more than 200 nodes ( $n_{total} = 3,138$ ;  $n_{BIL} = 1,739$ ,  $n_{JML} = 889$ ,  $n_{ACT} = 510$ ) and evaluated the proposed classes as assigned by the TREEMOCO code base. We replicated the proposed data preprocessing by centering the somata at  $(0, 0, 0)$  and aligning the neurons’ first principal component to the y-axis.

Chen et al. (2022b) performs the quantitative evaluation on a 80-20% training-test data split. Since we did not have access to the exact split, we performed five cross-validations instead and report the average accuracy over folds.

According to the paper, Chen et al. (2022b) perform k-nearest neighbor classification ( $k = 5$  or  $k = 20$  depending on the dataset). We unify the evaluation and report the kNN accuracy with  $k = 5$  for all experiments in this paper. For reference, we list the  $k = 20$  performance in Tab. C.4. In their code base, the implementation of kNN is weighted, where the neighbors vote is weighted by the cosine similarity of the embeddings. We follow the description in the paper (Chen et al., 2022b) and use the standard kNN classification without weighing the neighbors’ votes.

<sup>6</sup><https://github.com/sunfanyunn/InfoGraph>

<sup>7</sup>We additionally tried to reach out to the authors but did not get a reply.

Table C.4: Cell-type classification on the TreeMoCo dataset. Performance of our model (GRAPHDINO) averaged over three random seeds and given as mean  $\pm$  standard deviation when using  $k = 20$  for the kNN classifier.

Model	BIL-6	ACT
Ours	78 $\pm$ 2	54 $\pm$ 4

Figure C.5: Example neurons labeled as Isocortex 4 of the ACT dataset.



The performances reported by Chen et al. (2022b) are overfitted on the test set: They picked the best test set performance over epochs (for the three datasets separately) (see Fig. C1 in Chen et al. (2022b)). Additionally, they picked whether to use the latent embedding  $z$  or the projection head’s output  $p$  based on the test set performance per dataset. To give an estimate of the less overfitted performance of TREEMOCO (Chen et al., 2022b) (at least with respect to which epoch to evaluate), we report the averaged performance over the last five epochs given by Fig. C1 (Chen et al., 2022b).

Similarly, the performance of GRAPHCL (You et al., 2020) as reported by Chen et al. (2022b) is picked as the best test set performance per dataset over training epochs. We therefore report the average accuracy over the last five epochs with the given by Fig. C1 (Chen et al., 2022b).

## D Complete cluster visualizations

In the Fig. D.6 and Fig. D.7, we show the cluster assignments of all samples of the excitatory BBP dataset ( $n = 286$ ) and the Trees dataset ( $n = 391$ ), respectively.

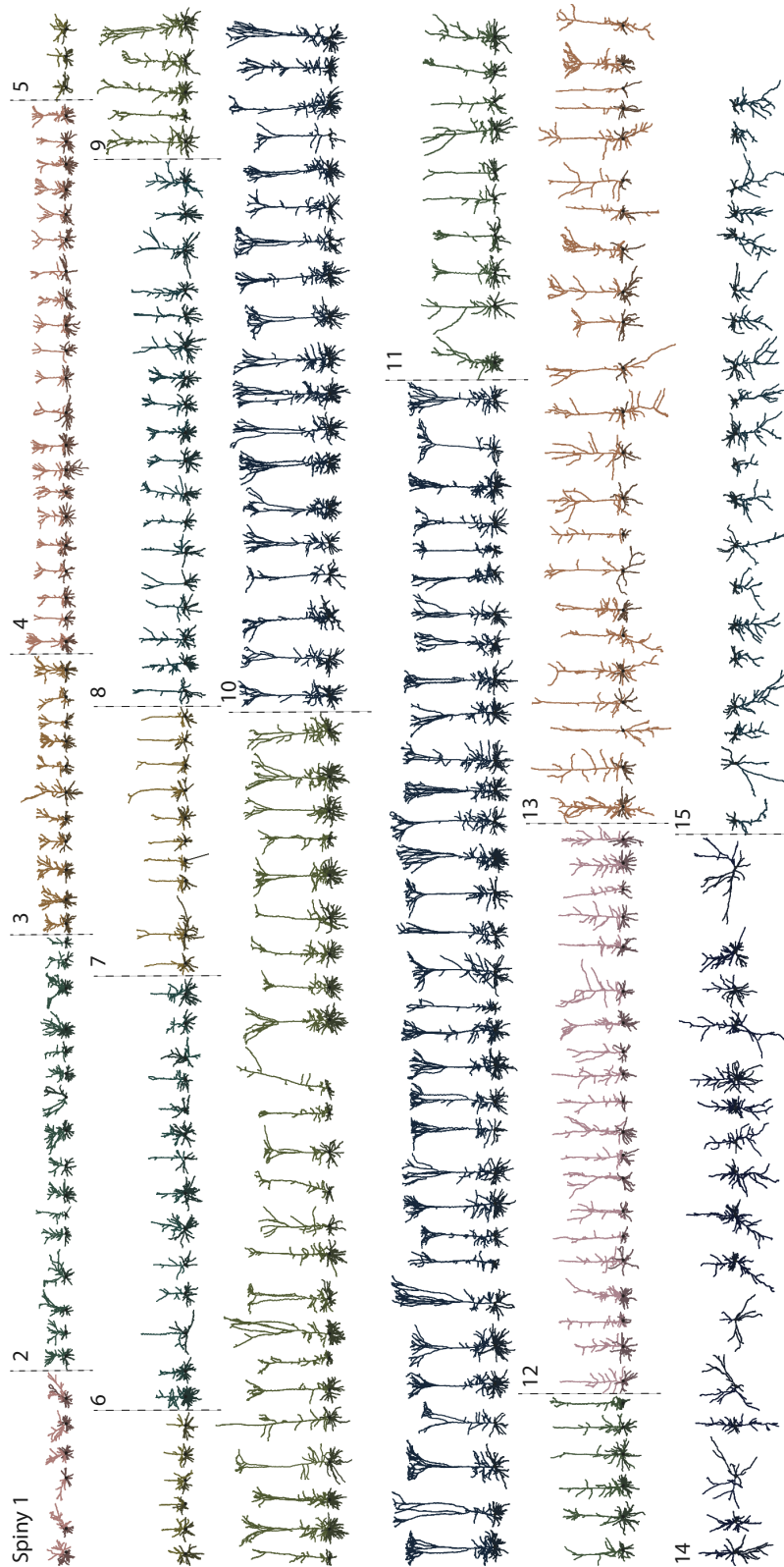


Figure D.6: Clusters of spiny neurons of BBP dataset as identified by GMM based on our learned feature space. Apical dendrites are colored lighter, while basal dendrites are shown in a darker color. Soma is marked by a black circle.

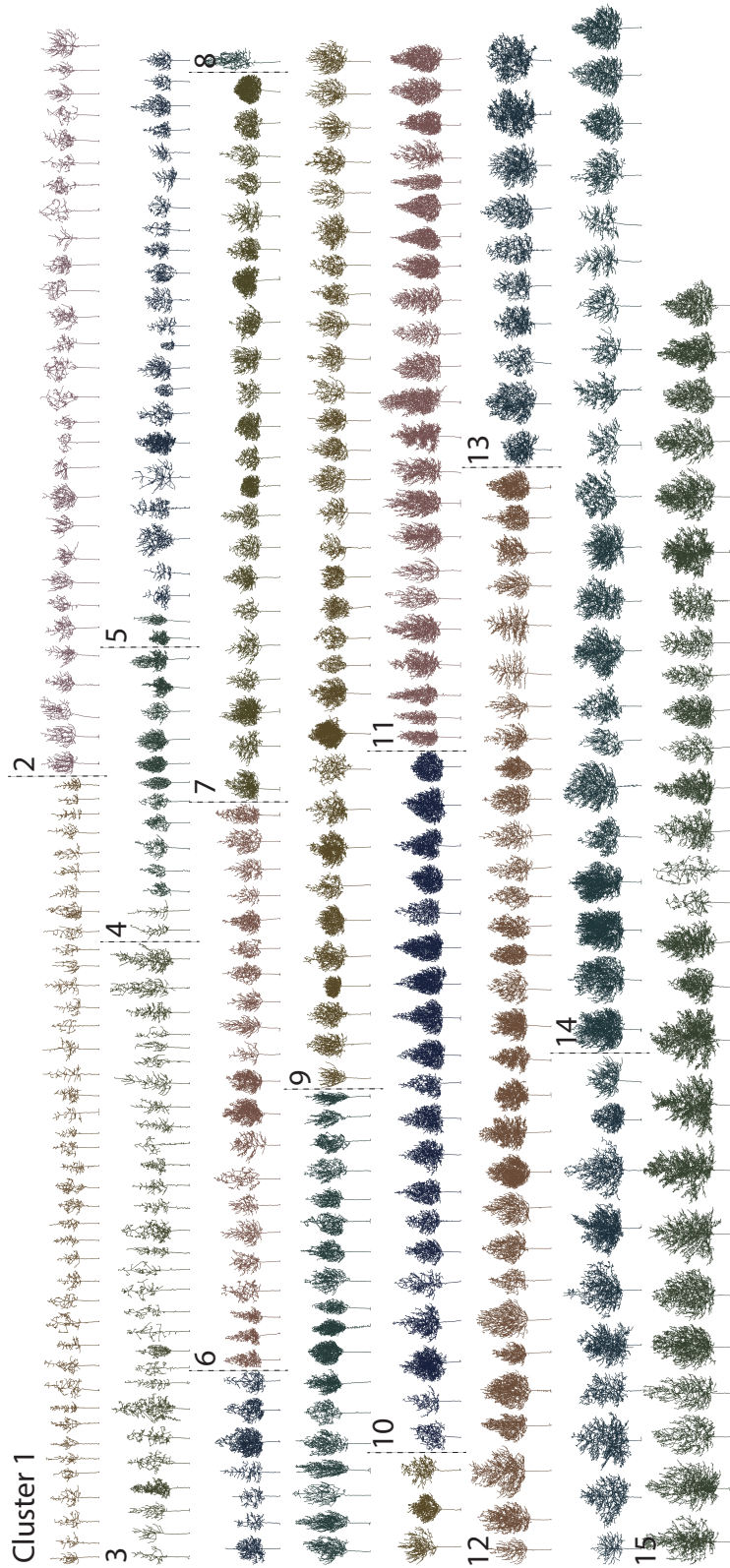


Figure D.7: Clusters of trees as identified by GMM based on our learned feature space.







# Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex

*The following 19 pages have been published as:*

Marissa A. Weis, Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G. Fahey, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, R. Clay Reid, Casey M. Schneider-Mizell, H. Sebastian Seung, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Jacob Reimer, Andreas S. Tolias, and Alexander S. Ecker. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex. *bioRxiv*, 2023.

*A summary of the motivation, results, and discussion can be found in Section 3.3 on page 32.*

## Abstract

Neurons in the neocortex exhibit astonishing morphological diversity which is critical for properly wiring neural circuits and giving neurons their functional properties. The extent to which the morphological diversity of excitatory neurons forms a continuum or is built from distinct clusters of cell types remains an open question. Here we took a data-driven approach using graph-based machine learning methods to obtain a low-dimensional morphological “bar code” describing more than 30,000 excitatory neurons in mouse visual areas V1, AL and RL that were reconstructed from the millimeter scale MICrONS serial-section electron microscopy volume. We found a set of principles that captured the morphological diversity of the dendrites of excitatory neurons. First, their morphologies varied with respect to three major axes: soma depth, total apical and total basal skeletal length. Second, neurons in layer 2/3 showed a strong trend of a decreasing width of their dendritic arbor and a smaller tuft with increasing cortical depth. Third, in layer 4, atufted neurons were primarily located in the primary visual cortex, while tufted neurons were more abundant in higher visual areas. Fourth, we discovered layer 4 neurons in V1 on the border to layer 5 which showed a tendency towards avoiding deeper layers with their dendrites. In summary, excitatory neurons exhibited a substantial degree of dendritic morphological variation, both within and across cortical layers, but this variation mostly formed a continuum, with only a few notable exceptions in deeper layers.

## Note

The subsequent paper corresponds to the submitted version that is currently under review and includes the following changes compared to the published version on *bioRxiv*: We rectified typing errors and made minor modifications in the phrasing of sentences for clarification. We added two paragraphs describing data and code availability in the methods section. We removed the line numbering.



# Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex

Marissa A. Weis<sup>1,2</sup>, Stelios Papadopoulos<sup>4,5</sup>, Laura Hansel<sup>1</sup>, Timo Lüddecke<sup>1</sup>, Brendan Celii<sup>4,5,10</sup>, Paul G. Fahey<sup>4,5</sup>, J. Alexander Bae<sup>6,8</sup>, Agnes L. Bodor<sup>9</sup>, Derrick Brittain<sup>9</sup>, JoAnn Buchanan<sup>9</sup>, Daniel J. Bumbarger<sup>9</sup>, Manuel A. Castro<sup>6</sup>, Forrest Collman<sup>9</sup>, Nuno Maçarico da Costa<sup>9</sup>, Sven Dorkenwald<sup>6,7</sup>, Leila Elabbady<sup>9</sup>, Akhilesh Halageri<sup>6</sup>, Zhen Jia<sup>6,7</sup>, Chris Jordan<sup>6</sup>, Dan Kapner<sup>9</sup>, Nico Kemnitz<sup>6</sup>, Sam Kinn<sup>9</sup>, Kisuk Lee<sup>6,11</sup>, Kai Li<sup>6,7</sup>, Ran Lu<sup>6</sup>, Thomas Macrina<sup>6,7</sup>, Gayathri Mahalingam<sup>9</sup>, Eric Mitchell<sup>6</sup>, Shanka Subhra Mondal<sup>6,8</sup>, Shang Mu<sup>6</sup>, Barak Nehoran<sup>6,7</sup>, Sergiy Popovych<sup>6,7</sup>, R. Clay Reid<sup>9</sup>, Casey M. Schneider-Mizell<sup>9</sup>, H. Sebastian Seung<sup>6,7</sup>, William Silversmith<sup>6</sup>, Marc Takeno<sup>9</sup>, Russel Torres<sup>9</sup>, Nicholas L. Turner<sup>6,7</sup>, William Wong<sup>6</sup>, Jingpeng Wu<sup>6</sup>, Wenjing Yin<sup>9</sup>, Szi-chieh Yu<sup>6</sup>, Jacob Reimer<sup>4,5</sup>, Andreas S. Tolias<sup>4,5</sup>, and Alexander S. Ecker<sup>1,3,\*</sup>

<sup>1</sup>Institute of Computer Science and Campus Institute Data Science, University of Göttingen, Germany

<sup>2</sup>Institute for Theoretical Physics, University of Tübingen, Germany

<sup>3</sup>Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany

<sup>4</sup>Department of Neuroscience, Baylor College of Medicine, Houston, USA

<sup>5</sup>Center for Neuroscience and AI, Baylor College of Medicine, Houston, USA

<sup>6</sup>Princeton Neuroscience Institute, Princeton University, USA

<sup>7</sup>Department of Computer Science, Princeton University, USA

<sup>8</sup>Department of Electrical Engineering, Princeton University, USA

<sup>9</sup>Allen Institute for Brain Science, Seattle, WA, USA

<sup>10</sup>Rice University, Houston, TX, USA

<sup>11</sup>Massachusetts Institute of Technology, Cambridge, MA, USA

\*Correspondence: ecker@cs.uni-goettingen.de

## Abstract

Neurons in the neocortex exhibit astonishing morphological diversity which is critical for properly wiring neural circuits and giving neurons their functional properties. The extent to which the morphological diversity of excitatory neurons forms a continuum or is built from distinct clusters of cell types remains an open question. Here we took a data-driven approach using graph-based machine learning methods to obtain a low-dimensional morphological “bar code” describing more than 30,000 excitatory neurons in mouse visual areas V1, AL and RL that were reconstructed from the millimeter scale MICrONS serial-section electron microscopy volume. We found a set of principles that captured the morphological diversity of the dendrites of excitatory neurons. First, their morphologies varied with respect to three major axes: soma depth, total apical and total basal skeletal length. Second, neurons in layer 2/3 showed a strong trend of a decreasing width of their dendritic arbor and a smaller tuft with increasing cortical depth. Third, in layer 4, atufted neurons were primarily located in the primary visual cortex, while tufted neurons were more abundant in higher visual areas. Fourth, we discovered layer 4 neurons in V1 on the border to layer 5 which showed a tendency towards avoiding deeper layers with their dendrites. In summary, excitatory neurons exhibited a substantial degree of dendritic morphological variation, both within and across cortical layers, but this variation mostly formed a continuum, with only a few notable exceptions in deeper layers.

## 1 Introduction

Neurons have incredibly complex and diverse shapes. Ever since Ramón y Cajal, neuroanatomists have studied their morphology [19] and have classified them into different types. From a computational point of view, their dendritic morphology constrains which inputs a neuron receives, how these inputs are integrated and, thus, which computations the neuron and the circuit it is part of can learn to perform.

Less than 15% of neocortical neurons are inhibitory, yet they are morphologically the most diverse and can be classified reliably into well-defined subtypes [3, 7]. The vast majority of cortical neurons are excitatory. Excitatory cells can be divided into spiny stellate and pyramidal cells [16]. Although pyramidal cells have a very stereotypical dendritic morphology, they exhibit a large degree of morphological diversity. Recent studies subdivide them into 10–20 cell types using manual classification [14] or clustering algorithms applied to dendritic morphological features [6, 10, 15].

Existing studies of excitatory morphologies have revealed a number of consistent patterns, such as the well-known thick-tufted pyramidal cells of layer 5 [8, 6, 10, 14, 15]. However, a commonly agreed-upon morphological taxonomy of excitatory neuron types is yet to be established. For instance, Markram et al. [14] describe two types of thick-tufted pyramidal cells based on the location of the bifurcation point of the apical dendrite (early vs. late). Later studies suggest that these form two ends of a continuous spectrum [10, 6]. Other authors even observe that morphological features overall do not form isolated clusters and suggest an organization into families with more continuous variation within families [21]. There are two main limitations of previous morphological characterizations: First, many rely on relatively small numbers of reconstructed neurons used to assess the morphological landscape. Second, they represent the dendritic morphology using summary statistics such as point counts, segment lengths, volumes, density profiles (so-called morphometrics; [15, 23, 13]) or graph-based topological measures [9]. These features were handcrafted by humans and may not capture all crucial axes of variation.

We here take a data-driven approach using a recently developed unsupervised representation learning approach [26] to extract a morphological feature representation directly from the dendritic skeleton. We apply this approach to a large-scale anatomical dataset [2] to obtain low-dimensional vector embeddings (“bar codes”) of more than 30,000 neurons in mouse visual areas V1, AL and RL. Our analysis suggests that excitatory neurons’ morphologies form a continuum, with notable exceptions such as layer 5 thick-tufted cells, and vary with respect to three major axes: soma depth, total apical and total basal skeletal length. Moreover, we found a number of novel morphological features in the upper layers: Neurons in layer 2/3 showed a strong trend of a decreasing width of their dendritic arbor and a smaller tuft with increasing cortical depth. In layer 4, morphologies showed area-specific variation: atufted neurons were primarily located in the primary visual cortex, while tufted neurons were more abundant in higher visual areas. Finally, layer 4 neurons in V1 on the border to layer 5 showed a tendency towards avoiding layer 5 with their dendrites.

## 2 Results

### 2.1 Self-supervised learning of embeddings for 30,000 excitatory neurons from visual cortex

Our goal was to perform a large-scale census of the dendritic morphologies of excitatory neurons without prescribing a-priori which morphological features to use. Therefore, we used machine learning techniques [26] to learn the features directly from the neuronal morphology.

Our starting point was a  $1.3 \times 0.87 \times 0.82 \text{ mm}^3$  volume of tissue from the visual cortex of an adult P75–87 mouse, which has been densely reconstructed using serial section electron microscopy [2]. This volume has been segmented into individual cells, including non-neuronal types and more than 54,000 neurons whose soma was located within the volume. From these detailed reconstructions we extracted each neuron’s dendritic tree and represented it as a skeleton (Fig. 1A) [1]: each neuron’s dendritic morphology was represented as a graph, where each node had a location in 3d space. This means we focused on the location and branching patterns of the dendritic tree, not fine-grained details of spines or synapses (see companion paper [22]), or any subcellular structures (see companion paper [4]).

Our next step was to embed these graphs into a vector space that defined a measure of similarity, such that similar morphologies were mapped onto nearby points in embedding space (Fig. 1B). To do so, we employed a recently developed self-supervised learning method called GRAPHDINO [26] that learns semantic representations of graphs without relying on manual annotations. The idea of this method is to generate two “views” of the same input by applying random identity-preserving transformations such as rotations around the vertical axis, slightly perturbing node locations or dropping sub-branches (Fig. 1B, top and bottom). Then both views are encoded using a neural network. The neural network is trained to map both views onto similar vector embeddings. For model training, the data was split into training, validation and test data to ensure that the model did not overfit (Sec. 4.5). The model outputs a 32-dimensional vector for each neuron that captures the morphological features of the neuron’s dendritic tree. Thus, each neuron is represented as a point in this 32-dimensional vector space (Fig. 1C).

At this stage, we performed another quality control step: Using the learned embeddings as a similarity metric between neurons, we clustered the neurons into 100 clusters and manually inspected the resulting clusters. We found a non-negligible fraction of neurons whose apical dendrite left the volume or was lost during tracing (see Methods for details). We removed neurons whose somata are in close proximity to the imaged volume boundary (Fig. 2A). Additionally, we used the clusters containing fragmented neurons as examples for broken neurons and trained a classifier to predict whether a neuron has reconstruction errors using the learned morphological embeddings as input features (Fig. 2B, Fig. A.2A, B). We then removed all neurons from the dataset that were classified as erroneous. Also, at this point we removed all interneurons from the dataset since we focused on excitatory neurons in this paper (Fig. 2C, Fig. A.2B, C).

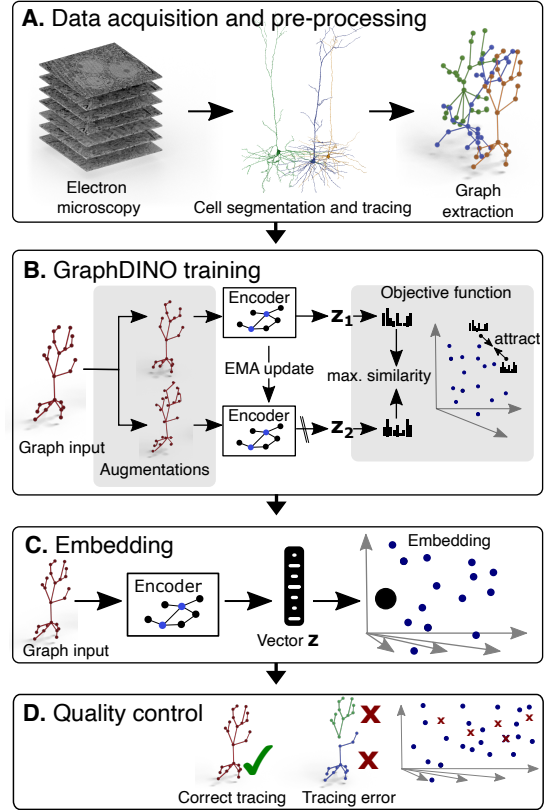
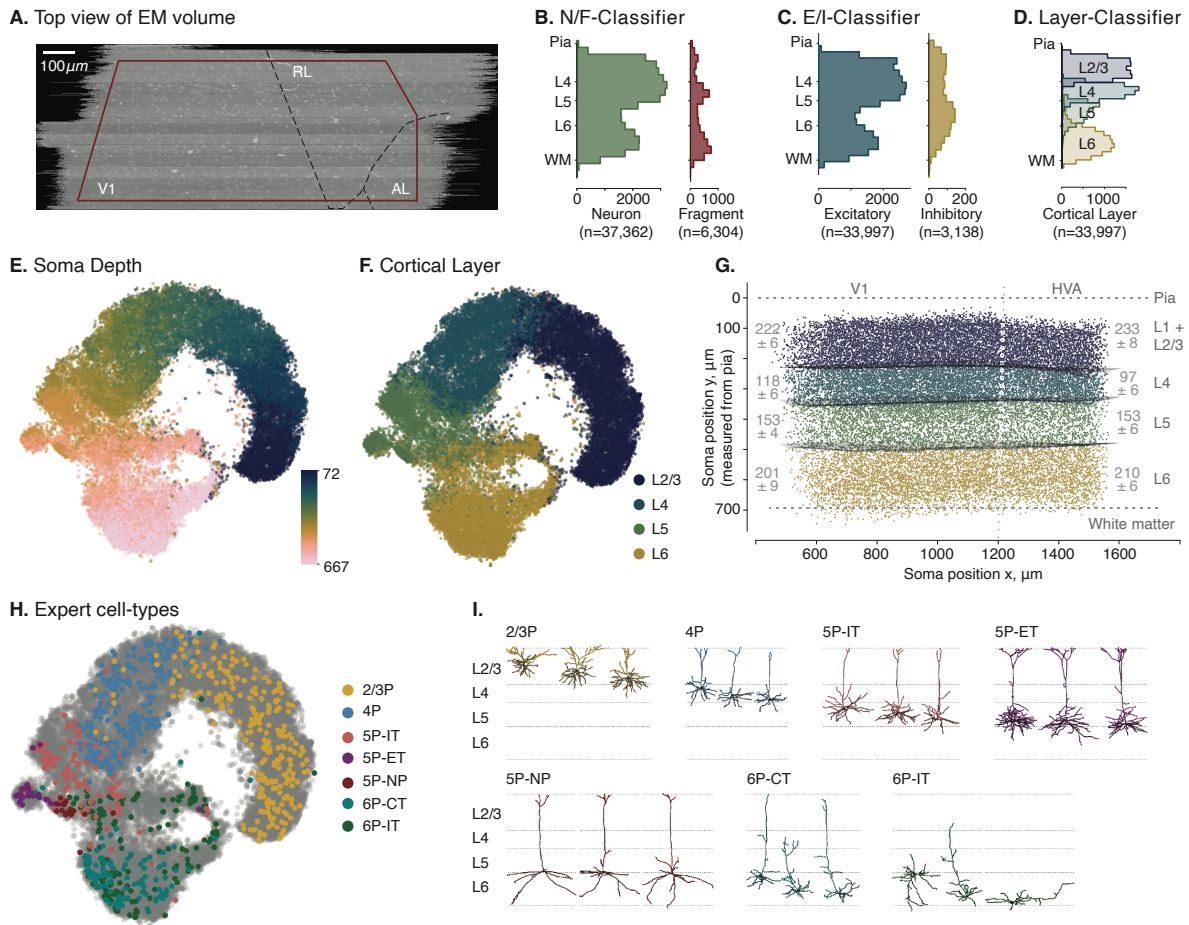


Figure 1: **Pipeline to generate vector embeddings for large scale datasets that capture the morphological features of the neurons’ dendritic trees.** **A.** Imaging of brain volume via electron microscopy and subsequent segmentation and tracing to render 3D meshes of individual neurons that are used for skeletonization. **B.** Self-supervised learning of low dimensional vector embeddings  $z_1, z_2$  that capture the essence of the 3D morphology of individual neurons using GraphDINO. Two augmented “views” of the neuron are input into the network, where the weights of one encoder (bottom) are an exponential moving average (EMA) of the other encoder (top). The objective is to maximize the similarity between the vector embeddings of both views. Vector embeddings of similar neurons are close to each other in latent space. **C.** An individual neuron is represented by its vector embedding as a point in the 32-dimensional vector space. **D.** Quality control to remove neurons with tracing errors.



**Figure 2: Visualization of soma depths and cortical layer assignments of excitatory neuronal morphologies showing mostly a continuum with distinct clusters only in deeper layers.** **A.** Top view of the EM volume with approximate visual areas indicated. All neurons with their soma origin within the red boundary were used for analysis. **B.** Distribution of complete neurons and fragments along cortical depth as determined by our classifier based on the morphological embeddings. **C.** Distribution of excitatory neurons and interneurons along cortical depth. **D.** Classifier prediction for cortical layer origin based on the learned morphological embeddings. **E.** t-SNE embedding (perplexity = 300) of the vector embeddings of excitatory neuronal morphologies colored by the respective soma depth of the neurons relative to the pia ( $n = 33,997$ ). **F.** t-SNE embedding colored by cortical layer assignments as predicted by a cross-validated classifier trained on the morphological embeddings as features and a subset of manually labeled excitatory neurons ( $n = 922$ ). **G.** Cross-section of the brain volume depicting soma positions of neurons colored by their assigned cortical layer. Cortical layer thicknesses for primary visual cortex (V1) (left) and higher visual areas (HVA) (right) given as mean  $\pm$  standard deviation. **H.** t-SNE embedding of excitatory neuronal morphologies colored by expert-defined cell types. **I.** Example morphologies of the expert-defined cell types.

## 2.2 Dendritic morphologies mostly form a continuum with distinct clusters only in deeper layers

We computed the vector embeddings of all excitatory neurons in our volume, which spanned the mouse visual areas V1, RL and AL (Fig. 2A). Dendritic morphology followed mostly a continuum that tracked the cortical depth of the soma from the pia, in counter-clockwise direction in Fig. 2E from layer 2/3 to layer 6. Note that the soma location within the cortex was not provided to the model, but the soma was centered on the origin of the coordinate system. Cells were mostly organized along a continuum and only a few distinct clusters were visible in the deeper layers. Therefore we decided to not assume the existence of clusters a-priori, as many previous studies [6, 15, 10, 14] did, but instead investigated the major axes of variation within the morphological embedding space.



The learned embedding space also reflects known cell types (Fig. 2H, I) that were assigned by expert neuroanatomists [22] using cortical origin of the somata and their long-range projection type (IT: intratelencephalic or intracortical; ET: extratelencephalic or subcortical projecting, NP: near projecting, and CT: corticothalamic). Note that both sources of information were not provided to the model, showing that dendritic morphology alone is sufficient to infer these broad cell type classes. One exception are the 6P-CT and 6P-IT cells, who were partly intermingled in embedding space. 6P-IT cells show a high variance in their dendritic morphology which in some cases are indistinguishable from 6P-CT cells when no information about the projection type is used (Fig. 2H, I).

Since an organization into cortical layers is well established, we separated cells by cortical layer to study the morphological rules of organization. We determined the layer boundaries by training a classifier using our 32-dimensional embeddings and a set of 922 neurons manually assigned to layers by experts (Fig. 2D, F, G). As expected, the inferred layer boundaries indicated that layer 4 was approximately 20% thicker in V1 than in higher visual areas RL and AL (Fig. 2G; mean  $\pm$  SD:  $118 \pm 6 \mu\text{m}$  in V1 vs.  $97 \pm 6 \mu\text{m}$  in HVA), the difference being compensated for by layers 2/3 and 6 each being approximately  $10 \mu\text{m}$  thinner. In the following we proceed by assigning neurons to layers based on their soma location relative to these inferred boundaries.

To visualize the main axes of morphological variation within each layer, we performed nonlinear dimensionality reduction using t-distributed stochastic neighbor embedding (t-SNE; [24]) and identified a number of morphological features that formed major axes of variation within the two-dimensional space (Fig. 4).

What do these axes of variation in the two-dimensional t-SNE embeddings mean in human-interpretable terms? To answer this question, we looked for morphological metrics that formed gradients within the t-SNE embedding space. Based on visual inspection, we found the following six morphological metrics to account well for a large fraction of the dendritic morphological diversity in our dataset (see Fig. 3 for an illustration): (1) depth of the soma relative to the pia, (2) height of the cell, (3) total length of the apical dendrites, (4) width of the apical dendritic tree, (5) total length of the basal dendrites, and (6) location of the basal dendritic tree relative to the soma (“basal bias”).

### 2.3 Layer 2/3: Width and length of apical dendrites decrease with depth

We start with layer 2/3 (L2/3) where we found a continuum of dendritic morphologies that formed a gradient from superficial to deep, with deeper neurons (in terms of soma depth) becoming thinner and less tufted (Fig. 4A1, A2, A3). The strongest predictors of the embeddings were the depth of the soma relative to the pia and the total height of the cell (coefficient of determination  $R^2 > 0.9$ ; Fig. 4B). These two metrics were also strongly correlated (Spearman’s rank correlation coefficient,  $\rho = 0.93$ ; Fig. 4C), since nearly all L2/3 cells had an apical dendritic tree that reached to the pial surface (see example morphologies in Fig. 4A, top). L2/3 cells varied in terms of their degree of tuftedness: both total length and width of their apical tuft decreased with the depth of the soma relative to the pia (Fig. 4A2, A3). L2/3 cells also varied along a third axis: the skeletal length of their basal dendrites (Fig. 4A4), but this property was not strongly correlated with either soma depth or shape of the apical dendrites (Fig. 4C).

### 2.4 Layer 4: Small or no tufts and some cells’ basal dendrites avoid layer 5

The dendritic morphology of layer 4 (L4) was again mostly a continuum and appeared to be a continuation of the trends from L2/3: The skeletal length of the apical dendrites was shorter, on average, than that of most L2/3 cells (Fig. 4A2) and approximately 20% of the cells were untufted. Within L4 the total apical skeletal length was not correlated with the depth of the soma ( $\rho = 0.0$ ; Fig. 4C), suggesting that it forms an independent axis of variation. There was also quite some variability in terms of the total length of the basal dendritic tree, but – as in L2/3 – it was not correlated with any of the other properties.

Our data-driven embeddings revealed another axis of variation that had previously not been considered important: the location of the basal dendritic tree relative to the soma (“basal bias”; Fig. 3). We found that many L4 cells avoided reaching

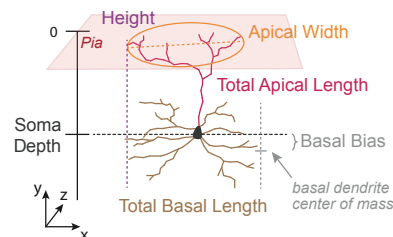


Figure 3: **Schematic of morphometric descriptors computed from neuronal skeletons and their labeled compartments.** SOMA DEPTH. Depth of the centroid of the soma relative to the pia. HEIGHT. Extent of the cell in y-axis. TOTAL APICAL LENGTH. Total length of the skeletal branches of the apical dendrites. APICAL WIDTH. Maximum extent of the apical dendritic tree in the xz-plane. TOTAL BASAL LENGTH. Total length of the skeletal branches of the basal dendrites. BASAL BIAS. Depth in y-axis of center of mass of basal dendrites relative to the soma.

into L5 with their dendrites (Fig. 4A3). As a result, the depth of the basal dendrites was anticorrelated with the depth of the soma ( $\rho = -0.29$ ; Fig. 4A3 and Fig. 4C). We will come back to this observation later (see Sec. 2.8).

## 2.5 Layer 5: Thick-tufted cells stand out

Layer 5 (L5) showed a less uniformly distributed latent space than L2/3 or L4. Most distinct was the cluster of well-known thick-tufted pyramidal tract (PT) cells [8, 6, 10, 14, 15] on the bottom right (Fig. 4A4, light green points), also known as extratelencephalic (ET) projection neurons. These cells accounted for approximately 17% of the cells within L5 (based on a classifier trained on a smaller, manually annotated subset of the data; see Methods). They were restricted almost exclusively to the deeper half of L5 (Fig. 4A1, A4 inset 2, C inset top right) and compared to other L5 cells they have the longest skeleton for all three dendritic compartments: apical, basal and oblique.

Another morphologically distinct type of cell was apparent: the near-projecting (NP) cells [11, 6] with their long and sparse basal dendrites (Fig. 4A4, inset 3). These cells accounted for approximately 4% of the cells within L5. They tended to send their dendrites deeper (relative to the soma), had little or no obliques and tended to have small or no apical tufts. However, the dendritic morphology of this cell type appeared to represent the extreme of a continuum rather than being clearly distinct from other L5 cells.

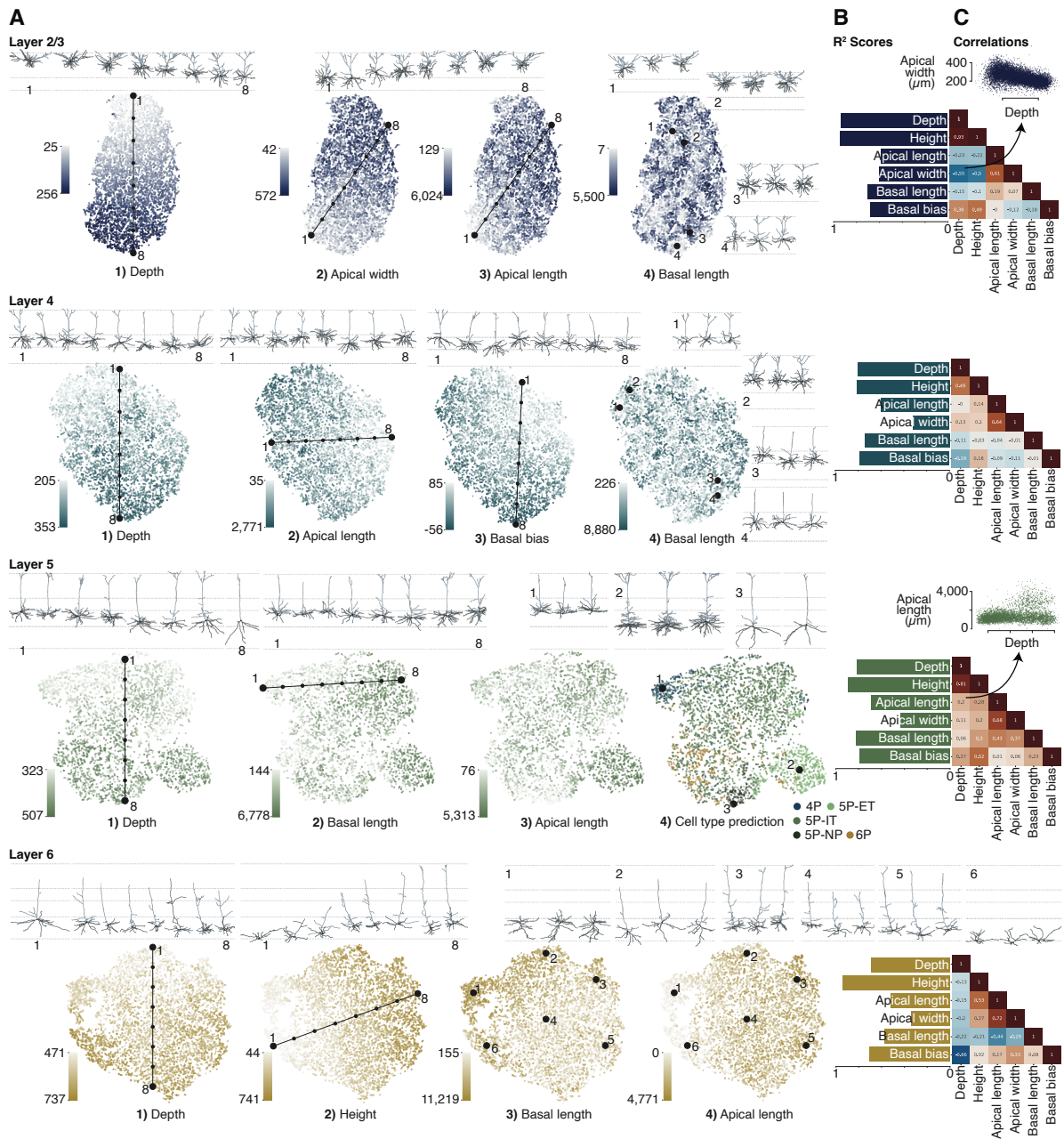
The remaining roughly 80% of the cells within L5 varied continuously in terms of the skeletal length of the different dendritic compartments. While there was a correlation between apical and basal skeletal length (apical vs. basal:  $\rho = 0.43$ ; Fig. 4C), there was also a significant degree of diversity. Within this group there was no strong correlation of morphological features with the location of the soma within L5 (depth vs. apical length  $\rho = 0.2$ , depth vs. basal  $\rho = 0.06$ ; Fig. 4C).

In upper L5 we found a group of cells that resembled the L4 cells whose dendrites avoid L5 (Fig. 4A4, inset 1). This type of cell was restricted to the uppermost portion of L5 and morphologically resembled L4 cells by being mostly atufted. We refer to these cells as displaced L4 cells. The presence of these cells suggests there are no precise laminar boundaries based on morphological features of neurons, but instead different layers blend into one another, a finding also observed by other authors [15, 4].

## 2.6 Layer 6: Long and narrow, oblique and inverted pyramidal neurons

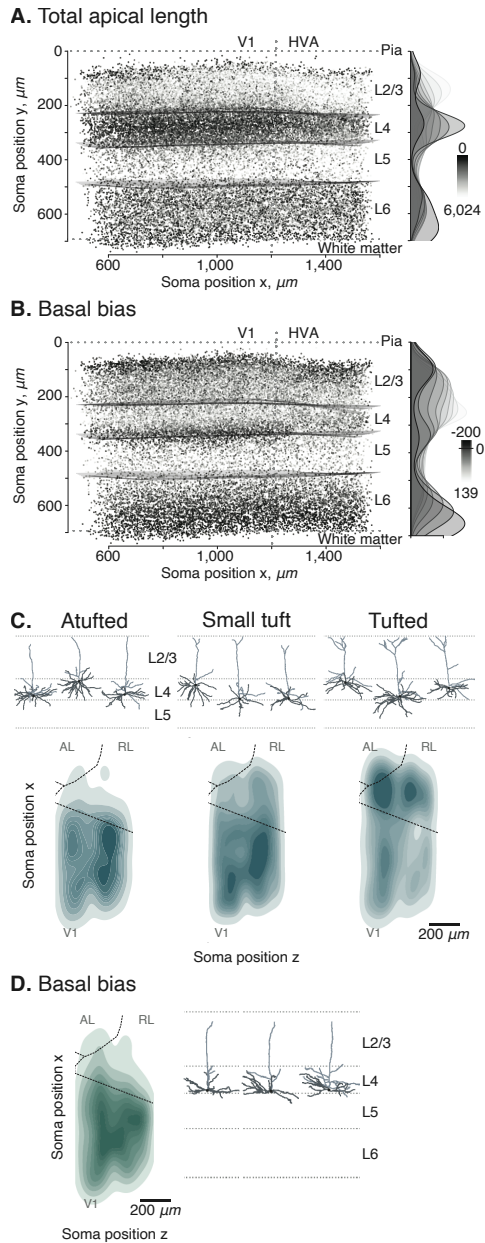
Dendritic morphology in layer 6 (L6) also formed a continuum with a large degree of morphological diversity. The dominant feature of L6 was the large variety of cell heights ( $R^2 > 0.9$ ; Fig. 4B). Overall, the height of a cell was not strongly correlated with its soma's location within L6 ( $\rho = -0.13$ ; Fig. 4C). Unlike other layers, where the apical dendrites usually reach all the way up to layer 1, many cells in L6 have shorter apical dendrites. However, due to tracing errors, our analysis overestimated the number of such short cells. We therefore manually inspected 183 putative untufted early-terminating neurons within L6 and found that, among those, 45% were incompletely traced, whereas 55% were true untufted cells whose apical dendrite terminated clearly below L1.

As described previously [6, 15], the dendritic tree of L6 cells is narrower than in the layers above. Also consistent with previous work, we found a substantial number of horizontal and inverted pyramidal neurons, where the apical dendrite points sideways or downwards, respectively (Fig. 4A4, inset 1 & 6).



**Figure 4: t-SNE visualization of vector embeddings per cortical layer reveal axes of variation in neuronal morphologies.** **A.** t-SNE embeddings per layer colored by percentiles of various morphometric descriptors with example neuronal morphologies along the axis of variation displayed above the embedding. **B.**  $R^2$  scores of the six morphometric descriptors (see Fig. 3) per layer showing the strength as predictors of the 32d embeddings. **C.** Spearman's rank correlation coefficient between morphometric descriptors per layer. **Layer 2/3** (blue) Continuum of dendritic morphologies with thinner and less tufted neurons in increasing distance to the pia. **Layer 4** (turquoise) Continuation of L2/3 trends with shorter apical dendrites and more tufted cells. Many cells avoid reaching dendrites into L5 (basal bias). **Layer 5** (green) Clustering of thick-tufted ET and NP cells. Upper L5 cells resemble L4 cells that avoid reaching into L5, indicating too strict laminar borders. **Layer 6** (orange) Continuum with a large morphological diversity e.g. in cell heights, and existence of horizontal and inverted pyramidal neurons.

## 2.7 Pyramidal neurons are less tufted in V1 than in higher visual areas



**Figure 5: Inter-areal differences between primary visual cortex (V1) and higher visual areas (HVAs).** **A.** Side view showing apical skeletal length, color-coded by percentiles (dark=short, bright=long). Projection from the side orthogonal to the V1/HVA border after a 14 degree rotation around y-axis (vertical dashed line); top: pia; bottom: white matter. **B.** Side view showing basal bias (as in A) (dark=negative basal bias: center of mass of basal dendrites is above the soma; bright=positive basal bias: center of mass of basal dendrites is below soma). **C.** Top view showing density of untufted (left), small tufted (middle) and tufted (right) L4 cells. Untufted neurons are mostly confined to V1, while tufted neurons are more abundant in HVA. Dashed lines: area borders between primary visual cortex (V1), anterolateral area (AL) and rostralateral area (RL), estimated from reversal of the retinotopic map measured using functional imaging. **D.** Top view (as in C) showing horizontal distribution of L4 cells whose dendrites avoid reaching into L5 and who are mostly located in V1.

After our layer-wise survey of excitatory neurons' morphological features, we next asked whether there are inter-areal differences between primary visual cortex (V1) and higher visual areas (HVAs). The total length the apical dendrites of neurons in V1 was significantly shorter than for neurons in HVA (Fig. 5A): For L2/3, neurons in V1 had on average 16% shorter apical branches than in HVA (mean  $\pm$  SD:  $1,423 \pm 440 \mu\text{m}$  in V1 vs.  $1,688 \pm 554 \mu\text{m}$  in HVA; *t*-test:  $p < 0.0025$ , Cohen's  $d = 0.53$ ). Similarly, L4 neurons in V1 had on average 16% shorter apical branches than in HVA ( $851 \pm 264 \mu\text{m}$  vs.  $1,019 \pm 313 \mu\text{m}$ ;  $p < 0.0025$ ,  $d = 0.58$ ). In L5, neurons in V1 had on average 14% shorter apical branches than L5 neurons in HVA ( $1,326 \pm 661 \mu\text{m}$  vs.  $1,549 \pm 745 \mu\text{m}$ ;  $p < 0.0025$ ,  $d = 0.32$ ). While the trend continued in L6, the difference in apical length between V1 and HVA neurons was smaller. There was only a 4% increase in apical length in HVA compared to V1 ( $1,112 \pm 383 \mu\text{m}$  vs.  $1,159 \pm 397 \mu\text{m}$ ;  $p < 0.0025$ ,  $d = 0.12$ ). For this analysis, only neurons with identified apical dendrites were taken into account (see companion paper; Celii et al. [1]).

Upon closer inspection, we observed that L4 contained substantially more untufted neurons than higher visual areas RL and AL (Fig. 5A). We clustered each layer's morphological embeddings into 15 clusters using a Gaussian Mixture Model and looked for clusters that were restricted to particular brain areas. Clusters that were clearly confined to V1 or HVAs were primarily found in L4. When classifying (manually, at the cluster-level) L4 neurons into untufted, small tufted and tufted, we observed that untufted neurons were almost exclusively located in V1, while tufted neurons were more frequent in HVAs (Fig. 5C).

## 2.8 Layer 4 cells avoiding layer 5 are located primarily in primary visual cortex

We observed a second area difference, which was related to the novel morphological cell type in L4. Recall that these cells' dendrites avoid reaching into L5. Interestingly, these cells were located in a very narrow strip of around  $50 \mu\text{m}$  above the border between L4 and L5 (Fig. 5B). Moreover, they were also untufted and almost exclusively located in V1 (Fig. 5D).

### 3 Discussion

In summary, our data-driven unsupervised learning approach identified the known morphological features of excitatory cortical neurons' dendrites and enabled us to make four novel observations: (1) Superficial L2/3 neurons are wider than deep ones; (2) L4 neurons in V1 are less tufted than those in HVAs; (3) a novel untufted L4 cell type that is specific to V1 whose basal dendrites avoid reaching into L5; (4) excitatory cortical neurons form mostly a continuum with respect to dendritic morphology, with some notable exceptions.

First, our finding that superficial L2/3 neurons are wider than deeper ones is clearly visible in the data both qualitatively and quantitatively. A similar observation has been made recently in concurrent work [25].

Second, the trend of deeper neurons being less tufted continues into L4 where a substantial number of cells are completely untufted. Here we see a differentiation with respect to brain areas: completely untufted cells are mostly restricted to V1 while HVA neurons in L4 tend to be more tufted. Why would V1 neurons be less tufted than those in higher visual areas? V1 – as the first cortical area for visual information processing – and L4 – as the input layer, in particular – might be less modulated by feedback connections than other layers and higher visual areas. Therefore, these neurons might sample the feedback input in L1 less than other neurons.

Third, we found that some neurons in L4 of V1 avoid reaching into L5 with their dendrites. To our knowledge, this morphological pattern has not been described before in the visual cortex. Retrospectively, it can be observed in Gouwens and colleagues' data: their spiny m-types 4 and 5, which are small- or untufted L4 neurons, show a positive basal bias (assuming their "basal bias  $y$ " describes the same property; Gouwens et al. [6]; Suppl. Fig. 15). What function could this avoiding L5 have? Similarly to the nonexistent tuft of these neurons, avoiding L5 could support these neurons in focusing on the thalamic input (which targets primarily L4) and, thus, represent and distribute the feedforward drive within the local circuit. It is therefore tempting to speculate that these untufted, L5-avoiding L4 neurons might be precursors of spiny stellate cells, which are nearly absent in the mouse visual cortex [20], but exist only in somewhat more developed sensory areas like barrel cortex or in cat and primate V1.

Fourth, except from the well-known L5 thick-tufted extratelencephalic (ET) projection neurons that form a cluster in L5, our data and methods suggest that excitatory neurons in the mouse visual cortex form mostly a continuum with respect to dendritic morphology. This result does not rule out the possibility that there are in fact distinct types; it simply suggests that features beyond dendritic morphology need to be taken into account to clearly identify them. For instance, the results of [22] suggest that the 5P-NP cells can be separated from other layer 5 pyramidal neurons by considering the class of interneurons that target them. It is also not guaranteed that our data-driven method identifies all relevant morphological features. Every method has (implicit or explicit) inductive biases. We tried to stay clear of explicit human-defined features, but by choosing a graph-based input representation we provide different inductive biases than, for instance, a voxel-based representation or one based on point clouds. However, the fact that we could reconcile known morphological features, discover novel ones and achieve excellent classification accuracy on an annotated subset of the data suggests that our learned embeddings indeed contain a rich and expressive representation of a neuron's dendritic morphology.

Our observation that morphologies forms mostly a continuum is in line with a recent study in motor cortex examining the relationship between transcriptomic and morphological cell types [21]. These authors found a substantial degree of (continuous) morphological variation within transcriptomically defined cell types. Moreover, they found that morphological and transcriptomic features correlated, suggesting a more fine-grained organization of neurons into a relatively small number of distinct and broad "families," each of which exhibits substantial continuous variation among its family members. Our analysis supports this notion: excitatory cells can be mostly separated by layers into roughly a handful of families, each of which contains a substantial degree of variation in terms of morphology, which might also co-vary with other modalities.

## 4 Methods

### 4.1 Dataset

The dataset consists of a  $1.3 \times 0.87 \times 0.82 \text{ mm}^3$  volume of tissue from the visual cortex of an adult P75-87 mouse, which has been densely reconstructed using serial section electron microscopy (EM) [2]. We here use the subvolume 65, which covers approximately  $1.3 \times 0.56 \times 0.82 \text{ mm}^3$ . It includes all layers of cortex and spans primary visual cortex (V1) and two higher visual areas, anterolateral area (AL) and rostrolateral area (RL). We refer to the original paper on the dataset [2] for details on the identification and morphological reconstruction of individual neurons.

### 4.2 Skeletonization and cell compartment label assignment

The EM reconstructions yield neuronal meshes. These meshes might be incomplete or exhibit different kinds of errors including merges of other neuronal or non-neuronal compartments onto the neurons. Therefore an automatic proof-reading pipeline that results in neuronal skeletons was executed (companion paper; Celii et al. [1]).

For the skeletal detection from the reconstructed meshes, the meshes were first downsampled to 25% of their resolution and made watertight. Then glia and nuclei meshes were identified and removed. For the remaining meshes the locations of the somata were identified using a soma detection algorithm [27]. Each neurite submesh was then skeletonized using a custom skeletonization algorithm which transformed axonal and dendritic processes into a series of line segments to obtain the skeleton (companion paper; Celii et al. [1]). For each skeleton, the highest probability axon subgraph was determined and all other non-soma nodes were labeled as dendrites. A final heuristic algorithm classifies subgraphs of dendritic nodes into compartments, such as apical trunks generally projecting from the top half of somas and with a general upward trajectory and obliques as projections off the apical trunks at an approximate 90 degree angle. For further details on the compartment label assignment please see companion paper [1].

### 4.3 Coordinate transformations

The EM volume is not perfectly aligned. First, the pial surface is not a horizontal plane parallel to the  $(x, z)$  plane, but is instead slightly tilted. Second, the thickness of the cortex varies across the volume such that the distance from pia to white matter is not constant. Without any pre-processing, an unsupervised learning algorithm would pick up these differences and, for instance, find differences of layer 6 neurons across the volume simply because in some parts of the volume they tend to be located deeper than in others and their apical dendrites that reach to layer 1 tend to be larger. Using *relative* coordinates can solve such issues if pia and white matter correspond to planes (approximately) parallel to the  $(x, z)$  plane. To transform our coordinate system in such standardized coordinates, we first applied a rotation about the  $z$ -axis of 3.5 degrees. This transformation removes the systematic rotation with respect to the native axes (Fig. A.1B). To standardize measurements across depth ( $y$  axis) and to account for differential thickness of the cortex, we estimated the best linear fit for both pial surface and white matter boundary by using a set of manually placed points, which were located on a regular grid along  $(x, z)$  with a spacing of  $25\mu\text{m}$ . Then for each  $(x, z)$  coordinate, the  $y$  coordinate was normalized such that the pia's  $z$  coordinate corresponds to the average depth of the pia and the same for the white matter. This transformation results in an approximation of the volume where pia and white matter boundaries are horizontal planes orthogonal to the  $y$  axis and parallel to the  $(x, z)$  plane. Fig. A.1C shows example neurons before and after normalization. All training and subsequent analysis were performed on this pre-processed data.

### 4.4 Expert cell type labels

For a subset of the neurons in the volume experts labeled neurons according the following cell types: layer 2/3 and 4 pyramidal neurons, layer 5 near-projecting (NP), extratelencephalic (ET) and intratelencephalic (IT) neurons, layer 6 intratelencephalic (IT) and cortico-thalamic (CT) neurons, Martinotti cells (MC), basket cells (BC), bipolar cells (BPC) and neurogliaform cells (NGC). Cell types were assigned based on visual inspection of individual cells taking into account morphology, synapses and connectivity, nucleus features and their  $(x, y, z)$  location. All neurons were taken from one  $100\mu\text{m}$  column in the primary visual cortex (see companion paper, Schneider-Mizell et al. [22]). We did not use neurons with expert labels to train GRAPHDINO, but used them only for evaluation.

## 4.5 Morphological feature learning using GRAPHDINO

For learning morphological features in an unsupervised, purely data-driven way, we used a recently developed machine learning method called GRAPHDINO [26]. GRAPHDINO maps the skeleton graph of a neuron onto a 32-dimensional feature vector, which we colloquially refer to as the neuron’s “bar code”. For training GRAPHDINO, each neuron’s skeleton is represented as an undirected graph  $G = (V, E)$ .  $V$  is the set of nodes  $\{v_i\}_{i=1}^N$  and  $E$  the set of undirected edges  $E = \{e_{ij} = (v_i, v_j)\}$  that connect two nodes  $v_i, v_j$ . Each node has a feature vector attached to it that holds the 3d Cartesian coordinate of the node, relative to the soma of the neuron. The soma has the coordinate  $(0, 0, 0)$ , i.e. is at the origin of the coordinate system. Because axons are not well reconstructed in the data yet, we focus on the dendritic skeleton only and remove segments labeled as axon. We train GRAPHDINO on a subset of the dataset, retaining 5,113 neurons for validation and 2,941 neurons for testing. The test set is chosen to contain the 1,011 neurons that were labeled by expert anatomists into morphological cell types (Sec. 4.4; [22]). The training and validation sets are i.i.d. sampled from the remaining neurons with a 90%-10% split (Fig. A.3).

GRAPHDINO is trained by generating two “views” of the same input graph by applying random identity-preserving transformations (described below). These two views are both encoded by the same neural network. The training objective is to maximize the similarity between the embeddings of these two views. To obtain the two views of one input graph, we subsampled the graph, randomly rotated it around the  $y$ -axis (orthogonal to pia), dropped subbranches and perturbed node locations. When subsampling the graph, we randomly dropped all but 200 nodes, always retaining the branching points. Rotations around the  $y$ -axis were uniformly distributed around the circle. During subbranch deletion we removed  $n = 5$  subbranches. For node location jittering we used  $\sigma = 1$ . In addition the entire graph was randomly translated with  $\sigma = 1$ . For further details on the augmentation strategies, see Weis et al. [26].

The ADJACENCY-CONDITIONED ATTENTION network architecture we used had seven AC-ATTENTION layers with four attention heads each. The dimensionality of the latent representation  $z$  was set to 32 and the dimensionality of the projection  $p$  was 5,000. All other architecture details are as described in the original paper [26]. For training we used the Adam optimizer [12] with a batch size of 128 for 50,000 iterations. The learning rate increased linearly to  $10^{-3}$  during the first 1,000 iterations and then decayed using an exponential schedule with a decay rate of 0.5.

## 4.6 Morphological clustering

For qualitative inspection of the data and the analyses in Fig. 5C+D we clustered the neurons using the learned vector embedding of each neuron’s morphological features. We fit a Gaussian Mixture model (GMM) with diagonal covariance matrix using `scipy` [17] on the whole dataset as well as per cortical layer using 60 clusters and 15 clusters, respectively. As we found no evidence that these clusters (or any other clustering with fewer or more clusters) represent distinct cell types, we do not use this clustering to define cell types, but rather think of them as modes or representing groups of neurons with similar morphological features.

## 4.7 Data quality control steps

The dataset was generated by automatic segmentation of EM images and subsequent automatic processing into skeletons. As a consequence, not all cells were reconstructed perfectly. There was a significant fraction of wrongly merged or incompletely segmented cells. We used a combination of our learned GRAPHDINO embeddings and supervised classifiers trained on a subset of the neurons ( $n = 1,011$ ) which were manually proofread and annotated by experts (see Sec. 4.4 and companion paper, Schneider-Mizell et al. [22]). Our quality control pipeline was as follows: First, we computed GRAPHDINO embeddings on the full dataset of 54,192 neurons (including both excitatory and inhibitory neurons). Next, we removed neurons which were close to the boundaries of the volume, as these neurons were only partly reconstructed. After this step we were left with 43,666 neurons. Within this dataset we identified neurons which were incorrectly reconstructed using a supervised classifier described in the next section, reducing the dataset to 37,362 neurons. Subsequently, we identified interneurons using a supervised classifier described in the next section, reducing the dataset to 33,997 excitatory neurons. Finally, on this dataset we manually proofread around 480 atufted neurons. As a result, we identified and removed another set of 2,684 neurons whose reconstructions were incomplete, leaving us with a final sample size of 31,313 putative excitatory and correctly reconstructed neurons for our main analyses.

## 4.8 Supervised classifiers

To identify reconstruction errors and interneurons, we used a subset of the dataset ( $n = 1,011$ ) that was manually proofread and annotated with cell type labels by experts (see Sec. 4.4 and companion paper, Schneider-Mizell et al. [22]). Based on these and additional neurons we identified, we trained classifiers to detect segmentation errors, inhibitory cells and cortical layer membership using our learned 32-dimensional vector embeddings of the neurons' skeletons (see Sec. 4.5). In our subsequent analysis, we focused on neurons that were identified as complete and excitatory by our classifier. We used the inferred cortical layer labels to perform layer-specific analyses.

For all classifiers, we used ten-fold cross-validation on a grid search to find the best hyperparameters. We tested logistic regression with the following hyperparameters: type of regularization (none, L1, L2 or elastic net), regularization weight ( $C \in 0.5, 1, 3, 5, 10, 20, 30$ ) and whether to use class weights that are inversely proportional to class frequencies or no weights. In addition, we tested support vector machines (SVMs) with the following hyperparameters: type of kernel (Linear, RBF or polynomial), L2 regularization weight ( $C \in 0.5, 1, 3, 5, 10, 20, 30$ ) and degree of polynomial ( $d \in 2, 3, 5, 7, 10, 20$  for the polynomial kernel and whether to use class weights or no weights. After having determined the optimal hyperparameters using cross-validation, we retrained the classifier using the optimal hyperparameters on its entire training set.

**Removal of fragmented neurons.** To remove fragmented neurons prior to analysis, we trained a classifier to differentiate between the manually proofread neurons from all layers ( $n = 1,011$ ) and fragmented cells ( $n = 240$ ). We identified fragmented cells using our clustering of the vector embeddings of the whole dataset without boundary neurons ( $n = 43,666$ ) into 25 clusters per layer and manually identified clusters that contained fragmented cells (2–3 clusters per layer). We then sampled 60 fragmented cells per layer as training data for our classifier.

We trained a support vector machine (SVM) using cross-validation as described above. Its cross-validated accuracy was 95%. The best hyperparameters were: polynomial kernel of degree 4 and  $C = 3$ . We used those hyperparameters to retrain on the full training set of 1,251 neurons. Using this classifier, we inferred whether a neuron is fragmented for the entire dataset ( $n = 43,666$ ). We then removed cells predicted to be fragmented ( $n = 6,304$ ) from subsequent analyses.

To validate the classification into fragmented and whole cells, we manually inspected ten neurons that were not in “fragmented” clusters before classification, but were flagged as fragmented by the classifier. Nine out of the ten had missing segments due to segmentation errors or due to apical dendrites leaving the volume.

**Removal of inhibitory neurons.** Analogously, we trained a classifier to predict whether a neuron was excitatory or inhibitory by using the manually proofread and annotated neurons ( $n = 1,011$ ) (Sec. 4.4). As input features to the classifier we used our learned bar codes and additionally two morphometric features: synaptic density on apical shafts (number of synapses per micrometer of skeletal length except those located on spines) and spine density (number of spines per micrometer of skeletal length). These two features have been shown to separate excitatory from inhibitory neurons well in previous work (see companion paper, Celii et al. [1]). The annotated dataset contained 922 excitatory and 89 inhibitory neurons.

We trained a logistic regression. Its cross-validated balanced accuracy was 99%. The best hyperparameters were: L2 regularization ( $C = 5$ ) and using class weights. We used those hyperparameters to retrain on the full training set of 1,011 neurons. Using this classifier, we inferred whether a neuron was excitatory or inhibitory for the entire dataset after removing fragmented cells and after removal of 227 neurons that did not have spine and synapse densities available ( $n = 37,135$ ). We then removed all inhibitory cells from subsequent analyses ( $n = 3,138$ ).

**Inference of cortical layers.** To determine cortical layer labels for the entire dataset, we followed a two-stage procedure. First, we inferred the layer of each neuron using a trained classifier. Then we determined anatomical layer boundaries based on the optimal cortical depth that separates adjacent layers.

We first trained a SVM classifier for excitatory cells on the 922 manually annotated excitatory neurons by pooling the cell type labels per layer. Its cross-validated balanced accuracy was 89%. The best hyperparameters were: polynomial kernel of degree 5,  $C = 3$ . Using this classifier, we inferred the cortical layer of all excitatory neurons ( $n = 33,997$ ; Fig. 2).

The spatial distribution of inferred layer assignments was overall well confined to their respective layers. As expected, there was some spatial overlap of labels at the boundaries, since layer boundaries are not sharp. We nevertheless opted for assigned neurons to layers based on their anatomical location rather than their inferred label. To do so, we determined the optimal piecewise linear function that separates two consecutive layers. At the end, the layer assignments were purely based on the soma depth of each neuron relative to the inferred layer boundaries – not on the classifier output.



**Inference of coarse cell type labels.** In Fig. 4 we show cell type labels for layer 5. These were determined by training a support vector machine to classify the excitatory neurons into cell types using the 922 manually annotated neurons. The cross-validated balanced accuracy of this classifier was 85%. The best hyperparameters were: polynomial kernel of degree 2,  $C = 20$ , using class weights. Using this classifier, we inferred cell type labels for all excitatory neurons ( $n=33,997$ ).

#### 4.9 Manual validation of apical skeletons

We found a significant fraction of atufted neurons across layers 4–6. To determine the extent to which these cells are actually atufted or an artifact of incomplete reconstructions, we manually inspected ca. 480 neurons in Neuroglancer [5] with respect to the validity of their apical termination. During manual inspection, we annotated neurons’ reconstruction as “naturally terminating,” “out-of-bounds,” “reconstruction issue” or “unsegmented region.” Reconstruction issues are the case where the EM slice was segmented correctly, but the tracing missed to connect two parts of the same neuron. Unsegmented regions are the case where one or multiple EM images or parts thereof were not segmented correctly and therefore the neuron could not be traced correctly. In addition, we classified the neurons as either “atufted,” “small tufted” or “tufted,” both before validation and after correcting reconstruction errors.

For layer 4, we inspected 120 atufted neurons. Of those, 64% had missing segments on their apical dendrites and 36% had a natural termination. Note, however, that 74% of the neurons had a consistent tuft before and after validation. Even though parts of the apical dendrite were missing, qualitatively the degree of tuftedness did not change. For atufted neurons this means that their apical dendrite merely terminated early, but this reconstruction error did not change their classification as atufted. In layer 4, neurons with a natural termination ended more superficially than neurons with missing segments. We therefore excluded L4 neurons from the analysis whose apicals ended more than 154 micrometers below the pia to exclude neurons with reconstruction errors from our analysis. This threshold was selected such that the F1-score was maximized, i.e. retaining as many atufted neurons with natural termination, while removing as many neurons with missing segments as possible. The threshold was computed on the 120 validated neurons. This process excluded 557 neurons from layer 4.

For layer 5, we inspected 176 neurons with early-terminating apical dendrites. Of those, 59 showed a natural apical termination, while 117 had reconstruction issues or left the volume. We found no clear quantitative metric like the depth of the apical to exclude neurons with unnatural terminations. Therefore, we excluded neurons based on their cluster membership from further analysis if the cluster contained more than 50% of neurons with unnatural terminations. Of the 15 clusters, we excluded four, corresponding to 1,258 out of 5,858 L5 neurons.

For layer 6, we inspected 183 neurons with early terminating apicals. Of those, 100 showed a natural apical termination, while 83 had reconstruction issues or left the volume. Due to the slant of the volume, long, narrow L6 cells near the volume boundary had a high likelihood of leaving the boundary with their apical dendrite. Therefore, we excluded all L6 neurons whose apical dendrite leaves the volume ( $n = 867$ ) prior to our analysis. We considered a neuron as leaving the volume if the most superficial point of its apical tree was within a few micrometers of the volume boundary.

Overall, we excluded 2,684 neurons as a result of this manual validation step, resulting in a final sample size of 31,313 neurons used in our analysis (Figs. 4+5).

#### 4.10 Cortical area boundaries

Cortical area boundaries were manually drawn from retinotopic maps of visual cortex taken before EM imaging. For further details see companion paper [2].

#### 4.11 Dimensionality reduction

For visualization of the learned embeddings, we reduced the dimensionality of the 32d embedding vector to 2d using t-distributed stochastic neighbor embedding (t-SNE; [24]) using the openTSNE package [18] with cosine distance and a perplexity of 30 for t-SNE plots for individual cortical layers and a perplexity of 300 for the whole dataset.

#### 4.12 Morphometric descriptors

We computed morphometrics based on the neuronal skeletons for analysis of the learned latent space. Morphometrics are not used for learning the morphological vector embeddings. We computed morphometrics based on compartment labels: soma, apical dendrites, basal dendrites and oblique dendrites (Sec. 4.2). They are visualized in Fig. 3. TOTAL APICAL LENGTH is defined as the total length of all segments of the skeletons that are classified as apical dendrites. TOTAL BASAL LENGTH

is computed analogously. DEPTH refers to the depth of the soma centroid relative to the pia after volume normalization (Sec. 4.3), where pia depth is equal to zero. HEIGHT is the absolute difference between the highest and the lowest skeleton node of a neuron in y-direction. APICAL WIDTH refers to the widest extent of apical dendrites in the xz-plane. BASAL BIAS describes the difference between the soma depth and the center of mass of the basal dendrites along the y-axis.

### 4.13 Statistics

Apical lengths in Sec. 2.7 were compared between V1 and HVA per laminar layer with four independent two-tailed Student's t-tests. The single-test significance level of 0.01 was corrected for multiple tests using Bonferroni correction to 0.0025. Only neurons that had any nodes labeled as apical were taken into account for this analysis. In L2/3,  $n = 6,760$  neurons were taken into account from V1 and  $n = 3,436$  from HVA; for L4  $n = 5,217$  (V1) and  $n = 2,534$  (HVA); for L5  $n = 3,708$  (V1) and  $n = 1,924$  (HVA); and for L6  $n = 3,959$  (V1) and  $n = 2,618$  (HVA).

### Data availability

Data for this paper was analyzed at materialization version 374. Data is publicly available via <https://www.microns-explorer.org/cortical-mm3> and will be updated closer to publication.

### Code availability

The code for GRAPHDINO is available at <https://eckerlab.org/code/weis2021b/>. Analysis code will be made available on the Eckerlab Github repository (forthcoming). Analyses were performed in Python 3.10 using custom code and the libraries Matplotlib362, Numpy124, openTSNE062, Pandas152, Pytorch113, Scikit-learn120, Scipy110, and Seaborn012 for general computation, machine learning and data visualization.

### Author Contributions

We use the CRediT system for author roles. Conceptualization: ASE, MAW. Methodology: MAW, ASE. Software: MAW, SP, TL. Validation: MAW, SP. Formal analysis: MAW, SP, LH. Investigation: MAW, SP, BC. Resources: BC, PGF, JAB, ALB, DB, JB, DJB, MAC, FC, NMdC, SD, LE, AH, ZI, CJ, DK, NK, SK, KiL, KaL, RL, TM, GM, EM, SSM, SM, BN, SP, RCR, CMSM, HSS, WS, MT, RT, NLT, WW, JW, WY, SY. Data curation: MAW, SP, BC. Writing - Original draft: MAW, SP, ASE. Writing - Review & editing: MAW, SP, LH, TL, AST, ASE. Visualization: MAW, SP, TL. Supervision: ASE, AST, JR. Project administration: ASE. Funding acquisition: ASE, AST, JR.

### Acknowledgements

M.A.W. was supported by the International Max Planck Research School for Intelligent Systems (IMPRS-IS), Tübingen. A.S.E. received funding for this project from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101041669). The authors thank David Markowitz, the IARPA MICrONS Program Manager, who coordinated this work during all three phases of the MICrONS program. We thank IARPA program managers Jacob Vogelstein and David Markowitz for co-developing the MICrONS program. We thank Jennifer Wang, IARPA SETA for her assistance. The work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003, D16PC00004, and D16PC00005. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government. We also thank the Allen Institute for Brain Science founder, Paul G. Allen, for his vision, encouragement and support. This work was also supported by the National Institute of Mental Health under Award Numbers R01 MH109556, P30EY002520 and the NSF NeuroNex program through grant NSF-1707400. A.S.T. acknowledges support from National Institute of Mental Health and National Institute of Neurological Disorders And Stroke under Award Number U19MH114830 and National Eye Institute award numbers R01 EY026927 and Core Grant for Vision Research T32-EY-002520-37.

### Conflict of Interest

A.S.T is cofounder of Vathes Inc., and UploadAI LLC companies in which he has financial interests. J.R. is co founder of Vathes Inc., and UploadAI LLC companies in which he has financial interests. A.S.E. is cofounder of Maddox AI GmbH, in which he has financial interests. TM and HSS disclose financial interests in Zetta AI LLC.

### References

- [1] Brendan Celii, Stelios Papadopoulos, Zhuokun Ding, Paul G. Fahey, Eric Wang, Christos Papadopoulos, Alexander Kunin, Saumil Patel, J. Alexander Bae, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger,

- Manuel A. Castro, Erick Cobos, Sven Dorkenwald, Leila Elabbady, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Sergiy Popovych, Casey M. Schneider-Mizell, William Silversmith, Marc Takeno, Russel Torres, Nicholas L. Turner, William Wong, Jingpeng Wu, Szi-chieh Yu, Wenjing Yin, Daniel Xenes, Lindsey M. Kitchell, Patricia K. Rivlin, Victoria A. Rose, Caitlyn A. Bishop, Brock Wester, Emmanouil Froudarakis, Edgar Y. Walker, Fabian H. Sinz, H. Sebastian Seung, Forrest Collman, Nuno Maçarico da Costa, R. Clay Reid, Xaq Pitkow, Andreas S. Tolias, and Jacob Reimer. Neurd: A mesh decomposition framework for automated proofreading and morphological analysis of neuronal em reconstructions. *bioRxiv*, 2023. doi: 10.1101/2023.03.14.532674.
- [2] MICrONS Consortium, J. Alexander Bae, Mahaly Baptiste, Agnes L. Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Brendan Celii, Erick Cobos, Forrest Collman, Nuno Maçarico da Costa, Sven Dorkenwald, Leila Elabbady, Paul G. Fahey, Tim Fliss, Emmanouil Froudarakis, Jay Gager, Clare Gamlin, Akhilesh Halageri, James Hebditch, Zhen Jia, Chris Jordan, Daniel Kapner, Nico Kemnitz, Sam Kinn, Selden Koolman, Kai Kuehner, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Gayathri Mahalingam, Sarah McReynolds, Elanine Miranda, Eric Mitchell, Shanka Subhra Mondal, Merlin Moore, Shang Mu, Taliah Muhammad, Barak Nehoran, Oluwaseun Ogedengbe, Christos Papadopoulos, Stelios Papadopoulos, Saamil Patel, Xaq Pitkow, Sergiy Popovych, Anthony Ramos, R. Clay Reid, Jacob Reimer, Casey M. Schneider-Mizell, H. Sebastian Seung, Ben Silverman, William Silversmith, Amy Sterling, Fabian H. Sinz, Cameron L. Smith, Shelby Suckow, Marc Takeno, Zheng H. Tan, Andreas S. Tolias, Russel Torres, Nicholas L. Turner, Edgar Y. Walker, Tianyu Wang, Grace Williams, Sarah Williams, Kyle Willie, Ryan Willie, William Wong, Jingpeng Wu, Chris Xu, Runzhe Yang, Dimitri Yatsenko, Fei Ye, Wenjing Yin, and Szi-chieh Yu. Functional connectomics spanning multiple areas of mouse visual cortex. *bioRxiv*, 2021. doi: 10.1101/2021.07.28.454025.
- [3] Javier DeFelipe, Pedro L. López-Cruz, Ruth Benavides-Piccione, Concha Bielza, Pedro Larrañaga, Stewart Anderson, Andreas Burkhalter, Bruno Cauli, Alfonso Fairén, Dirk Feldmeyer, et al. New insights into the classification and nomenclature of cortical GABAergic interneurons. *Nature Reviews Neuroscience*, 14(3):202–216, 2013.
- [4] Leila Elabbady, Sharmishta Seshamani, Shang Mu, Gayathri Mahalingam, Casey Schneider-Mizell, Agnes Bodor, J. Alexander Bae, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Manuel A. Castro, Erick Cobos, Sven Dorkenwald, Paul G. Fahey, Emmanouil Froudarakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Dan Kapner, Nico Kemnitz, Sam Kinn, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Barak Nehoran, Stelios Papadopoulos, Saamil Patel, Xaq Pitkow, Sergiy Popovych, Jacob Reimer, William Silversmith, Fabian H. Sinz, Marc Takeno, Russel Torres, Nicholas Turner, William Wong, Jingpeng Wu, Wenjing Yin, Szi-chieh Yu, Andreas Tolias, H. Sebastian Seung, R. Clay Reid, Nuno Maçarico Da Costa, and Forrest Collman. Quantitative census of local somatic features in mouse visual cortex. *bioRxiv*, 2022. doi: 10.1101/2022.07.20.499976.
- [5] Jeremy Maitin-Shepard et al. google/neuroglancer, 2021. URL <https://github.com/google/neuroglancer>.
- [6] Nathan Gouwens, Staci Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan Sunkin, David Feng, Costas Anastassiou, Eliza Barkan, Kris Bickley, Nicole Blesie, Thomas Braun, Krissy Brouner, Agata Budzillo, Shiella Caldejon, Tamara Casper, Dan Castelli, Peter Chong, and Christof Koch. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature Neuroscience*, 22, 2019.
- [7] Xiaolong Jiang, Shan Shen, Cathryn R. Cadwell, Philipp Berens, Fabian Sinz, Alexander S. Ecker, Saamil Patel, and Andreas S. Tolias. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science*, 350(6264):aac9462, November 2015. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aac9462.
- [8] Brian E. Kalmbach, Rebecca D. Hodge, Nikolas L. Jorstad, Scott Owen, Trygve E. Bakken, Rebecca de Frates, Anna Marie Yanny, Rachel Dalley, Lucas T. Graybuck, Tanya L. Daigle, Cristina Radaelli, Matt Mallory, Medea McGraw, Nick Dee, Philip R. Nicovich, C. Dirk Keene, Ryder P. Gwinn, Daniel L. Silbergeld, Charles Cobbs, Jeffrey G. Ojemann, Andrew L. Ko, Anoop P. Patel, Richard G. Ellenbogen, Staci A. Sorensen, Kimberly Smith, Hongkui Zeng, Bosiljka Tasic, Christof Koch, Ed S. Lein, and Jonathan T. Ting. Signature morpho-electric, transcriptomic, and dendritic properties of extratelencephalic-projecting human layer 5 neocortical pyramidal neurons. *bioRxiv*, 2020. doi: 10.1101/2020.11.02.365080.
- [9] Lida Kanari, Pawel Dlotko, Martina Scolamiero, Ran Levi, Julian C. Shillcock, Kathryn Hess, and Henry Markram. A topological representation of branching neuronal morphologies. *Neuroinformatics*, 16:3 – 13, 2017.

- [10] Lida Kanari, Srikanth Ramaswamy, Ying Shi, Sebastien Morand, Julie Meystre, Rodrigo Perin, Marwan Abdellah, Yun Wang, Kathryn Hess, and Henry Markram. Objective morphological classification of neocortical pyramidal cells. *Cerebral Cortex*, 29(4):1719–1735, 2019.
- [11] Euseok J. Kim, Ashley L. Juavinett, Espoir M. Kyubwa, Matthew W. Jacobs, and Edward M. Callaway. Three types of cortical layer 5 neurons that differ in brain-wide connectivity and function. *Neuron*, 88(6):1253–1267, 2015. ISSN 0896-6273. doi: <https://doi.org/10.1016/j.neuron.2015.11.002>.
- [12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015.
- [13] Sophie Lathunus, Dmitry Kobak, and Philipp Berens. A Systematic Evaluation of Interneuron Morphology Representations for Cell Type Discrimination. *Neuroinform*, 18(4):591–609, October 2020. ISSN 1559-0089. doi: 10.1007/s12021-020-09461-z.
- [14] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael Reimann, Marwan Abdellah, Carlos Aguado, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, Atnekeng Kahou Guy Antoine, Thomas K Berger, Ahmet Bilgili, Nenad Buncic, Athanassia Chalimourda, Giuseppe Chindemi, Jean-Denis Courcol, Fabien Delalandre, Vincent Delattre, and Felix Schürmann. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163:456–492, 2015.
- [15] Marcel Oberlaender, Christiaan P. J. de Kock, Randy M. Bruno, Alejandro Ramirez, Hanno S. Meyer, Vincent J. Derksen, Moritz Helmstaedter, and Bert Sakmann. Cell Type–Specific Three-Dimensional Structure of Thalamocortical Circuits in a Column of Rat Vibrissal Cortex. *Cerebral Cortex*, 22(10):2375–2391, 2012.
- [16] James L. O’Leary. Structure of the area striata of the cat. *Journal of Comparative Neurology*, 75(1):131–164, 1941. ISSN 1096-9861. doi: 10.1002/cne.900750107.
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.
- [18] Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. opentsne: a modular python library for t-sne dimensionality reduction and embedding. *bioRxiv*, 2019. doi: 10.1101/731877.
- [19] Santiago Ramón y Cajal. *Histologie du système nerveux de l’homme et des vertébrés*. 1911.
- [20] Federico Scala, Dmitry Kobak, Shen Shan, Yves Bernaerts, Sophie Lathunus, Cathryn Rene Cadwell, Leonard Hartmanis, Emmanouil Froudarakis, Jesus Ramon Castro, Zheng Huan Tan, et al. Layer 4 of mouse neocortex differs in cell types and circuit organization between sensory areas. *Nature communications*, 10(1):4174, 2019.
- [21] Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn Cadwell, Jesus Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Lathunus, Elanine Miranda, Shalaka Mulherkar, Zheng Tan, Zizhen Yao, Hongkui Zeng, Rickard Sandberg, Philipp Berens, and Andreas Tolias. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598:1–7, 2021.
- [22] Casey M Schneider-Mizell, Agnes Bodor, Derrick Brittain, JoAnn Buchanan, Daniel J. Bumbarger, Leila Elabbady, Daniel Kapner, Sam Kinn, Gayathri Mahalingam, Sharmishta Seshamani, Shelby Suckow, Marc Takeno, Russel Torres, Wenjing Yin, Sven Dorkenwald, J. Alexander Bae, Manuel A. Castro, Paul G. Fahey, Emmanouil Froudakis, Akhilesh Halageri, Zhen Jia, Chris Jordan, Nico Kemnitz, Kisuk Lee, Kai Li, Ran Lu, Thomas Macrina, Eric Mitchell, Shanka Subhra Mondal, Shang Mu, Barak Nehoran, Stelios Papadopoulos, Saumil Patel, Xaq Pitkow, Sergiy Popovych, William Silversmith, Fabian H. Sinz, Nicholas L. Turner, William Wong, Jingpeng Wu, Szi-chieh Yu, , Jacob Reimer, Andreas S. Tolias, H Sebastian Seung, R Clay Reid, Forrest Collman, and Nuno Maçarico da Costa. Cell-type-specific inhibitory circuitry from a connectomic census of mouse visual cortex. *bioRxiv*, 2023. doi: 10.1101/2023.01.23.525290.
- [23] Ruggero Scorcioni, Sridevi Polavaram, and Giorgio A Ascoli. L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature protocols*, 3(5):866–876, 2008.
- [24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(86):2579–2605, 2008.

- [25] Simon Weiler, Drago Guggiana Nilo, Tobias Bonhoeffer, Mark Hübener, Tobias Rose, and Volker Scheuss. Orientation and direction tuning align with dendritic morphology and spatial connectivity in mouse visual cortex. *Current Biology*, 32(8):1743–1753.e7, April 2022. ISSN 09609822. doi: 10.1016/j.cub.2022.02.048.
- [26] Marissa A. Weis, Laura Pede, Timo Lüddecke, and Alexander S. Ecker. Self-supervised representation learning of neuronal morphologies, 2021.
- [27] Ilker O. Yaz and Sébastien Lorient. Triangulated surface mesh segmentation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.1 edition, 2022. URL <https://doc.cgal.org/5.5.1/Manual/packages.html#PkgSurfaceMeshSegmentation>.

## A Appendix

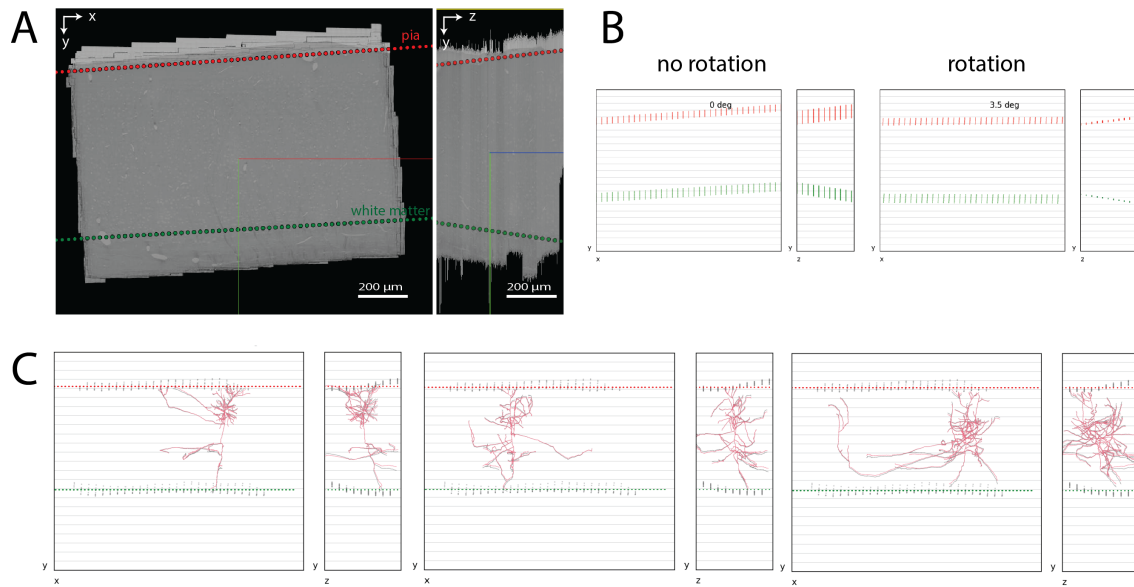


Figure A.1: Volume Pre-processing. A. x-y (left) and y-z (right) 2D cross-sectional views of the EM volume as seen in Neuroglancer. Red scatter points - linear model of pia, Green scatter points - linear model of L6 - white matter boundary. B. Pia and white matter boundary models shown with (right) and without (left) rotating the volume by 3.5 degrees about the z-axis. C. Three example excitatory neuronal skeletons shown from two 2D projections (x-y) and (y-z) after rotation and depth normalization to the mean pia and white matter depths. Red scatter points - pia model after normalization. Green scatter points - white matter boundary after normalization. Gray shadow - pia, white matter and neuronal skeleton after rotation but before normalization.

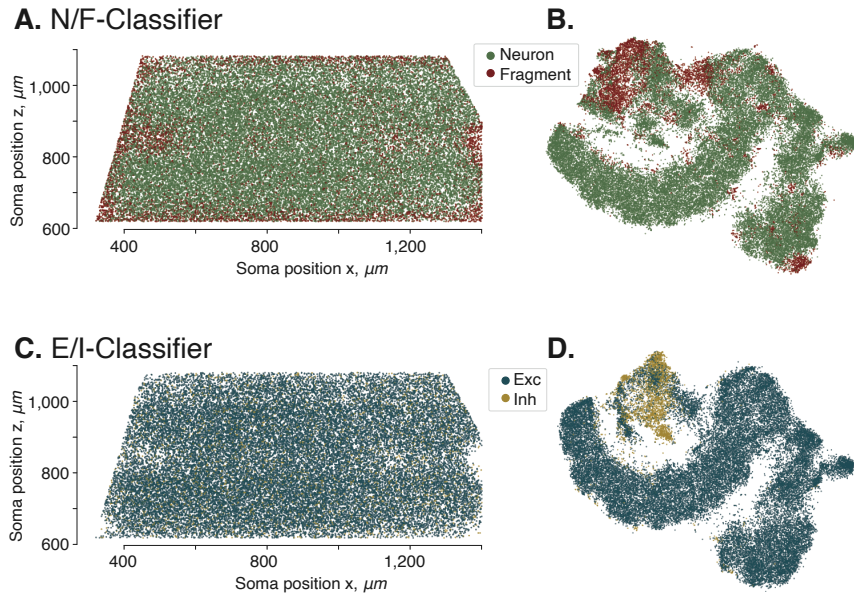


Figure A.2: **A.** Top view of the volume showing the distribution of complete neuron versus fragments predictions by our classifier based on the learned morphological embeddings. Density of fragmented neurons is high at the volume borders, since neurons have a high likelihood of leaving the imaged volume with their dendrites. Additionally, we see a high number of fragmented neurons in areas where we know there have been issues during the imaging process, proving that the classifier works as intended. **B.** t-SNE embedding of neuronal morphologies colored by neuron-fragment predictions. **C.** Top view of the volume showing a uniform distribution of excitatory versus inhibitory neurons across the volume. **D.** t-SNE embedding of neuronal morphologies colored by excitatory-inhibitory predictions.

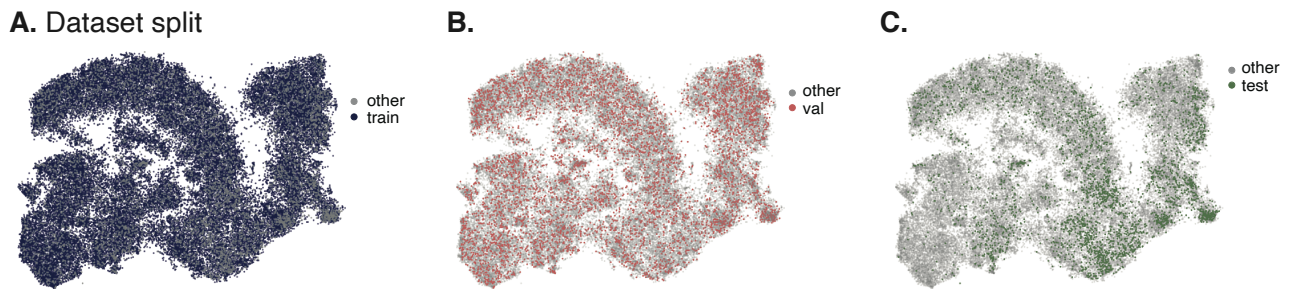


Figure A.3: t-SNE embedding of neuronal morphologies colored by the dataset split: **A.** training, **B.** validation and **C.** test set as used for GRAPHDINO training.





# Acknowledgments

I would like to acknowledge the many people who have accompanied and supported me during my doctoral studies.

First and foremost, I would like to thank my supervisors Alexander Ecker and Matthias Bethge for their support and advice throughout my PhD. Thank you for creating a great research environment and fostering an open and collaborative lab culture.

I owe special thanks to Alexander Ecker, who took me on as a student and gave me the opportunity to work under his supervision. I have learned a great deal from him over the years and I am grateful for his guidance, but also for the trust and freedom he has given me throughout my PhD.

I would like to thank all my co-authors. The publications described in this dissertation are the result of a collaborative effort involving many great people. My special thanks go to Stelios, Laura and Timo, without whom the publications would not have been possible.

Thanks to my fellow group members of both Ecker and Bethge Lab as well as my colleagues from Sinz Lab for engaging scientific and non-scientific discussions, for entertaining coffee breaks and sociable evenings outside of the lab. COVID times made it abundantly clear how valuable it is to have an amazing team of people around for inspiration and support.

I appreciate the support of Leila Masri and my thesis advisory committee: Andreas Geiger, Alexander Ecker and Matthias Bethge. Thank you for your time and input. Furthermore, I am grateful to Philipp Berens for taking on the role of my second reviewer. I owe special thanks to Heike and Marita for helping me navigate the University administration.

I would like to thank my friends in Tübingen, in Göttingen and from around the world for their support, welcome distractions and for always being there for me.

Last but not least, I am deeply grateful to my parents for their unwavering support and their continuous encouragement in all my endeavors.