Manifolds and Modules: How Function Develops in a Neural Foundation Model

Johannes Bertram
University of Tübingen
johannes.bertram@student.uni-tuebingen.de

Luciano Dyballa
School of Science & Technology
IE University

T. Anderson Keller
The Kempner Institute for Natural and Artificial Intelligence
Harvard University

Savik Kinger
Department of Computer Science
Yale University

Steven W. Zucker
Depts. of Computer Science and Biomedical Engineering
Wu Tsai Institute
Yale University

Abstract

Foundation models have shown remarkable success in fitting biological visual systems; however, their black-box nature inherently limits their utility for understanding brain function. Here, we peek inside a SOTA foundation model of neural activity (Wang et al., 2025) as a physiologist might, characterizing each 'neuron' based on its temporal response properties to parametric stimuli. We analyze how different stimuli are represented in neural activity space by building *decoding manifolds*, and we analyze how different neurons are represented in stimulus-response space by building *neural encoding manifolds*. We find that the different processing stages of the model (i.e., the feedforward *encoder*, *recurrent*, and *readout* modules) each exhibit qualitatively different representational structures in these manifolds. The *recurrent* module shows a jump in capabilities over the *encoder* module by "pushing apart" the representations of different temporal stimulus patterns; while the *readout* module achieves biological fidelity by using numerous specialized feature maps rather than biologically plausible mechanisms. Overall, we present this work as a study of the inner workings of a prominent neural foundation model, gaining insights into the biological relevance of its internals through the novel analysis of its neurons' joint temporal response patterns.

1 Introduction

Deep neural network models are powerful tools for modeling the mouse visual system by learning to predict neural responses directly from visual input (Cowley et al., 2023; Ustyuzhaninov et al., 2022; Huang et al., 2023). While much prior work has explored different computational models (Averbeck et al., 2006; Ustyuzhaninov et al., 2022; Qazi et al., 2025), foundation models are becoming extremely valuable: they not only fit neural activity at the unit level but are capable of generalizing beyond training data (Wang et al., 2025; Li et al., 2023). Nevertheless, their complexity can overwhelm understanding of the learned input-output map. Control theory teaches us that, without a perfect model, one must "look inside the box" to achieve identifiability (cf. (Åström, 2012)). Without this, we cannot guarantee correct, robust, and generalizable behavior, especially on out-of-distribution data, to

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Data on the Brain Mind.

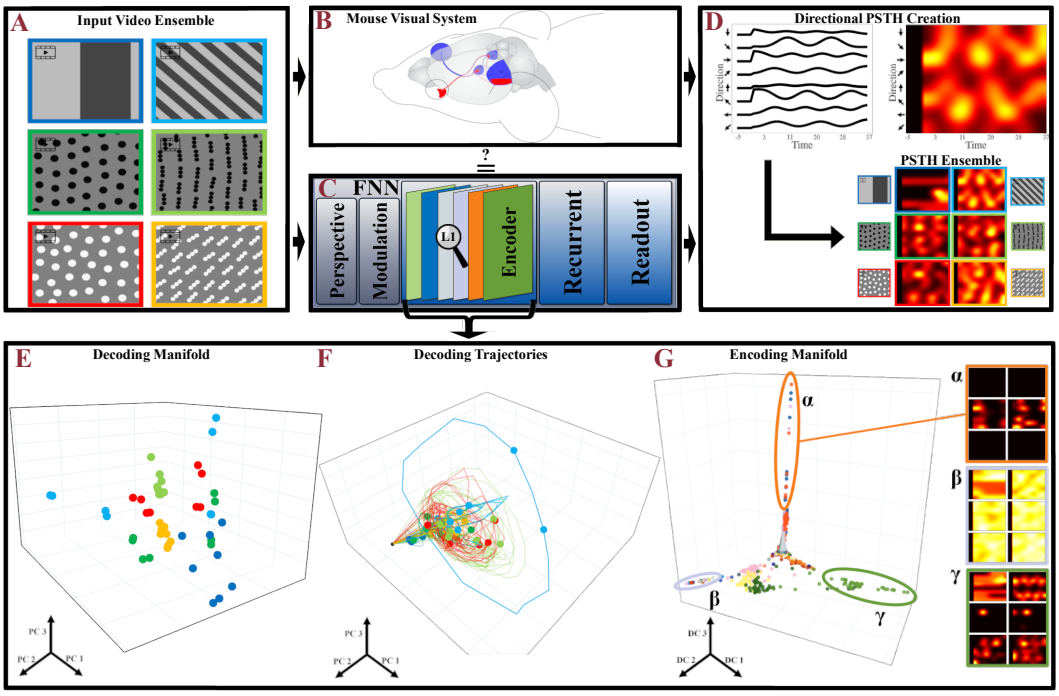

Figure 1: **Set up and techniques. A**: The stimuli consist of drifting gratings (at two spatial frequencies and 8 directions of motion) plus dotted and oriented flows, at two contrasts, drifting in 8 directions. **B, C**: Stimuli exercise both the mouse visual system (data from published literature, image from Wilks et al. (2013)) and the trained FNN (this paper) to yield activity (firing rate) in time for each direction. **D**: The firing rates are collected as a PeriStimulus Time Histogram (PSTH), denoted as a heatmap image (higher firing rate is brighter). **E**: Neural decoding manifold (each point is a trial; coordinates are PCA-reduced neural firings); colors for each trial point match the boxes around stimuli in **A**. While the trials are weakly clustered by stimulus, the representations do not allow for clear classification at this stage. **F**: Decoding trajectories show development of neural activity over time for each stimulus, also in PCA coordinates. As expected in the early stage feedforward *encoder*, neural activity barely changes after the onset of activity compared to the 0-activity point (black). Only circular temporal developments are observable for periodic input stimuli, such as moving gratings (light blue). **G**: Neural encoding manifold, in which each point is a neuron, in diffusion coordinates. Average PSTHs for neurons in circled clusters show average activity for each of the stimulus classes (arranged as in **A**). Note the multi-selectivity of neurons to different stimulus classes and especially the "amplification" induced by neurons in cluster β. We study the *encoder* (shown above), *recurrent* and *readout* modules, and ask whether they have analogues in the mouse visual system.

confidently build hypotheses about the brain using these models. Thus, we study the state-of-the-art foundation model, the FNN (Wang et al., 2025), as a (computational) neuroscientist might. The FNN consists of multiple stages (Figure 1C) and was trained on the MICrONS functional connectomics dataset (Bae et al., 2025) using artificial and 'natural' videos across multiple animals. We use modeling tools available online (references in Methods), stimuli similar to those used in FNN's original training (Wang et al., 2025), and add naturalistic flow stimuli used in mouse physiology (Dyballa et al., 2018). The last of these allows us to examine out-of-distribution performance (Figure 1A).

Prior work has explored how artificial models represent neural responses (Averbeck et al., 2006; Ustyuzhaninov et al., 2022; Qazi et al., 2025), and has examined deep neural networks with regard to functionality; some are supportive (Kriegeskorte, 2015; Yamins et al., 2014; Margalit et al., 2024), while some raise questions (Serre, 2019), in many different species. The FNN is a data-driven predictive model with a Gaussian readout (Klindt et al., 2018; Turishcheva et al., 2024; Nellen et al., 2025; Lurz et al., 2021) that interprets the output as per-neuron basis functions with individual readout weights. The readout provides an encoding embedding of biological neurons. For comparability, we use our encoding method to compare the embeddings of biological neurons and individual readout neu-

rons, investigating both the final and the readout embeddings. Different loss functions have been used for fitting mouse models (Nayebi et al., 2023; Bakhtiari et al., 2021; Shi et al., 2022), and others have studied decoding manifolds for mouse (Froudarakis et al., 2020; Beshkov and Tiesinga, 2022; Beshkov et al., 2024), focusing on topological properties. For a recent general review, see (Doerig et al., 2023).

To understand how the FNN performs, we analyze the internal representations at each processing stage (Figure 1) using three techniques popular in neuroscience. (1) We build *neural decoding manifolds* (Chung and Abbott, 2021), in which trials are embedded in the space of neural activity coordinates (Figure 1E), then dimensionality-reduced using PCA (Cunningham and Yu, 2014). Typically, trials involving the same stimulus cluster together, facilitating a read-out of the brain's state. (2) To switch from trials to neurons, we build *neural encoding manifolds* (Figure 1G) (Dyballa et al., 2024a) in which each point is a neuron in the space of stimulus-response coordinates, dimensionality-reduced using tensor factorization (Williams et al., 2018). Proximity between neurons in an encoding manifold denotes similar responses to similar stimuli; i.e., groupings of neurons that are likely to share circuit properties. Finally, (3) the relationship between these two manifolds is captured by the temporal evolution of each neuron's activity for each stimulus trial. We note the popular view that a 'neural computation' can be viewed as the result of a dynamical system in neural state space Hopfield (1984). We plot these both as PSTHs (Fig. 1.D.) and as streamline traces (decoding trajectories, Fig. 1.F.). While streamline representations have been used previously for decision tasks (Duncker and Sahani, 2021) and the motor system (Churchland et al., 2012; Safaie et al., 2023), we note: (i) the activity integral along such *decoding trajectories* (Figure 1F) defines the decoding manifold, while (ii) shared tubular neighborhoods specify position in the encoding manifold.These three perspectives allow us to examine distinct aspects of alignment: (1) Decoding manifolds indicate whether the model preserves stimulus separability observed in biological systems; (2) Encoding manifolds assess whether the functional topology of neurons resembles that of the brain; (3) Trajectories evaluate whether the model executes computations using dynamics similar to those in the brain. Importantly, a model may demonstrate alignment at one level while failing at others. To our knowledge, this is the first time all three of these techniques have been utilized together for analysis of a perceptual system; i.e., toward interpretability for a foundation model. For a review of classical encoding/decoding in neuroscience, see (Mathis et al., 2024).

With this framework, we ask: *Do neural decoding and encoding manifolds reveal new insights into how foundation models represent temporal response patterns? Are the representations brain-like?* We hypothesize that each processing stage contributes distinct representational capabilities, all essential for fitting neural data. In particular, one might expect the *recurrent* module to enrich the temporal structure of representations, analogous to cortex, and the encoder layers to resemble a retina with relatively limited recurrence.

2 Methods

Our work makes novel use of publicly available open-source resources. Specifically, we employed the pretrained foundation model of neural activity (denoted FNN) provided by Wang et al. (2025), available here; and the stimulus generation tools and neural encoding manifold construction pipeline introduced by Dyballa et al. (2024a), accessible here. Below we briefly outline our methods, and refer readers to Appendix A for the full details.

Model: The FNN consists of five modules: perspective, modulation, *encoder*, *recurrent*, and *readout*. The perspective and modulation modules model the mouse's state and transform the inputs to approximate the actual visual information received. Thus, only the *encoder*, *recurrent*, and *readout* modules perform the core computation and are the focus of this work. The *encoder* module is a 10-layer DenseNet-style convolutional encoder. Notably, it includes 3D convolutions, which in principle enable the encoder to capture temporal patterns for up to 12 time steps. The *recurrent* module is preceded by an attention layer and consists of a convolutional LSTM, followed by a single convolutional layer that produces its output. This feedforward–recurrent combination constitutes the core of the FNN, which is trained on data from all mice in combination. Finally, a separate *readout* module is trained on each mouse individually: it performs an interpolation on the recurrent output followed by a linear transformation to produce the FNN output.

Stimuli: Our stimulus set is composed of drifting square-wave gratings and optical flows moving in eight directions. The flow stimuli include oriented (lines) and non-oriented (dots) stimuli with spatial frequencies between 0.04 and 0.5 $\frac{\text{cycles}}{\text{deg}}$. This yields 88 unique input sequences with stochastic initial positions and velocities. The stimuli are scaled and cropped to match the FNN requirements. A subset of these stimuli are visualized in Figure 1A. To ensure that these stimuli would drive the network in a representative manner, we compared the output of the network for these stimuli with the output for the original natural movie stimuli used to train the network (Appendix Figures 8, 9); and found the results to be quantitatively similar in all measured respects.

PSTH visualization: To visualize the network responses to many stimuli concisely, we group together the model's PeriStimulus Time Histogram responses (PSTH) corresponding to all flow directions of a single given stimulus pattern, forming the black-yellow heatmaps of spike rate, with time on the x-axis and flow direction on the y-axis, depicted in Figure 1D. We then organize these joint PSTH plots according to 6 of the unique stimulus patterns in a 2×3 grid.

Decoding manifolds & Trajectories: Following traditional analysis techniques, we first constructed decoding manifolds by performing PCA on the stimulus-time-averaged activity data. In total therefore, the decoding manifold contains 48 points, one for each unique sequence, and colored by the corresponding base-stimulus (shown in Figure 1). Different spatial frequencies of the same stimulus are summarized with the same color. To construct *decoding trajectories*, we treat each time step as a separate data point rather than averaging across time before applying PCA. We compare with biological *decoding trajectories* using the experimental data from Dyballa et al. (2024a). For additional quantitative analysis of these trajectories, we compute a novel 'clustering' of bundles of trajectories, which we term *tubularity*, yielding values for 'Tightness' and 'Crossings', depicted in Figure 3 (definition in Appendix A.9.1).

Encoding manifolds: To understand the response properties of *neurons* with respect to all stimuli (rather than the representation of *stimuli* in the space of all neurons), we finally construct *encoding manifolds*. At a high level, these manifolds allow one to examine the global topology of neuronal populations based on their stimulus selectivities and temporal response patterns (Dyballa et al., 2024a). The neural encoding manifold is constructed in a three-step procedure. First, a 3-tensor is built with the temporal responses from each neuron for each stimulus, and decomposed using Nonnegative Tensor Factorization (details in Appendix); each component is comprised of neural, stimulus, and temporal response factors. The neural factors then serve as position coordinates, embedding the neurons into a stimulus-response framework called the neural encoding space. Second, we construct a data graph in this neural encoding space using the IAN algorithm (Dyballa and Zucker, 2023). Third, applying diffusion maps (Coifman et al., 2005; Coifman and Lafon, 2006) to the data graph yields the manifold. We follow the methodological choices of Dyballa et al. (2024a), where extensive parameter analysis for biological neural data was conducted.

3 Results

We built encoding manifolds, as well as decoding trajectories and manifolds, for all layers (of considered modules) in the FNN. Here we highlight the results most helpful toward interpreting the computational role of each stage of the FNN network. Beginning with the first encoder layer (L1), we found that its decoding manifold was poorly clustered (Figure 1E), with the different stimulus classes quite mixed. This implies that, at this point within the FNN, the latent feature representation is not sufficient to distinguish between the different stimuli (indeed, its classification accuracy is lowest; see Table 1). The decoding trajectories for L1, however, reveal a more complete picture: from Figure 1F we note that periodic stimuli are represented as loops, likely due to the translation

Table 1: Stimulus classification accuracy for Leave-One-Out 3-Nearest Neighbor (3-NN) and Logistic Regression (LR) classifiers trained on each layer's activations. Methods' details in Appendix A.

Accuracy	Enc1	Enc2	Enc4	Enc5	Enc7	Enc8	Rec	RecOut	Readout	Out
LR	0.59	0.62	0.66	0.65	0.71	0.74	0.89	**0.90**	0.88	0.77
3-NN	0.41	0.66	0.58	0.52	0.53	0.61	**0.73**	0.64	0.63	0.67

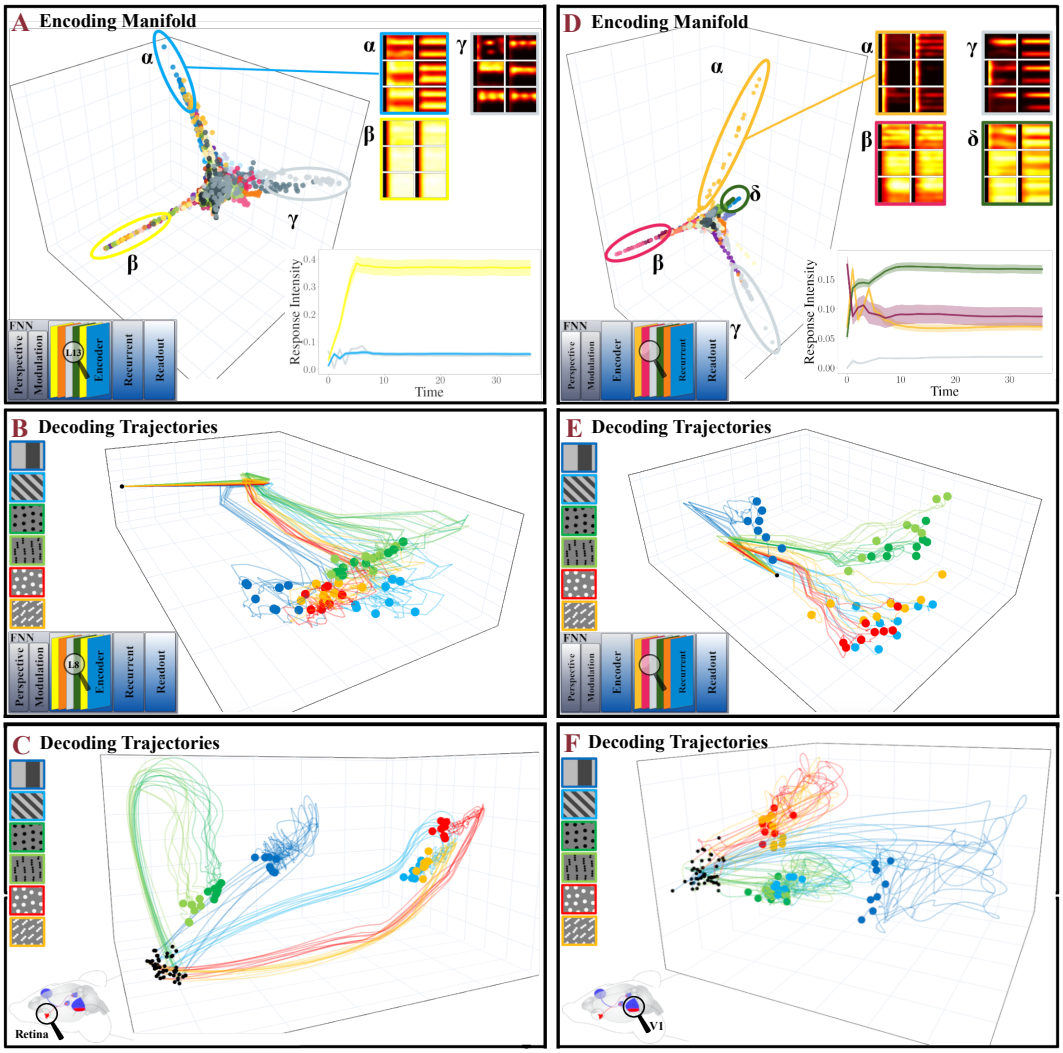

Figure 2: **Encoding and recurrent layers**. **A**: Encoding manifold for final encoding block. PSTHs for arm β amplify all stimulus signals; inset shows mean response intensity development ± 1 s.e.m. within units in β (yellow) compared with others. **B**: Explosive growth of trajectories for FNN is caused by initial intensity increase in β. Ensuing temporal dynamics are negligible. This differs from the trajectory bundles found in mouse retina (**C**), showing stimulus dependence instead of nonselective intensity induced temporal patterns. **D**: Recurrent hidden state shows multi-selectivity of units and no explosive intensity growth (cf. inset). **E**: Decoding trajectories show increased stimulus-dependent temporal patterns leading to better discriminability of stimuli in PCA space. However, trajectories are more temporally monotonic than in primary visual cortex (**F**).

equivariance of the convolutional layers of the encoder preserving the circular geometric structure of these stimulus sequences (Cohen and Welling, 2016). However, we see that these loops can take on many different forms (such as the high spatial frequency gratings, shown in light blue), defined by the responses of individual neurons to each stimulus. Finally, the encoding manifold for L1 (Figure 1G) completes the characterization by revealing that most neurons belonging to the same feature map (points with the same color label) form contiguous clusters, or regions, over the manifold; this is not entirely surprising given the weight-sharing property of these convolutional layers. Nevertheless, several feature maps are found mixed into the same "arm" (labeled β). Examining the response profile (PSTH ensemble) of these neurons in detail, we notice strong, continuous activity throughout the trial duration to all stimulus classes.

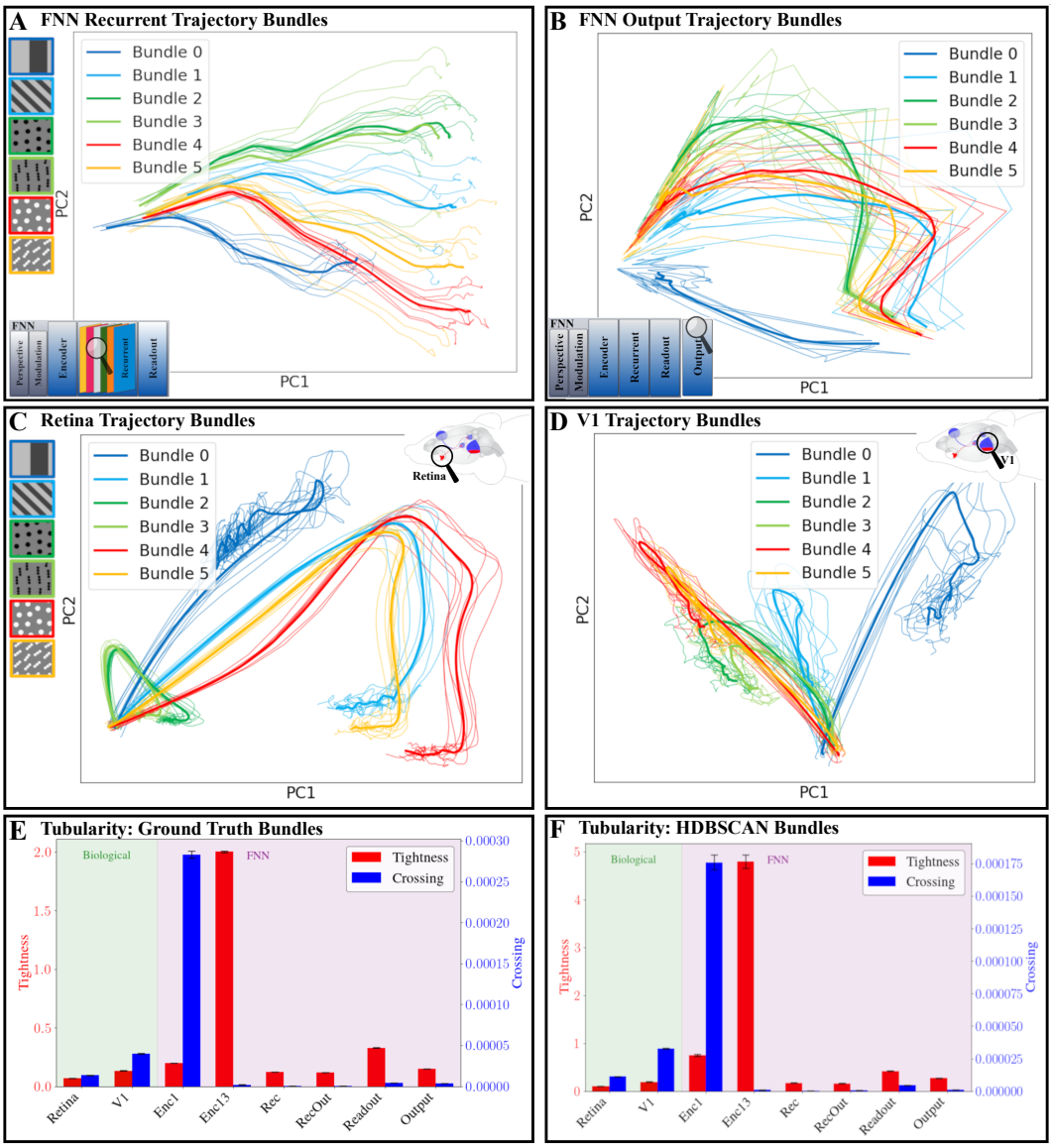

Figure 3: **Tubularity of *recurrent* module and *output* compared to mouse data.** Similar decoding trajectories can be clustered into bundles and averaged. Here we cluster by stimulus class; the mean contour is displayed in the stimulus-class color for **A.** the recurrent module (note class separation developing) and **B.** the output layer (note the 'circular' dynamics). These differ from biological trajectories for **C.** mouse retina and **D.** V1. These differences are quantified by the tightness and crossing measures (Appendix A.9.1) for both ground truth and unsupervised (HDBSCAN) groupings.

We now move on to the late-stage encoder, layer 8 (L8). First, although its encoding manifold shows that the grouping by feature maps is still apparent, especially in the right-hand side of the manifold (Figure 2A), the overall manifold appears less clustered and more mixed. On the other hand, again we find a poorly-selective "intensity arm" of neurons (β) from multiple feature maps representing a strong response to all stimuli. This is supported by plotting the mean activity of neuron groups (inset in Figure 2A). The marked increase in response magnitude early in the trial among units in the β arm can be readily noted in the decoding trajectories' visualization (Figure 2B). Further investigation revealed that the intensity arm arises from padding artifacts at the edges of feature maps. Sampling from the feature maps' central regions eliminates the intensity arm and the shared activity development in the decoding trajectories (see Supplemental Figure 10). However, these artifacts impact the representation, as the smoothness of the intensity arm visualizes the spread of

the information of the intensity artifacts across the feature maps. Since these artifacts are present during normal network operation, excluding them would misrepresent the model's actual internal dynamics. We therefore retain these artifacts in our manifold analysis. The L8's decoding manifold was qualitatively similar to Layer 1's (not shown).

How do these findings for the encoder stage (L1 and L8) compare to the retina, the first stage in the mouse visual system Baden et al. (2016)? We applied the same procedure to analyze the physiological data from Dyballa et al. (2024a). The non-selective groups of neurons with high activity (β arms in Figs. 1E and 2A) are the first departure from what is found in biological networks: in the retina there are no such non-selective neurons. Such low-selectivity in cortex is restricted to inhibitory (inter)neurons, and continuously mixes with other, more selective responses; they do not segregate as an arm, or cluster.Perhaps the biggest difference is that retinal decoding trajectories formed largely segregated, stimulus-dependent bundles whose temporal dynamics allowed for linear separability during much of the trial's time-course (Figure 2C,F). As will be shown, this is also in stark contrast with what was found along the encoder stage of the FNN (Figure 2B). Thus, despite temporal convolutions, the FNN feedforward encoder appears to lack biologically plausible stimulus-dependent temporal patterns and mainly reports features present in the input with varying intensity.

The *recurrent* module is qualitatively different. Its encoding manifold shows that different regions exhibit a variety of distinct selectivity and temporal response patterns, cf. their PSTHs (Figure 2D). Furthermore, although the segregation by feature map is still present, no longer do we find a cluster of neurons with no selectivity (e.g., even the highlighted β and δ groups show clear selectivity for particular directions or orientations). Moreover, this is the first stage where the FNN is capable of reasonably decoding the different stimulus classes, as revealed by the somewhat segregated bundles of decoding trajectories in Figure 2E. In fact, this is where the network reaches its highest stimulus classification accuracy (see Table 1).

While the recurrent module shows the presence of stimulus-dependent temporal patterns, the organization of decoding trajectories is noticeably more entangled than both retina and V1 (compare with Figure 2C,F). This phenomenon is quantified using *tubularity* metrics based on the geometry of the observed decoding bundles (see Method). We found that tubes have similar tightness from the recurrent stage onwards, but fall short of representing all complexity of activity development in biology as shown by significantly lower crossings ($p<0.005$, Bonferroni-corrected, for all layers) (Figure 3).

The final stages of the network—the *readout* and *output* layers—are different again. The encoding manifold for the readout layer is highly disconnected (Figure 4A), with each cluster corresponding almost exclusively to neurons sampled from a single feature map. Each feature map exhibits a distinct response pattern that is invariant across neurons within it. Compared to this, the biological results (e.g., Baden et al. (2016); Dyballa et al. (2024a)) show more variability within cell "types", even in the retina. Curiously, and despite this intra-map constancy, the large number of feature maps (see PSTHs) and the rich dynamics within each one, somehow enable the *output* to represent the complex behavior of neurons (Figure 4B). These behaviors are captured in the FNN output via a linear combination of *readout* features. Since classification accuracy has declined slightly at this stage, but orientation and direction selectivity agree (Supplemental Figure 10), we conjecture these dynamics are interpolating the spiking activity.

4 Discussion

Decoding manifolds (and trajectories) allow us to compare whether networks can achieve similar degrees of stimulus representation and separability. Encoding manifolds, on the other hand, allow us to check at a global level how the responses of individual neurons (and their global organization) compare to those of biological neurons; in other words, whether the FNN and biological networks employ similar encoding mechanisms for achieving similar outputs. Since decoding trajectories are a surrogate for "computation" as dynamics over neural state space (cf. (Hopfield, 1984), this investigation moves beyond pairwise or average unit comparisons (e.g., RSA (Kriegeskorte et al., 2008)) and may be useful in analyzing other foundation models.

Our analysis of the FNN revealed an increasing richness of representation up to the *recurrent* module (cf. Hoeller et al. (2024) and contrasting with Xu et al. (2023); Nayebi et al. (2023); Froudarakis

et al. (2020)), albeit with most PSTHs revealing a lack of typical biological-like responses Ringach et al. (2016); Ko et al. (2011). Since the FNN was trained to predict neural spike trains, classification evolved implicitly (cf. Table 1)). Thus it is likely that the recurrent features are sufficiently complex for feature representation and that the subsequent modules work toward fitting the neural data instead.

However, the highly clustered topology of the latent representation found for the *readout* module was not a good fit for retina or cortex (cf. Baden et al. (2016); Dyballa et al. (2024a), nor for higher visual areas (cf. Glickfeld and Olsen (2017); Dyballa et al. (2024b); Yu et al. (2022)). Regardless, the rich dynamics within each feature map (see PSTHs), combined with the large number of them, seem to enable the *output* layer to represent the complex behavior of neurons (Figure 4B), resulting in the network's high performance in predicting neural activity. Still, it is somewhat surprising that such behaviors are produced in the FNN output via a simple linear combination of *readout* features—one would expect that fitting the neural activity should happen throughout the entire network, and not as a separate appendage module.

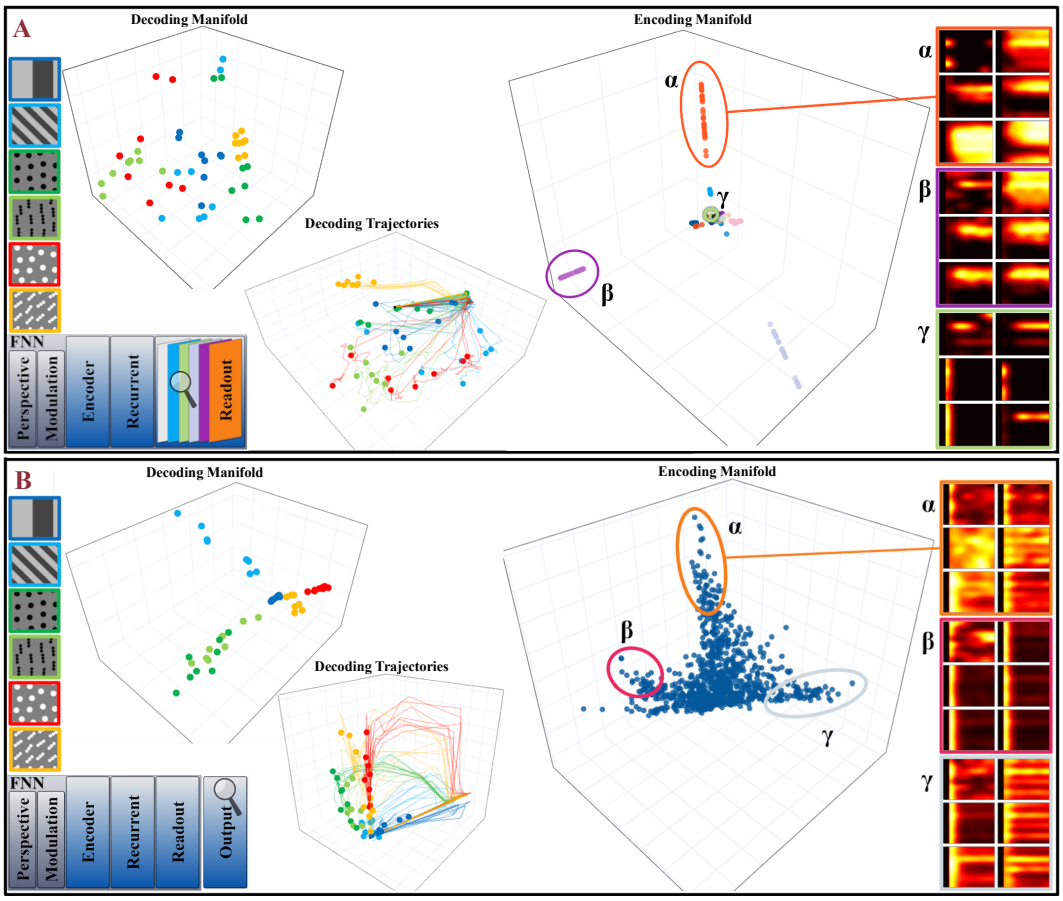

Figure 4: **Contrasting the readout (A) and output (B) layers.** While the decoding manifolds and trajectories appear qualitatively similar, the encoding manifolds have remarkably distinct topologies: while the *readout* module is highly clustered, the *output* is continuous. The clustered topology is caused by the interpolation step producing a large amount of features with low within-feature variability. The smooth output is obtained by collapsing the many feature maps to a single output by a linear combination.

Limitations Our analysis utilized a single foundation model, due to the limited availability of other video-based foundation models of neural activity over time. Moreover, we worked with a restricted set of stimuli (see Methods) to ensure comparability to biological results. Nevertheless, there is evidence that these stimuli exercise much of the mouse visual cortex Dyballa et al. (2018), so they provide at least a necessary component for out-of-sample examination. Moreover, we show that

these stimuli exercise FNN like the natural movies on which they were trained (Appendix Figure 8), empirically validating their usefulness.

5 Conclusion

We found a rich diversity of encoding and decoding topologies in the FNN, highlighting its capability to fit complex neural data. Distinct representations emerge from each module, reflecting its architecture: First, the *recurrent* module appears to learn generalizable representations of temporal stimuli, encouraging uniformity and alignment, as in general self-supervised foundation models (Wang and Isola, 2022). Second, we found that the *readout* module accounts for rich biological variability, but does so by relying on a large number of self-similar feature maps, differing from known biological counterparts in V1. Finally, the output layer is able to achieve a continuous representation by linearly combining the readout representation; this ultimately enables the network to (a posteriori) associate spike trains to the input movies.

Together, these findings imply neural foundation models may be more similar in internal operation to other foundation models, rather than their biological counterparts. While this does not alter the usefulness of neural foundation models, it suggests that future architectures incorporating recurrence directly after a more light-weight retinal-like encoder and constraining feature dimensionality to match biological cell type diversity could yield models that bridge the gap between computational performance and biological plausibility.

Acknowledgments and Disclosure of Funding

This article has received funding from the European Commission's Marie Skłodowska-Curie Action under grant agreement no. 101207931 (LD), and by the MICIU /AEI /10.13039/501100011033 / FEDER, UE Grant no. PID2024-155187OB-I00 (LD). SWZ was supported by NIH grant 1R01EY031059 and NSF Grant 1822598.

References

Karl J Åström. *Introduction to stochastic control theory*. Courier Corporation, 2012.

Bruno B. Averbeck, Peter E. Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature Reviews Neuroscience*, 7(5):358–366, May 2006. ISSN 1471-003X, 1471-0048. doi: 10.1038/nrn1888. URL https://www.nature.com/articles/nrn1888.

Tom Baden, Philipp Berens, Katrin Franke, Miroslav Román Rosón, Matthias Bethge, and Thomas Euler. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586):345, 2016.

Brett W. Bader, Tamara G. Kolda, et al. Tensor toolbox for matlab, version 3.6. https://www.tensortoolbox.org, 2023.

J. Alexander Bae, Mahaly Baptiste, Maya R. Baptiste, Caitlyn A. Bishop, Agnes L. Bodor, et al. Functional connectomics spanning multiple areas of mouse visual cortex. *Nature*, 640(8058): 435–447, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08790-w. URL https://doi.org/10.1038/s41586-025-08790-w.

Shahab Bakhtiari, Patrick Mineault, Timothy Lillicrap, Christopher Pack, and Blake Richards. The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *Advances in Neural Information Processing Systems*, 34: 25164–25178, 2021.

Kosio Beshkov and Paul Tiesinga. Geodesic-based distance reveals nonlinear topological features in neural activity from mouse visual cortex. *Biological Cybernetics*, 116(1):53–68, 2022.

Kosio Beshkov, Marianne Fyhn, Torkel Hafting, and Gaute T Einevoll. Topological structure of population activity in mouse visual cortex encodes densely sampled stimulus rotations. *Iscience*, 27(4), 2024.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37456-2.

SueYeon Chung and L. F. Abbott. Neural population geometry: An approach for understanding biological and artificial neural networks. *Current Opinion in Neurobiology*, 70:137–144, October 2021. ISSN 09594388. doi: 10.1016/j.conb.2021.10.010. URL http://arxiv.org/abs/2104.07059. arXiv:2104.07059 [q-bio].

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487 (7405):51–56, 2012.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, May 2005. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0500334102. URL https://pnas.org/doi/full/10.1073/pnas.0500334102. Publisher: Proceedings of the National Academy of Sciences.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006. ISSN 1063-5203. doi: 10.1016/j.acha.2006.04.006. URL https://linkinghub.elsevier.com/retrieve/pii/S1063520306000546. Publisher: Elsevier BV.

Benjamin R. Cowley, Patricia L. Stan, Jonathan W. Pillow, and Matthew A. Smith. Compact deep neural network models of visual cortex. *bioRxiv: The Preprint Server for Biology*, page 2023.11.22.568315, November 2023. ISSN 2692-8205. doi: 10.1101/2023.11.22.568315.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

Adrien Doerig, Rowan P Sommers, Katja Seeliger, Blake Richards, Jenann Ismael, Grace W Lindsay, Konrad P Kording, Talia Konkle, Marcel AJ Van Gerven, Nikolaus Kriegeskorte, et al. The neuroconnectionist research programme. *Nature Reviews Neuroscience*, 24(7):431–450, 2023.

Lea Duncker and Maneesh Sahani. Dynamics on the manifold: Identifying computational dynamical activity from neural population recordings. *Current Opinion in Neurobiology*, 70:163–170, October 2021. ISSN 0959-4388. doi: 10.1016/j.conb.2021.10.014. URL https://linkinghub.elsevier.com/retrieve/pii/S0959438821001264. Publisher: Elsevier BV.

Luciano Dyballa and Steven W. Zucker. IAN: Iterated Adaptive Neighborhoods for manifold learning and dimensionality estimation. *Neural Computation*, 35(3):453–524, February 2023. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco_a_01566. URL http://arxiv.org/abs/2208.09123. arXiv:2208.09123 [cs].

Luciano Dyballa, Mahmood S Hoseini, Maria C Dadarlat, Steven W Zucker, and Michael P Stryker. Flow stimuli reveal ecologically appropriate responses in mouse visual cortex. *Proc Natl Acad Sci USA*, 115(44):11304–11309, 2018.

Luciano Dyballa, Andra M. Rudzite, Mahmood S. Hoseini, Mishek Thapa, Michael P. Stryker, Greg D. Field, and Steven W. Zucker. Population encoding of stimulus features along the visual hierarchy. *Proceedings of the National Academy of Sciences*, 121(4), January 2024a. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2317773121. URL https://pnas.org/doi/10.1073/pnas.2317773121. Publisher: Proceedings of the National Academy of Sciences.

Luciano Dyballa, Greg D Field, Michael P Stryker, and Steven W Zucker. Functional organization and natural scene responses across mouse visual cortical areas revealed with encoding manifolds. *bioRxiv*, 2024b.

Emmanouil Froudarakis, Uri Cohen, Maria Diamantaki, Edgar Y Walker, Jacob Reimer, Philipp Berens, Haim Sompolinsky, and Andreas S Tolias. Object manifold geometry across the mouse cortical visual hierarchy. *BioRxiv*, pages 2020–08, 2020.

Lindsey L Glickfeld and Shawn R Olsen. Higher-order areas of the mouse visual cortex. *Annual review of vision science*, 3:251–273, 2017.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Judith Hoeller, Lin Zhong, Marius Pachitariu, and Sandro Romani. Bridging tuning and invariance with equivariant neuronal representations. *bioRxiv*, pages 2024–08, 2024.

John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088–3092, 1984.

Liwei Huang, Zhengyu Ma, Liutao Yu, Huihui Zhou, and Yonghong Tian. Deep Spiking Neural Networks with High Representation Similarity Model Visual Pathways of Macaque and Mouse. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(1):31–39, June 2023. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i1.25073. URL https://ojs.aaai.org/index.php/AAAI/article/view/25073.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Plotly Technologies Inc. Collaborative data science. https://plot.ly, 2015.

David A. Klindt, Alexander S. Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating "what" and "where", 2018. URL https://arxiv.org/abs/1711.02653.

Ho Ko, Sonja B Hofer, Bruno Pichler, Katherine A Buchanan, P Jesper Sjöström, and Thomas D Mrsic-Flogel. Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473(7345):87–91, 2011.

Nikolaus Kriegeskorte. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu Rev Vis Sci*, 1:417–446, 2015.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Nicholas Krämer et al. Tueplots, 2024. URL https://tueplots.readthedocs.io/en/latest/index.html.

Bryan M. Li, Isabel M. Cornacchia, Nathalie L. Rochefort, and Arno Onken. V1T: large-scale mouse V1 response prediction using a Vision Transformer, September 2023. URL http://arxiv.org/abs/2302.03023. arXiv:2302.03023 [cs].

Konstantin-Klemens Lurz, Mohammad Bashiri, Konstantin Willeke, Akshay Jagadish, Eric Wang, Edgar Y. Walker, Santiago A Cadena, Taliah Muhammad, Erick Cobos, Andreas S. Tolias, Alexander S Ecker, and Fabian H. Sinz. Generalization in data-driven models of primary visual cortex. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Tp7kI90Htd.

Eshed Margalit, Hyodong Lee, Dawn Finzi, James J. DiCarlo, Kalanit Grill-Spector, and Daniel L.K. Yamins. A unifying framework for functional organization in early and higher ventral visual cortex. *Neuron*, 112(14):2435–2451.e7, July 2024. ISSN 0896-6273. doi: 10.1016/j.neuron.2024.04.018. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627324002794. Publisher: Elsevier BV.

Mackenzie Weygandt Mathis, Adriana Perez Rotondo, Edward F Chang, Andreas S Tolias, and Alexander Mathis. Decoding the brain: From neural representations to mechanistic models. *Cell*, 187(21):5814–5832, 2024.

Aran Nayebi, Nathan C. L. Kong, Chengxu Zhuang, Justin L. Gardner, Anthony M. Norcia, and Daniel L. K. Yamins. Mouse visual cortex as a limited resource system that self-learns an ecologically-general representation. *PLOS Computational Biology*, 19(10):e1011506, October 2023. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1011506. URL https://dx.plos.org/10.1371/journal.pcbi.1011506.

Nina S. Nellen, Polina Turishcheva, Michaela Vystrčilová, Shashwat Sridhar, Tim Gollisch, Andreas S. Tolias, and Alexander S. Ecker. Learning to cluster neuronal function, 2025. URL https://arxiv.org/abs/2506.03293.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, et al. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL http://arxiv.org/abs/1912.01703.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ahmed Qazi, Hamd Jalil, and Asim Iqbal. Mice to Machines: Neural Representations from Visual Cortex for Domain Generalization, May 2025. URL http://arxiv.org/abs/2505.06886. arXiv:2505.06886 [cs].

Dario L Ringach, Patrick J Mineault, Elaine Tring, Nicholas D Olivas, Pablo Garcia-Junco-Clemente, and Joshua T Trachtenberg. Spatial clustering of tuning in mouse primary visual cortex. *Nat Commun*, 7, 2016.

Mostafa Safaie, Joanna C. Chang, Junchol Park, Lee E. Miller, Joshua T. Dudman, Matthew G. Perich, and Juan A. Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623(7988):765–771, 2023. doi: 10.1038/s41586-023-06714-0. URL https://doi.org/10.1038/s41586-023-06714-0.

Thomas Serre. Deep learning: the good, the bad, and the ugly. *Annual review of vision science*, 5(1): 399–426, 2019.

Jianghong Shi, Bryan Tripp, Eric Shea-Brown, Stefan Mihalas, and Michael A. Buice. MouseNet: A biologically constrained convolutional neural network model for the mouse visual cortex. *PLOS Computational Biology*, 18(9):e1010427, September 2022. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1010427. URL https://dx.plos.org/10.1371/journal.pcbi.1010427.

Polina Turishcheva, Max Burg, Fabian H. Sinz, and Alexander Ecker. Reproducibility of predictive networks for mouse visual cortex, 2024. URL https://arxiv.org/abs/2406.12625.

Ivan Ustyuzhaninov, Max F. Burg, Santiago A. Cadena, Jiakun Fu, Taliah Muhammad, Kayla Ponder, Emmanouil Froudarakis, Zhiwei Ding, Matthias Bethge, Andreas S. Tolias, and Alexander S. Ecker. Digital twin reveals combinatorial code of non-linear computations in the mouse primary visual cortex, February 2022. URL http://biorxiv.org/lookup/doi/10.1101/2022.02.10.479884.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Eric Y. Wang, Paul G. Fahey, Zhuokun Ding, Stelios Papadopoulos, Kayla Ponder, et al. Foundation model of neural activity predicts response to new stimulus types. *Nature*, 640(8058):470–477, April 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-08829-y. URL https://www.

nature.com/articles/s41586-025-08829-y. Publisher: Springer Science and Business Media LLC.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2022. URL https://arxiv.org/abs/2005.10242.

Tenelle A. Wilks, Alan R. Harvey, and Jennifer Rodger. Seeing with two eyes: Integration of binocular retinal projections in the brain. In Francesco Signorelli and Domenico Chirchiglia, editors, *Functional Brain Mapping and the Endeavor to Understand the Working Brain*, chapter 12. IntechOpen, Rijeka, 2013. doi: 10.5772/56491. URL https://doi.org/10.5772/56491.

Alex H. Williams, Tony Hyun Kim, Forea Wang, Saurabh Vyas, Stephen I. Ryu, Krishna V. Shenoy, Mark Schnitzer, Tamara G. Kolda, and Surya Ganguli. Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. *Neuron*, 98(6):1099–1115.e8, June 2018. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.05.015. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627318303878. Publisher: Elsevier BV.

Aiwen Xu, Yuchen Hou, Cristopher Niell, and Michael Beyeler. Multimodal deep learning model unveils behavioral dynamics of v1 activity in freely moving mice. *Advances in neural information processing systems*, 36:15341–15357, 2023.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc Natl Acad Sci USA*, 111(23):8619–8624, 2014.

Yiyi Yu, Jeffrey N Stirman, Christopher R Dorsett, and Spencer L Smith. Selective representations of texture and motion in mouse higher visual areas. *Current Biology*, 32(13):2810–2820, 2022.

Appendix

A Methods

A.1 Online material

Our work made use of publicly available open-source resources. Specifically, we employed the pretrained FNN model provided by Wang et al. (2025), available at `https://github.com/cajal/fnn/tree/main`. For the analysis of this model, we used the stimulus generation tools and neural encoding manifold construction pipeline introduced by Dyballa et al. (2024a), accessible at `https://github.com/dyballa/NeuralEncodingManifolds`.

A.2 FNN

The FNN consists of five modules: perspective, modulation, *encoder*, *recurrent*, and *readout*. The perspective and modulation modules model the mouse's state and transform the inputs to approximate the actual visual information received. Thus, only the *encoder*, *recurrent*, and *readout* modules perform the core computation and are the focus of this work.

The *encoder* module is a 10-layer DenseNet-style convolutional encoder. Notably, it includes 3D convolutions, which in principle enable the encoder to capture temporal patterns for up to 12 time steps. The *recurrent* module is preceded by an attention layer and consists of a convolutional LSTM, followed by a single convolutional layer that produces its output. This feedforward–recurrent combination constitutes the core of the FNN, which is trained on all data. Finally, the *readout* module is mouse-specific: it performs an interpolation on the recurrent output followed by a linear transformation to produce the FNN output. We used the FNN from session 8, scan 5. We validated results on other scans for the readout and output, and argue that our comparison at the population level is valid across animals. Moreover, the core was jointly trained on all scans.

A.3 Input videos

We used the visual stimuli from Dyballa et al. (2024a), consisting of drifting square-wave gratings and optical flows moving in eight directions. The flow stimuli include oriented (lines) and non-oriented (dots) stimuli with spatial frequencies between 0.04 and 0.5 $\frac{\text{cycles}}{\text{deg}}$. This yields 88 unique input sequences with stochastic initial positions and velocities. The stimuli were scaled and cropped to fit the required FNN input shape of 144×256 pixels. This resulted in an image sequence: $\{\mathbf{x}_0, \ldots, \mathbf{x}_T\}$, where each $\mathbf{x}_i \in \mathbb{R}^{H\times W}$. Stimuli were generated using the tools available at `https://github.com/dyballa/NeuralEncodingManifolds`.

The FNN (Wang et al., 2025) processes 2.33-second sequences of 70 frames each, corresponding to 30 frames per second. Since in Dyballa et al. (2024a) the trials were 1.25 s long, we adapted the stimuli to contain 37 frames to maintain consistency with the FNN framework. We scaled all stimuli by a factor of 0.7 to optimize stimulus discriminability across different network layers.

A.4 Sampling

Neural responses were computed using PyTorch and extracted by sampling activations from 2000 units across selected FNN layers. Within each layer, 40 feature maps were sampled. Then, 50 neurons were sampled from each feature map. Feature map sampling probabilities were calculated from the mean maximum response across all neurons within each map, while neuron sampling probabilities within each selected feature map were based on individual neuron maximum responses, biasing the sampling to include active neurons. This sampling procedure was chosen to ensure comparability to the biological results from Dyballa et al. (2024a). This sampling procedure was tested and validated in Dyballa et al. (2024a), we performed further tests with random sampling to validate this bias does not filter out relevant structures. Increasing the sampling rate beyond 2000 units did not significantly alter manifold topology but hindered cluster separation in diffusion map analysis. The resulting tensor data had dimensions $(N \times S \times O \times T)$ with $N = 2000$ neurons, $S = 11$ stimulus types, $O = 8$ orientations and $T = 37$ time steps. For manifold construction, the optimal spatial frequency was

selected (resulting in $S = 6$ stimuli) whereas for classification performance all spatial frequencies were kept. We report results from a single random seed per layer, as preliminary analysis showed consistent manifold structure across different random activity samples. These neural activation tensors served as input for subsequent classification and manifold analysis. This sampling procedure was developed by Dyballa et al. (2024a) and tested against other sampling methods there. We also experimented with the sampling procedure, finding that random sampling and increased sampling rate did not introduce qualitative changes to the manifolds.

A.5 Stimulus adequacy

For every FNN layer investigated in this paper, we extracted the activation to the stimulus ensemble consisting of gratings and flows (see Section A.3) as well as to a 100-second-long natural input video from the MICrONS functional dataset (Bae et al., 2025), downloaded from `s3://bossdb-open-data/iarpa_microns/minnie/functional_data/stimulus_movies/`. Both stimulus sets produced similar activation magnitudes across the entire network (see Figure 8), which shows the adequacy of the stimulus ensemble used for testing the FNN.

Additionally, we calculated the orientation selectivity (OSI) and direction selectivity (DSI) for gratings + flows and for a pink noise stimulus, as done in Wang et al. (2025). We found comparable OSI and DSI distributions (see Figure 9).

A.6 Classification accuracy

The stimulus classification accuracy based on the individual layer activities was obtained from training multinomial logistic regression classifiers (scikit-learn, with solver L-BFGS) using 5-fold cross-validation. We used only the sampled neurons for classifying the 11 stimuli. For each layer and each time point t, two feature sets were constructed: (`i`) the mean activity over frames $0 \rightarrow t$ (increasing window) and (`ii`) the mean activity over frames $t \rightarrow$ *end* (decreasing window). The maximal classification accuracy for all feature sets is reported in Table 1. For comparison, we also evaluated K-nearest neighbor classifiers ($K = 3$) using leave-one-out cross-validation. Results are summarized in Table 1 and Figure 7.

A.7 Construction of decoding manifolds

For building the *decoding manifolds*, we applied PCA (scikit-learn) to the averaged activity data. In total, the *decoding manifolds* contain 48 points, consisting of 6 stimuli and 8 movement directions each. The 6 stimuli were obtained from a majority vote of all neurons on the optimal spatial frequency eliciting higher responses. The decoding manifolds use different colors for each stimulus, as introduced in Figure 1. Different spatial frequencies of the same stimulus are summarized with the same color. To construct *decoding trajectories*, we treated each time step as a separate data point rather than averaging across time before applying PCA. We prepended each trajectory with a zero-activity time step to establish a common origin for all stimulus conditions. In both cases, we reduced the dimensionality to three components for visualization after verifying that further dimensions did not encode qualitatively new information. We constructed biological *decoding trajectories* using experimental data from Dyballa et al. (2024a), available at `https://github.com/dyballa/NeuralEncodingManifolds`. For the biological decoding trajectories, we did not use the additional zero-activity time step since a baseline activity level was already provided by the inter-stimulus intervals in the experiments.

A.8 Construction of neural encoding manifolds

At a high level, the motivation for constructing *neural encoding manifolds* is to find a space in which one can examine the global topology of neuronal populations based on their stimulus selectivities and temporal response patterns (Dyballa et al., 2024a). The neural encoding manifold is constructed in a three-step procedure. First, a 3-tensor is built with the temporal responses from each neuron for each stimulus, and decomposed using Nonnegative Tensor Factorization (details below); each component is comprised of neural, stimulus, and temporal response factors. The neural factors then serve as position coordinates, embedding the neurons into a stimulus-response framework called the neural encoding space. Second, we construct a data graph in this neural encoding space using the

IAN algorithm (Dyballa and Zucker, 2023). Third, applying diffusion maps (Coifman et al., 2005; Coifman and Lafon, 2006) to the data graph yields the manifold.

The methodological choices in our manifold construction procedure are made in accordance with Dyballa et al. (2024a), where extensive parameter analysis for biological neural data was conducted. Since *neural encoding manifolds* computed with these specific parameters represent the only available comparison for biological data from the visual system, we maintained their parameter settings to ensure direct comparability between artificial and biological neural representations. We further conducted analysis for FNN-specific parameters, such as the sampling procedure, by adapting their code to fit the FNN requirements.

A.8.1 Preprocessing

The input tensor of neuronal activity (see above) was preprocessed in several steps (using NumPy and SciPy). First, the individual responses were smoothed along the time dimension using a one-dimensional Gaussian kernel with $\sigma = 3$. Next, we grouped the stimuli into *medium* versus *high* spatial frequencies and selected the one exhibiting higher response magnitudes. The temporal responses for the 8 directions of motion were then concatenated together into a single vector. Finally, we normalized each response and rescaled it by the relative activations of the neuron. The resulting tensor $\mathbf{T}$ had shape $((N = 2000) \times (S = 6) \times (O * T = 296))$.

A.8.2 Nonnegative Tensor Factorization

Next, Nonnegative Tensor Factorization (see (Williams et al., 2018) for an overview and applications to neuroscience) was applied to our tensor $\mathbf{T}$. It was decomposed into typically 10–15 rank-1 tensors which are obtained from the outer product of three vectors each. The number of components was chosen separately for each data sample as specified in Dyballa et al. (2024a). The factors in each component are scaled to unit length, and their magnitudes absorbed by a scalar λ_r:

$$\tilde{\mathbf{T}} = \sum_{r=1}^{R} \lambda_r \mathbf{v}_r^{(1)} \circ \mathbf{v}_r^{(2)} \circ \mathbf{v}_r^{(3)} = [\lambda; \mathbf{X}^{(1)}; \mathbf{X}^{(2)}; \mathbf{X}^{(3)}] \tag{1}$$

For the second equality, the factor matrices $\mathbf{X}^{(k)}$ are constructed using the factor vectors $\mathbf{v}_r^{(k)}$ as columns, and the vector λ contains all individual λ_rs.

Decomposing the tensor $\mathbf{T}$ into these components is an optimization problem with the following objective function and non-negativity constraints:

$$\min_{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}} \frac{1}{2} ||\mathbf{T} - \tilde{\mathbf{T}}||^2 \tag{2}$$

$$\text{such that} \quad \mathbf{X}^{(k)} \geq 0, \forall k \tag{3}$$

The resulting decomposition is interpretable: the third group of vectors, $\mathbf{v}_r^{(3)}$, describes different temporal response patterns; $\mathbf{v}_r^{(2)}$ contain information about which stimuli exhibit these response patterns; and $\mathbf{v}_r^{(1)}$ are the neuronal factors determining which neurons exhibit the response patterns characterized by $\mathbf{v}_r^{(2)}$ and $\mathbf{v}_r^{(3)}$. During decomposition, circular permutations were applied to detect patterns irrespective of the preferred orientations of specific neurons (again, this is necessary to ensure compatibility with the biological results from (Dyballa et al., 2024a)).

Using the OPT method from Tensor Toolbox (Bader et al., 2023)), we ran the decomposition 50 times (different initializations) for each number of components and dataset to ensure robust decomposition results and the choice of the number of factors, R. The manifolds were robust to small changes in R, therefore the heuristic for choosing R based on the explained variance of the decomposition outlined in Dyballa et al. (2024a) proved sufficient. For building the manifolds, we used the result with smallest reconstruction error among the 50 initializations.

A.8.3 Neural encoding space

Following Dyballa et al. (2024a), we now reformulate the above decomposition to construct the neural encoding space. By defining the diagonal matrix $\mathbf{\Lambda}$ with $\mathbf{\Lambda}_{rr} = \lambda_r$, we obtain:

$$\tilde{\mathbf{T}} = \mathbf{X}^{(1)}\mathbf{\Lambda}(\mathbf{X}^{(2)} \circ \mathbf{X}^{(3)}) \tag{4}$$

Since the first matrix, $\mathbf{X}^{(1)}$, represents the neuronal factors, we denote it by $\mathcal{N}$. Now, define a matrix $\mathbf{B}$ with columns $\mathbf{b}_{:,r}$:

$$\mathbf{b}_{:,r} = vec(\mathbf{v}_r^{(2)} \circ \mathbf{v}_r^{(3)}) \tag{5}$$

Finally, we obtain a matrix representation of $\mathbf{T}$ with respect to neuronal factors as $\mathbf{X}_{\mathcal{N}}$:

$$\mathbf{X}_{\mathcal{N}} = \mathbf{B}\mathbf{\Lambda}\mathcal{N}^T \tag{6}$$

This reformulation constructs the neural encoding space. The unit-norm basis vectors of this space are given by the columns of $\mathbf{B}$. We define the neural matrix containing the positions of all neurons in this space as $\mathcal{N}_\lambda = \mathcal{N}\mathbf{\Lambda}$. The distances between any two neurons in this space reflect their similarity in stimulus-selective temporal response patterns. Intuitively, neurons with similar selectivity profiles and temporal dynamics should be positioned close together, while neurons with dissimilar response characteristics should be farther apart.

A.8.4 Iterated adaptive neighborhoods (IAN)

Within this neural encoding space, we construct a weighted graph of the data by inferring a similarity kernel. This is achieved using the Iterated Adaptive Neighborhoods (IAN) algorithm (Dyballa and Zucker, 2023), which infers an adaptive local kernel without the need for pre-specifying a fixed neighborhood size.

IAN first constructs the unweighted Gabriel graph for the data points. In addition, a weighted graph is constructed using a multiscale Gaussian kernel based on the discrete neighborhood graphs. Subsequently, the graph is iteratively pruned by ensuring consistency between the discrete and continuous neighborhoods. The resulting weighted graph is represented by the adjacency (kernel) matrix $\mathbf{K}$. This matrix contains similarities computed using locally tuned Gaussian kernels.

A.8.5 Diffusion maps

Diffusion Maps (Coifman et al., 2005; Coifman and Lafon, 2006) are a dimensionality reduction technique that retain distances and preserve the intrinsic geometry of the manifold. The diffusion process is based on graph Laplacian normalization from spectral graph theory.

In detail, we use the weighted graph obtained from IAN as the weighted adjacency matrix $\mathbf{K}$. The first step is to normalize and symmetrize it to produce $\mathbf{M}_s$:

$$\mathbf{d}_i = \sqrt{\sum_j \mathbf{K}_{ij} + \epsilon} \tag{7}$$

$$\mathbf{M}_s = \frac{\mathbf{K}}{\mathbf{d}\mathbf{d}^T} \tag{8}$$

This normalization ensures that nodes of high degree do not dominate the analysis. We then calculate the spectral decomposition of $\mathbf{M}_s$ with eigenvalues $\lambda_0 = 1 \geq \lambda_1 \geq \lambda_2 ...$ and eigenvectors $\boldsymbol{\psi}_i$ for $t = 1$ diffusion steps using $L = 20$ eigenvalues:

$$\mathbf{M}_{s,ij}^t = \sum_{l=0}^{L} \lambda_l^{2t} \boldsymbol{\psi}_l(i)\boldsymbol{\psi}_l(j) \tag{9}$$

Finally, from the spectral decomposition, we obtain the diffusion map with diffusion coordinates:

$$\Psi_t(i) = \begin{pmatrix} \lambda_0^t \psi_0(i) \\ \lambda_1^t \psi_1(i) \\ \vdots \\ \lambda_{L-1}^t \psi_{L-1}(i) \end{pmatrix} \tag{10}$$

Plotting the data using these diffusion coordinates yields the *neural encoding manifold*.

A.8.6 Encoding manifold visualization

For visualization purposes, we optionally applied metric multidimensional scaling (MDS) to the diffusion map coordinates. This was done by computing pairwise squared Euclidean distances using the first diffusion coordinates, constructing the corresponding Gram matrix $\mathbf{G} = -0.5 * \mathbf{D}^2$, and applying kernel PCA to obtain a lower-dimensional embedding. This preserves the distance relationships from the diffusion map while combining multiple diffusion coordinates, enabling a clearer visualization of the manifold structure.

Based on the manifold topology, we selected groups of neurons to investigate via their PeriStimulus Time Histograms (PSTH). We averaged their activity across trials and constructed the PSTHs as a 2-D heatmap, where each row contains the temporal activity in response to a particular direction of motion (as displayed in Figure 1). Additionally, we calculated the average response intensity over time for these groups and reported the s.e.m. using the shaded regions (see insets in Figure 2A,D).

A.9 Manifold metrics

We computed the following metrics to analyze neural encoding manifolds in Figures 5 and 6:

OSI An Orientation Selectivity Index was computed as:

$$OSI_n = \max_s OSI_{n,s} \tag{11}$$

$$OSI_{n,s} = \frac{R_{n,s}(\theta^*) - \frac{1}{2}\big(R_{n,s}(\theta^* + 90^\circ) + R_{n,s}(\theta^* - 90^\circ)\big)}{R_{n,s}(\theta^*) + \frac{1}{2}\big(R_{n,s}(\theta^* + 90^\circ) + R_{n,s}(\theta^* - 90^\circ)\big) + \epsilon} \tag{12}$$

$$\theta^* = \arg\max_\theta R_{n,s}(\theta) \tag{13}$$

where $R_{n,s}(\theta)$ is the mean response of neuron n to stimulus s at orientation θ.

Mean activity We computed mean activities by averaging each neuron's response across all time steps, directions, and stimuli.

Temporal variance We calculated the temporal variance for each stimulus and direction combination. We then averaged these variances for each neuron.

Preferred stimulus The preferred stimulus for each neuron was obtained by finding the stimulus exhibiting the highest average activity across stimuli and time steps for each neuron.

A.9.1 Tubularity of a Bundle of Trajectories

Goal. To investigate neural dynamics, we modeled trajectories by bundling them into tubular neighborhoods around a central skeleton (**?**). We operationlized this idea for discrete data using the tubular neighborhood theorem (**?**), which guarantees that smooth submanifolds admit non-intersecting neighborhoods diffeomorphic to their normal bundles. Let $\{\gamma_i\}_{i=1}^m \subset \mathbb{R}^D$ denote a set of m trajectories (curves). We define this set as *tubular* if it remains close to a common centerline c and exhibits minimal transverse intersections. Formally, the tube is obtained by expanding c with a radius profile $R(\cdot)$ such that all points at parameter u within distance $R(u)$ of $c(u)$ are included. In practice,

curves are first clustered (e.g., via HDBSCAN (Campello et al., 2013) using the Sobolev H^1 metric, or with ground truth) to separate distinct tubes before computing tubularity scores. We introduce *tightness*, which measures how tight a group of curves is around the centerline, and *crossings*, measuring how many transverse crossings occur in each trajectory bundle. Figure **??** illustrates this idea.

A.10 Tubularity

Before calculating tubularity metrics, we standard-scale the data and apply PCA to obtain a 10-dimensional embedding, thereby speeding up the computation. While visualizations use only the first 2-3 dimensions, all metrics are calculated in the 10-dimensional space. To ensure comparability, we resampled all trajectories to length 100. For statistical analysis, we generated 100 bootstrapped samples, and using ground-truth clusters, performed Bonferroni-corrected Mann-Whitney U tests on our hypotheses.

We formalize how "tight" a group of curves is around the centerline: We reparameterize each curve by normalized arc length $u \in [0,1]$ and resample to $\{u_k\}_{k=1}^{M}$. Let $x_i(u_k) \in \mathbb{R}^D$ denote the samples and $\tau_i(u_k)$ their unit tangents. We define the *mean curve* as the pointwise average:

$$c(u_k) \;=\; \frac{1}{m}\sum_{i=1}^{m} x_i(u_k), \qquad r_i(u_k) \;=\; \|x_i(u_k) - c(u_k)\|.$$

The tightness score is calculated by averaging quantile tube radii across bins $\{I_b\}_{b=1}^{B}$ that partition $[0,1]$, using a high quantile $q \in [0.8, 0.95]$ to ensure robustness to noise. We normalize each tube's tightness score by tube length.

$$S_{\text{tight}} \;=\; \frac{1}{B}\sum_{b=1}^{B} \text{quantile}_q\{\, r_i(u) : u \in I_b \text{ over all curves } \}.$$

The second quantity assessed is the uniformity of the tubes relative to one another. That is, the degree to which crossings occur in our defined bundle of curves. Tubes are considered disorganized when distinct curves pass near each other with *transverse* directions. Let $d_{ij}(u,v) = \|x_i(u) - x_j(v)\|$ and $\phi_{ij}(u,v) = 1 - \langle \tau_i(u), \tau_j(v)\rangle^2 \in [0,1]$ (large for near-orthogonal tangents). Using a Gaussian kernel $K_\varepsilon(\rho) = \exp(-\rho^2/(2\varepsilon^2))$, we softly count encounters:

$$\mathcal{X}_\varepsilon = \frac{2}{m(m-1)} \sum_{i<j} \int_0^1 \int_0^1 K_\varepsilon\big(d_{ij}(u,v)\big)\, \phi_{ij}(u,v)\, du\, dv.$$

S_{tight} and S_{cross} only depend on distance, unit-tangent inner product, and arc-length. Therefore, they are invariant to translations, rotations, and re-timing. We emphasize that, for both scores, smaller values indicate more tubular curve bundles, while larger values indicate fewer tubular curve bundles.

A.11 Visualizations

Interactive three-dimensional plots of the manifolds were computed using Plotly. Other plots were created with Matplotlib and TUEplots.

A.12 Software

All software (Table 2) is used in accordance with its respective license.

A.13 Compute

The experiments were conducted on an HPC cluster. FNN sampling uses randomly selected GPUs (RTX 2080 Ti, or better). All other experiments were performed on CPU. All experiments required less than 30 GB memory. In total, 10 tensor decomposition experiments were run on CPU, each

Table 2: Software packages used in this work.

Package	Version	License
MATLAB Tensor Toolbox (Bader et al., 2023)	3.6	BSD-2
IAN (Dyballa and Zucker, 2023)	1.1.2	BSD-3
NeuralEncodingManifolds (Dyballa et al., 2024a)	N/A	BSD-2
NumPy (Harris et al., 2020)	1.25.0	BSD-3
SciPy (Virtanen et al., 2020)	1.15.3	BSD-3
scikit-learn (Pedregosa et al., 2011)	1.7.1	BSD-3
PyTorch (Paszke et al., 2019)	2.6.0	MIT
Matplotlib (Hunter, 2007)	3.10.1	PSF-based (BSD-compatible)
Plotly (Inc., 2015)	6.0.0	MIT
TUEplots (Krämer et al., 2024)	0.2.0	MIT

taking 2 days on a single CPU. Preliminary results not included in the paper required another 50 tensor decomposition experiments.

B Data and code availability

The code is available at `https://github.com/JohannesBertram/FNN_Manifolds`

C Supplemental Figures

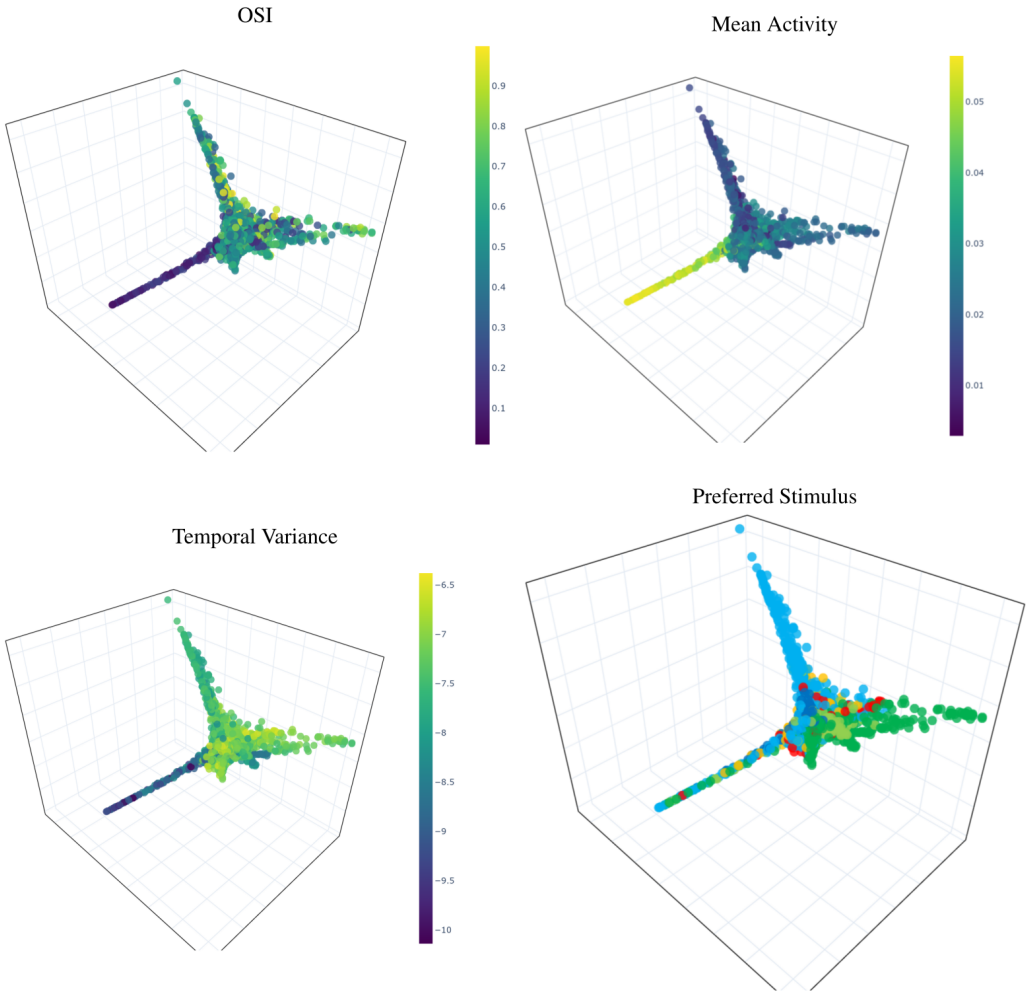

Figure 5: **Feedforward encoder L8**: Supporting information for Figure 2. The intensity arm is clearly visible, exhibiting high mean activity. The low temporal variance, low Orientation Selectivity Indices, and unstructured preferred stimuli show the absence of complex activity patterns in this intensity arm.

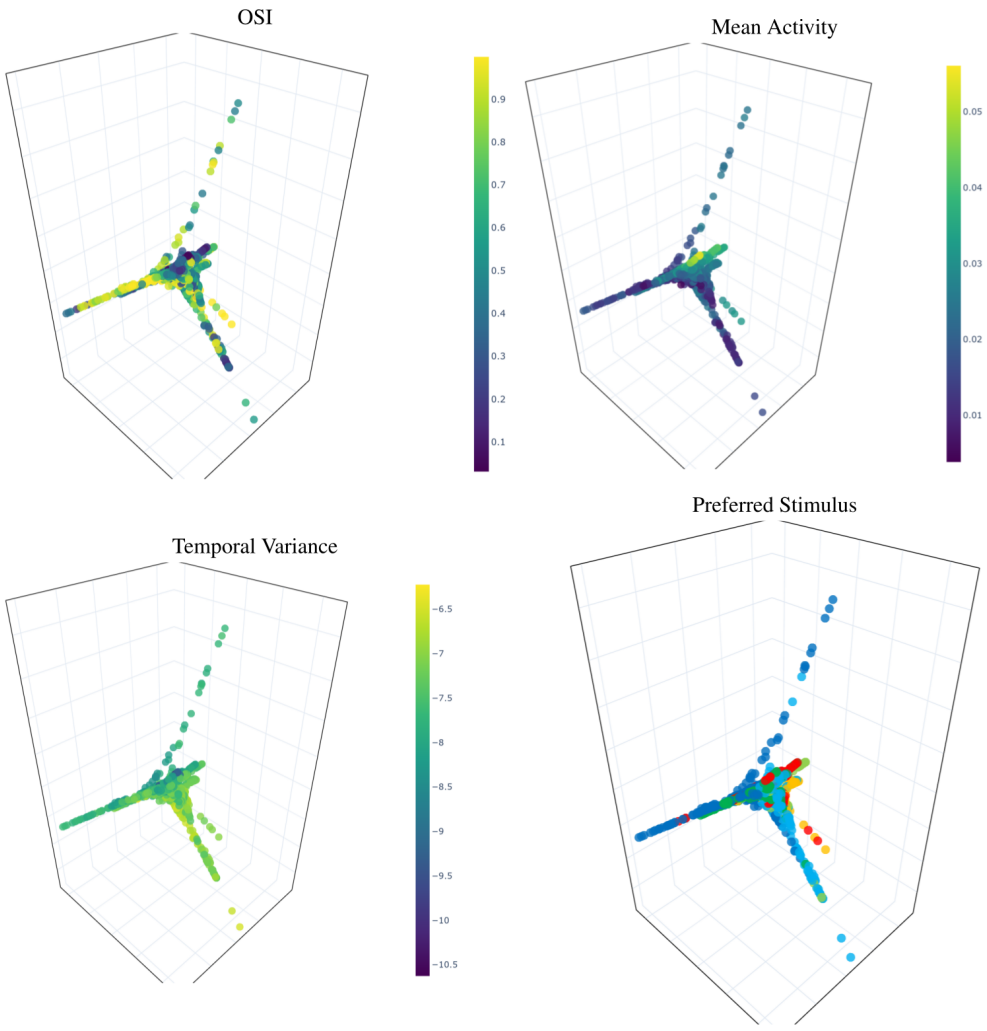

Figure 6: **Recurrent hidden state**: Supporting information for Figure 2. In contrast to Figure 4, there is no intensity arm dominating the manifold structure. Instead, all arms show structured, complex selectivity patterns.

Table 3: **Tubularity metrics for biological and FNN data**. Low tightness and crossings values indicate high tubularity. Aligning with the visualization in Figure 3, the biological trajectories show highly tubular organizations compared to FNN. Method details in Appendix A.9.1.

Layer	**Ground Truth Labels**		**HDBSCAN Labels**		
	$\mathbf{S}_{\text{tight}}$	$\mathbf{S}_{\text{cross}}$	$\mathbf{S}_{\text{tight}}$	$\mathbf{S}_{\text{cross}}$	**Clusters**
Retina	0.0688	1.29×10^{-05}	0.1017	1.06×10^{-05}	4
V1	0.1357	4.06×10^{-05}	0.1859	3.50×10^{-05}	4
Enc1	0.2018	2.87×10^{-04}	0.7680	1.66×10^{-04}	1
Enc13	1.9885	1.77×10^{-06}	4.3461	1.09×10^{-06}	3
Rec	0.1228	2.65×10^{-07}	0.1697	1.53×10^{-07}	4
RecOut	0.1209	5.72×10^{-07}	0.1650	5.34×10^{-07}	4
Readout	0.3307	3.96×10^{-06}	0.4320	5.40×10^{-06}	4
Output	0.1483	3.53×10^{-06}	0.2784	1.12×10^{-06}	3

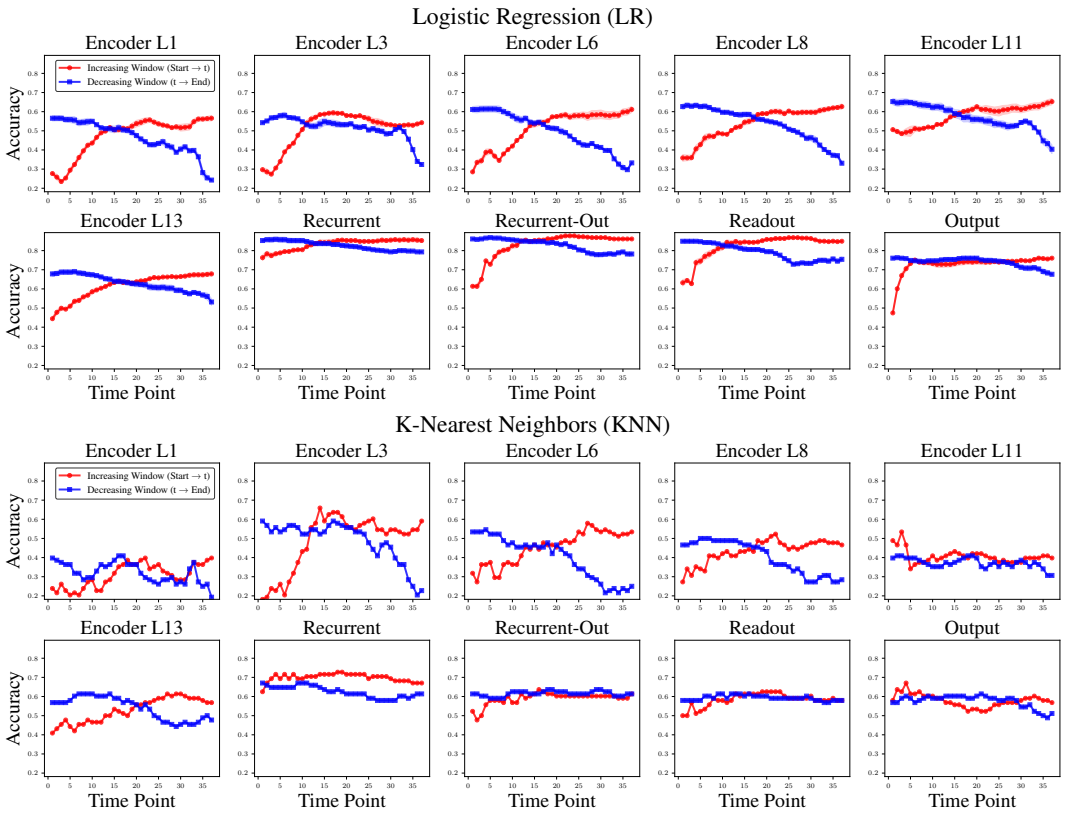

Figure 7: **Logistic regression (LR, top) and K-Nearest Neighbor (KNN, K=3, bottom) classifier accuracy** for each layer. We use increasing time windows (timesteps 0 $\rightarrow$ t, red) or decreasing time windows (t $\rightarrow$ 37, blue) to calculate the accuracies. Shaded regions for LR show the s.e.m. The maxima across panels are summarized in Table 1.

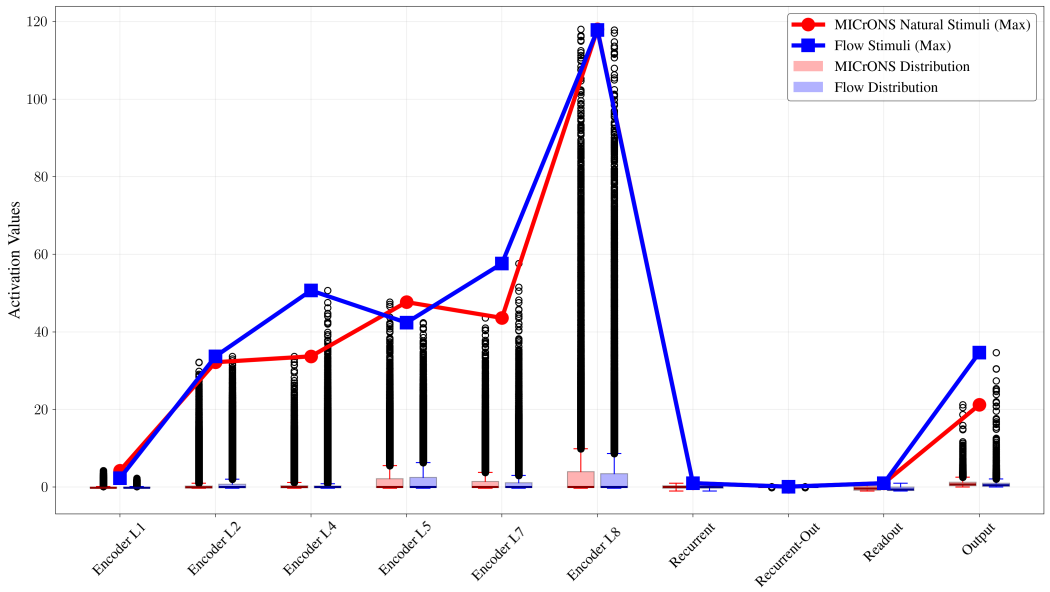

Figure 8: **Activation function output distributions and maxima** for natural MICrONS (Bae et al., 2025) input videos and the flow stimulus ensemble (Dyballa et al., 2024a). The comparable activity across network layers shows the adequacy of investigating the FNN with flow stimuli. The differences in magnitudes across layers are explained by the activations functions (GELU in the *encoder*, Tanh in the *recurrent* and *readout* modules).

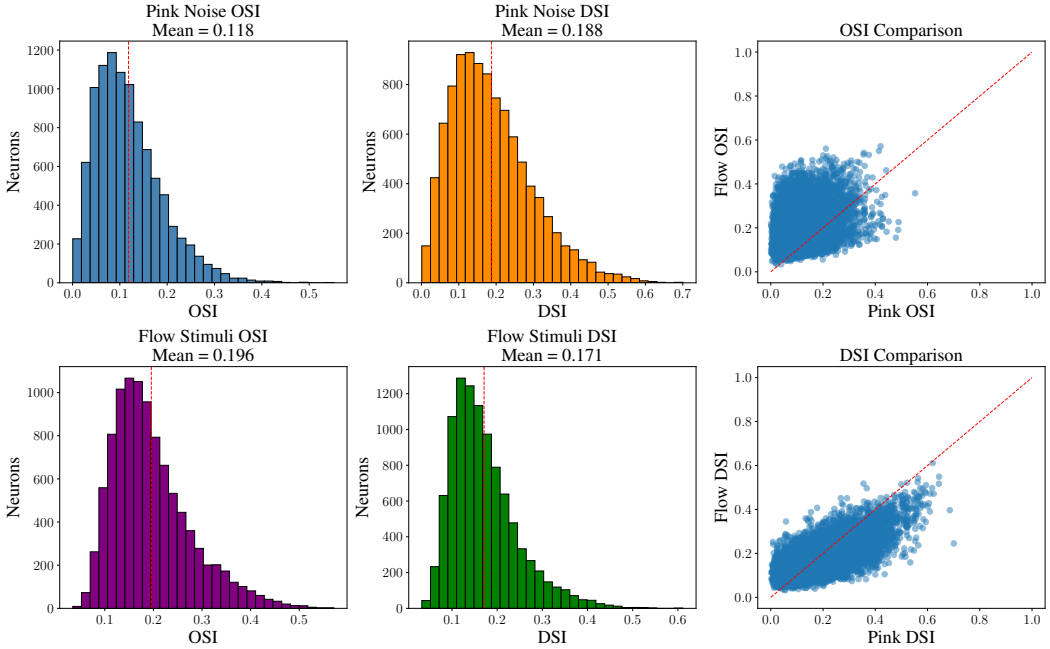

Figure 9: **OSI and DSI of FNN output** for pink noise (as used in Wang et al. (2025)) and for the stimulus ensemble from Dyballa et al. (2024a), meaned over the different stimuli.

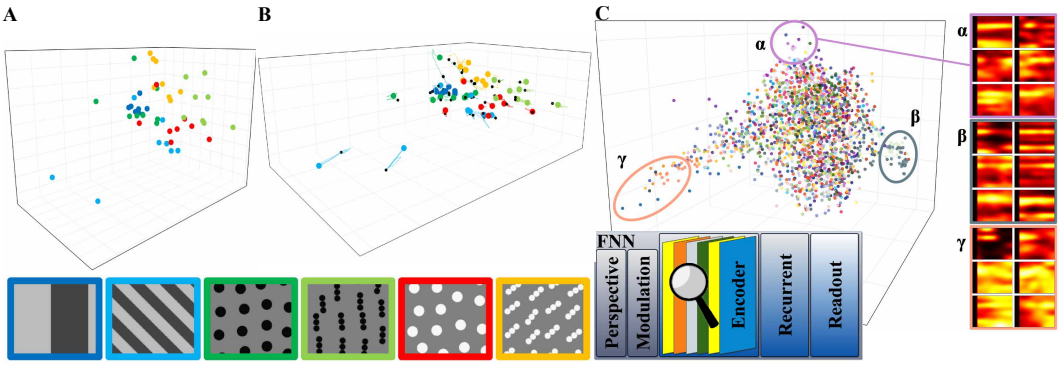

Figure 10: **Encoder L8 decoding manifold, trajectories and encoding manifold without intensity artifacts**. Without the intensity artifacts there is no temporal development at all in the decoding trajectories (comparable to encoder L1) apart from the jump after the 0-th step. The non-selective high intensity neurons are padding artifacts at the edges of the image. In the encoder, due to spatial convolutions, the effect of these artifacts spreads out across the feature maps. This is supported by the intensity smoothly organizing the manifold with a transition from intensity-only neurons to selective responses. In the recurrent stage, the function of the attention layer is capable of filtering exactly those artifacts out. The artifacts are reintroduced by the recurrent-output convolution, but then filtered out by the readout interpolation from central neurons only.