Time-Evolving Dynamical System for Learning Latent Representations of Mouse Visual Neural Activity

Liwei Huang[1,2], Zhengyu Ma[2],* Liutao Yu[2], Huihui Zhou[2], Yonghong Tian[1,2,3]

[1]School of Computer Science, Peking University, China
[2]Peng Cheng Laboratory, China
[3]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, China
huanglw20@stu.pku.edu.cn,
{mazhy, yult, zhouhh}@pcl.ac.cn, yhtian@pku.edu.cn

Abstract

Seeking high-quality representations with latent variable models (LVMs) to reveal the intrinsic correlation between neural activity and behavior or sensory stimuli has attracted much interest. In the study of the biological visual system, naturalistic visual stimuli are inherently high-dimensional and time-dependent, leading to intricate dynamics within visual neural activity. However, most work on LVMs has not explicitly considered neural temporal relationships. To cope with such conditions, we propose Time-Evolving Visual Dynamical System (TE-ViDS), a sequential LVM that decomposes neural activity into low-dimensional latent representations that evolve over time. To better align the model with the characteristics of visual neural activity, we split latent representations into two parts and apply contrastive learning to shape them. Extensive experiments on synthetic datasets and real neural datasets from the mouse visual cortex demonstrate that TE-ViDS achieves the best decoding performance on naturalistic scenes/movies, extracts interpretable latent trajectories that uncover clear underlying neural dynamics, and provides new insights into differences in visual information processing between subjects and between cortical regions. In summary, TE-ViDS is markedly competent in extracting stimulus-relevant embeddings from visual neural activity and contributes to the understanding of visual processing mechanisms. Our codes are available at *https://github.com/Grasshlw/Time-Evolving-Visual-Dynamical-System*.

1 Introduction

With the rapid development of neural recording technologies, researchers are able to simultaneously record the spiking activity of large populations of neurons, opening up new avenues for exploring the brain [55]. For high-dimensional neural data analysis, an important scientific problem is how to account for the intrinsic correlation between neural activity and behavioral patterns or sensory stimuli. One influential approach is latent variable models (LVMs), which construct low-dimensional latent representations that bridge to behavior or stimuli and explain neural activity well [47, 5, 56, 26, 34]. To deal with the increasing scale of neural population activity, LVMs have evolved from simple mathematical models early on, such as principal component analysis and factor analysis [9, 61, 44], to complex artificial neural networks. More recently, advanced deep learning algorithms have enabled LVMs to extract high-quality representations from neural activity without knowledge of experimental labels [59, 42, 22, 36], or to incorporate behavioral information into models to constrain the shaping of latent variables [37, 25, 45, 52, 2, 23]. These approaches have contributed to the analysis of neural

---
*Corresponding author.

activity in various ways, including predicting neural responses [42, 27], decoding related motion patterns or simple visual scenes [36, 48], and constructing interpretable latent structures [63, 3].

Although LVMs have sparked strong interest in uncovering the underlying dynamical structure of neural activity, most studies have focused on neural data recorded from motor brain regions under specific controlled behavioral settings [13, 42, 63, 36, 43], such as pre-planned reaching movements [18]. There is a paucity of studies on LVMs exploring neural data from the visual cortex [21, 62, 48]. Given the significant challenge of elucidating the correlation between neural activity and visual stimuli [15, 28, 58, 17], the development of robust models for extracting vision-related latent representations is critically imperative. Furthermore, due to the complex visual neural dynamics, incorporating temporal structure into models to obtain time-dependent latent representations may be a key point for the analysis of visual neural activity. However, there is also a lack of LVMs that explicitly model neural temporal relationships under visual stimuli.

In this work, we propose the Time-Evolving Visual Dynamical System (TE-ViDS), which aims to generate high-quality latent representations from visual neural activity by disentangling neural components related to visual stimuli from those influenced by internal states. First, we introduce temporal structures to explicitly establish temporal relationships in latent variables, allowing them to evolve over time and capture the temporal dependency inherent in neural activity. Second, to sufficiently utilize the characteristics of visual neural activity, we adopt the split structure approach [36] and design distinct loss functions to construct two specialized parts of latent representations. The *external* latent representations aim to capture stimulus-relevant components within visual neural activity, while the *internal* latent representations reflect the dynamical internal states that influence the animal's sensory capability [39, 4]. In doing so, TE-ViDS can generate latent representations that are more relevant to visual stimuli while also explaining the effects of internal states. We evaluate our model on synthetic and mouse visual datasets, comparing it with leading alternatives. Our model demonstrates its ability to construct meaningful latent representations, yield a greater degree of correlation between neural activity and visual stimuli, and explain the visual information processing mechanisms of the mouse visual cortex. Specifically, our main contributions are as follows.

- We introduce state factors to filter temporal information, enabling TE-ViDS to progressively compress neural activity and evolve latent variables. Regarding the distinct objectives of the two parts of latent representations, we apply self-supervised contrastive learning to shape the *external* and utilize a time-dependent prior distribution to guide the *internal*.

- Through evaluation on synthetic datasets, we show that our model more effectively recovers latent structure and handles time-sequential data well.

- Through evaluation on mouse visual datasets, we demonstrate that our model outperforms alternative models in decoding natural scenes and natural movies, as well as in extracting clear temporal trajectories of neural dynamics at different time scales.

- Further analysis reveals that our model can explain the potential variability in visual information processing between subjects and provide new evidence for the functional hierarchy of the mouse visual cortex.

2 Related Work

With the advancement of deep learning, the application of cutting-edge learning algorithms and the innovative design of model structures have greatly promoted the development of LVMs in neuroscience. Some prominent works are summarized below.

The approaches based on neural networks have become major avenues for discovering latent representations underlying neural activity, which better elucidate the mechanisms of neural representations. A well-known model is latent factor analysis via dynamical systems (LFADS), which used RNNs in a sequential VAE framework to extract precise firing rate estimates and predict observed behavior for single-trial data on motor cortical datasets [42, 29, 30]. Recurrent switching linear dynamical systems (rSLDS) [35, 53] and low-rank RNNs [41] also introduced recurrent structures, facilitating the understanding of complex nonlinear neural dynamics. Through specific latent variable designs, pi-VAE [63] and Swap-VAE [36] built interpretable latent structures linked to motor behavior patterns. Furthermore, numerous studies have made significant contributions to achieving the goal of dissociating behaviorally relevant and irrelevant components in neural activity. Targeted neural
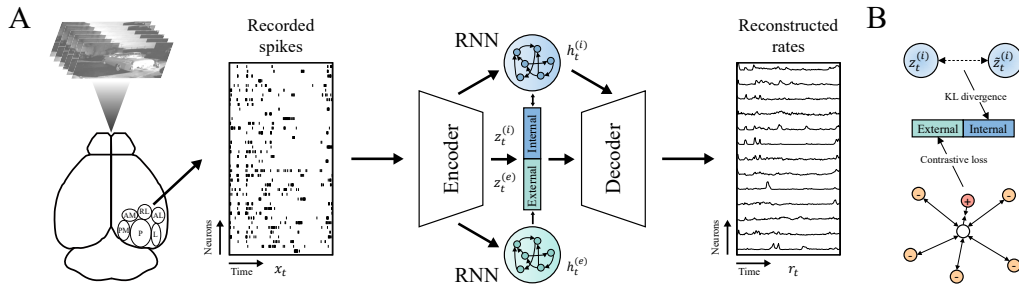
Figure 1: The method overview. **A**. The illustration of TE-ViDS for analyzing visual neural activity in the mouse visual cortex. The encoder extracts spatial features from sequential spike data. The latent variables are evolved conditionally on features of the encoder and RNNs' state factors over time. The decoder maps latent variables to inferred firing rates. **B**. The illustration of different learning objectives for the two parts of latent representations of TE-ViDS. For *external* latent representations, we apply contrastive loss to encourage them to distinguish the stimulus-relevant components. Given a reference sample (white dot), the red dot is a positive sample and the orange dots are negative samples. For *internal* latent representations, we use the KL divergence to constrain their distribution to a time-dependent prior distribution.

dynamical modeling (TNDM), based on LFADS, constructed a two-pathway structure for separation [25]. PSID [45], DFINE [1], and DPAD [46] were dedicated to introducing supervised information into dynamical systems to capture behavior-relevant dynamics.

In the visual domain, several latent variable models have made an effort to advance the understanding of visual neural dynamics. One work used Gaussian process models [19] and found that anesthesia-induced internal state fluctuations lead to correlated variability, while another proposed a network-based linear dynamical system (fLDS) that captures neural variability under drifting grating stimuli [21]. In addition, a rectified latent variable model built latent variables that show a spectrum of functional groups as neurons in the mouse visual cortex under drifting gratings [40], and a flow-based generative model recovered the distribution of visual neural activity [7]. Recently, CEBRA, a self-supervised learning model, obtained consistent latent representations and made progress in decoding movies [48]. Although these studies cover various types of models, most of them are limited to simple visual stimuli and the task of reconstructing neural responses, leaving a gap for constructing high-quality latent representations from visual neural activity under naturalistic stimuli.

3 Methods

To generate latent representations that can effectively explain the characteristics of visual neural activity, we introduce the Time-Evolving Visual Dynamical System (TE-ViDS), which evolves two parts of representations over time. In this section, we elaborate the concrete implementations of the model architecture and model learning (visualized in Figure 1).

Basic notations. Neural responses recorded from large numbers of neurons are mostly in the form of spikes, which are regarded as discrete events. In practice, it is common to discretize a period of time into small time windows and calculate the number of spikes within each window. Consequently, for the neural activity of a population of neurons over a period of time, we define a sequence input as $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T) \in \mathbb{R}^{T \times N}$, which represents spike counts of N neurons within T time windows. The corresponding low-dimensional latent variable is denoted as $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T) \in \mathbb{R}^{T \times M}$.

3.1 Model Architecture

As aforementioned, TE-ViDS compresses visual neural activity into *external* latent representations and *internal* latent representations ($\mathbf{z}_t = [\mathbf{z}_t^{(e)}, \mathbf{z}_t^{(i)}]$). To evolve the two latent variables, we incorporate two dynamical systems into our model, using the state factor $\mathbf{h}_t$ to sift and accumulate temporal information [8, 20, 12]. In this way, the latent variables of the current time step are only conditioned on the input and state factors of the previous time steps, facilitating the capture of dynamics in causal order.

3

External **latent variables.** This part of latent representations aims to capture the stimulus-relevant components of neural activity. To maintain our focus on variability arising from internal brain states, we construct *external* latent variables as deterministic values:

$$\mathbf{z}_t^{(e)} = f_{\text{enc}}^{(e)}\left(f_{\text{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}^{(e)}\right). \tag{1}$$

Internal **latent variables.** This part of latent representations aims to reflect dynamical internal states that contain high variability and noise. To model such dynamics, *internal* latent variables are constructed as stochastic values that evolve over time. The tractable parameterized distribution (approximate posterior) is conditioned on both $\mathbf{x}_t$ and $\mathbf{h}_{t-1}^{(i)}$, while the prior distribution is conditioned solely on $\mathbf{h}_{t-1}^{(i)}$, endowing it with a certain degree of spontaneity:

$$\mathbf{z}_t^{(i)}\Big|\mathbf{x}_{1:t}, \mathbf{h}_{1:t-1}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}^2 \cdot \mathbf{I}), [\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}] = f_{\text{enc}}^{(i)}\left(f_{\text{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}^{(i)}\right), \tag{2}$$

$$\tilde{\mathbf{z}}_t^{(i)}\Big|\mathbf{h}_{1:t-1}^{(i)} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{z,t}, \tilde{\boldsymbol{\sigma}}_{z,t}^2 \cdot \mathbf{I}), [\tilde{\boldsymbol{\mu}}_{z,t}, \tilde{\boldsymbol{\sigma}}_{z,t}] = f_{\text{prior}}^{(i)}\left(\mathbf{h}_{t-1}^{(i)}\right). \tag{3}$$

Neural activity reconstruction. High-quality latent representations, as compressed forms of the original neural activity, are required to effectively reconstruct their input. In practice, to enrich the dynamic information for a more accurate reconstruction, the spike count observation is related not only to the latent variables but also to the internal state factors:

$$\hat{\mathbf{x}}_t\Big|\mathbf{z}_{1:t}^{(e)}, \mathbf{z}_{1:t}^{(i)}, \mathbf{h}_{1:t-1}^{(i)} \sim P(\mathbf{r}_t), \mathbf{r}_t = f_{\text{dec}}\left(\mathbf{z}_t^{(e)}, \mathbf{z}_t^{(i)}, \mathbf{h}_{t-1}^{(i)}\right), \tag{4}$$

where the parameterized distribution P is chosen to be Poisson [21], i.e., the actual inferred neural activity is spike firing rates $\mathbf{r}_t$.

Recurrent neural networks. The state factor is updated by GRU [11]. By selectively integrating and exploiting input and latent variables, the state factor is crucial for capturing complex sequential dynamics. Besides, since the animal's dynamical internal states are inevitably affected by visual stimuli, $\mathbf{h}_t^{(e)}$ and $\mathbf{h}_t^{(i)}$ corresponding to the two latent representations are updated differently:

$$\begin{aligned}
\mathbf{h}_t^{(e)} &= f_{\text{GRU}}^{(e)}\left(f_{\text{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}^{(e)}\right), \\
\mathbf{h}_t^{(i)} &= f_{\text{GRU}}^{(i)}\left(f_{\text{x}}(\mathbf{x}_t), \mathbf{z}_t^{(e)}, \mathbf{z}_t^{(i)}, \mathbf{h}_{t-1}^{(i)}\right).
\end{aligned} \tag{5}$$

All the functions f above are parameter-learnable neural networks (see Appendix A for details).

3.2 Model Learning

Given that the two parts of latent representations hold different objectives for disentanglement, we introduce the following loss function to optimize all learnable parameters in an end-to-end manner:

$$\mathcal{L} = \mathcal{L}_{\text{recons}} + \beta\mathcal{L}_{\text{contrastive}} + \gamma\mathcal{L}_{\text{regular}}. \tag{6}$$

The first term which deals with the objective of inferring neural activity is formulated as $\frac{1}{T}\sum_{t=1}^{T}\left[\mathcal{L}_{\text{P}}(\mathbf{x}_t, \mathbf{r}_t)\right]$, where $\mathcal{L}_{\text{P}}$ is Poisson negative log likelihood.

The second term encourages *external* latent representations to distinguish stimulus-relevant components through self-supervised contrastive learning. Specifically, for a given sample $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, we randomly select another sequence offset by several time steps as a positive sample, denoted $\mathbf{x}_{\text{pos}} = (\mathbf{x}_{1+\Delta}, \ldots, \mathbf{x}_{T+\Delta})$, where Δ can be positive or negative. The offset is smaller than the length of the sequence to ensure that the positive sample pairs overlap, i.e., the visual stimuli corresponding to the positive sample pairs are similar, which establishes an association between the *external* latent variables and the visual stimuli. Then, a mini-batch of negative samples is randomly selected from the entire training set. Finally, we compute the *external* latent variables of all selected samples and apply the NT-Xent loss [10] as $\mathcal{L}_{\text{contrastive}}$ to them. In doing so, *external* latent representations of the positive sample pairs are driven to be distinguished from the negative samples. For this term, we do not apply the time-wise operation, but flatten the temporal and spatial dimensions of the *external* latent variables for the loss computation. In addition, to enhance the effect of the positive sample, we adopt the practice of the swap operation [36]. We exchange the *external* latent variables

of the given sample with those of the positive sample while keeping the *internal* latent variables unchanged. The new latent representations are then used to compute new inferred firing rates and an additional reconstruction loss.

The third term measures the difference between the prior and the approximate posterior of the *internal* latent variables by the KL divergence, formulated as $\frac{1}{T} \sum_{t=1}^{T} \left[D_{\text{KL}}(\mathbf{z}_t^{(i)} \| \tilde{\mathbf{z}}_t^{(i)}) \right]$. Besides, we compute the L2 norm of the expectation and log-variance of the prior distribution as a regularization to avoid excessive fluctuations over time and to stabilize model training.

β and γ are hyperparameters used to control the severity of the penalty for each loss term. The entire loss function actually obeys the strategy of maximizing the evidence lower bound (ELBO) of the marginal log likelihood in the VAE framework [31], while also accounting for the impact of introducing the contrastive loss [57]. A detailed derivation is given in Appendix B.

4 Experiments

To evaluate the utility of the latent representations constructed by TE-ViDS for the analysis of visual neural activity, we perform a series of experiments to compare our model with alternative LVMs from two aspects, similar to studies focusing on motor brain regions [36, 48]. First, we quantify the performance in decoding visual stimuli using latent representations, which has long served as a research hotspot for unraveling the mechanisms of visual processing [28, 58]. Second, we assess the clarity of latent temporal trajectories extracted from neural dynamics.

4.1 Datasets

We carry out the experiments on two commonly used synthetic datasets and a well-known publicly available mouse visual neural dataset.

Synthetic datasets. The two synthetic datasets are crucial for evaluating the fundamental ability of LVMs to construct accurate latent variables. One is a non-temporal dataset generated from several sets of labels [63, 36], facilitating the assessment of class discrimination capabilities. The other is a temporal dataset constructed by the Lorenz system to test for extracting temporal relationships [21, 54]. A detailed description of the generating procedure is presented in Appendix C.

Mouse visual neural dataset. We use a subset of the Allen Brain Observatory Visual Coding dataset [51], which has been used in a variety of studies, including the construction of brain-like networks [50], the modeling of functional mechanisms [6, 14], and the decoding of visual stimuli [48]. This dataset was collected by Neuropixel probes from six mouse visual cortical regions simultaneously, including VISp, VISl, VISrl, VISal, VISpm, and VISam. Notably, neural activity was recorded while mice passively viewed visual stimuli without any task-driven behavior. The dataset comprises 32 sessions, each involving one mouse. In this work, we choose to analyze five mice (subjects) that have the maximum number of recorded neurons (see Appendix D), and these neurons are evenly distributed across all regions (the coefficient of variation for the number of neurons across six regions is below 0.5). We focus on neural activity in response to natural scenes and a natural movie. For natural scenes, 118 images are presented in random order, each for 250ms and 50 trials. The neural activity in the form of spike counts is binned into 10-ms windows so that each trial contains 25 time points. The natural movie is 30 seconds long with a frame rate of 30Hz, presented for 10 trials. We bin the spike counts with a sampling frequency of 120Hz and align them with the movie timestamps, resulting in four time points for each frame.

Since spike responses are quite variable across trials even under identical experimental conditions, we conduct the basic evaluation on "held-out" trials. Specifically, for each dataset, we randomly split all trials into 80% for training, 10% for validation, and 10% for test.

4.2 Evaluation Metric

Following previous studies [54, 48], we apply two quantitative metrics for evaluation.

Reconstruction score. For the two synthetic datasets, we use linear regression to fit the latent variables of models to the true latent variables, and report the R^2 as the reconstruction score.
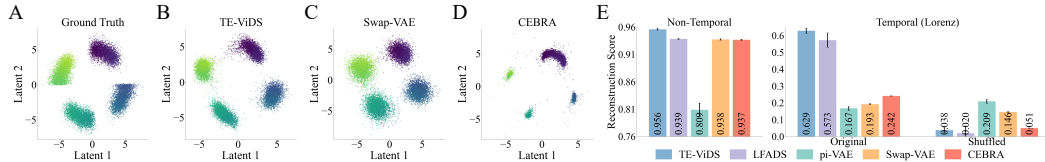
5

Figure 2: Results on synthetic datasets. **A**. The true latent variables of the non-temporal dataset. **B-D**. The inferred latent variables of our model and some alternative models. **E**. The reconstruction scores of all models on the non-temporal and temporal datasets. The standard error is computed on 10 runs with different random initializations.

Decoding score. For the mouse visual neural dataset, we quantify the performance of models in decoding natural scenes/natural movie frames. In terms of natural scene decoding, we obtain the latent variables of the last 20 time points (50ms-250ms) in each trial, since there is a response latency in the mouse visual cortex when presented with static stimuli [51]. These latent variables are then concatenated into a vector to form the latent representations of each trial, thereby retaining maximal temporal information. We use the KNN algorithm to classify the latent representations of each trial, i.e., to decode the corresponding natural scenes. In terms of natural movie frame decoding, we compute the latent variables of four time points within each frame, averaging them to create the latent representations for that frame. We also use the KNN to predict movie frames based on latent representations (900 frames in total, i.e., 900 classes). The specific KNN procedure is as follows: First, we fit the KNN on the training set and search for the optimal hyperparameter (the number of neighbors) on the validation set, using classification accuracy as the metric. The search range is set to odd numbers between 1 and 20, with odd numbers chosen to reduce decision uncertainty. Then, the accuracy on the test set is reported as the decoding score. Notably, for movie frame decoding, we follow the established approach of CEBRA [48], considering an error of less than 1s (constraint window) between the predicted frame and the true frame as a correct prediction.

4.3 Alternative Models

For a comprehensive analysis, we compare TE-ViDS with several leading LVMs, including three generative models (a sequential: LFADS [42], a supervised: pi-VAE [63], and a self-supervised: Swap-VAE [36]) and a nonlinear encoding method with contrastive learning (CEBRA) [48]. Specifically, pi-VAE[2] incorporates additional supervised information for latent variable construction, but does not model temporal dynamics. Swap-VAE utilizes contrastive learning to construct separated latent variables, aligning with our disentanglement goal, but it also does not explicitly model temporal dynamics. For the two models that are able to model the temporal relationships, LFADS processes sequential neural activity using bidirectional RNNs, while CEBRA encodes temporal features of sequential data with fixed convolutional kernels. Furthermore, for fair comparisons, we build a small version of our model (TE-ViDS-small) with fewer trainable parameters than Swap-VAE (see Appendix E for the number of trainable parameters).

For each dataset, all models are set to latent variables of the same dimension and trained for more than 20,000 iterations until convergence. Additionally, we apply the hyperparameter grid search for all models. More details of the training setup are presented in Appendix F.

4.4 Results on Synthetic Datasets

First, we visualize the reconstructed latent variables on the non-temporal dataset (Figure 2A-D). TE-ViDS reliably separates the different clusters as well as recovers the structure of the true latent variables to form clear arcs. In contrast, some of the alternative models fail to construct similar structures despite separating clusters (Swap-VAE and CEBRA), and others even struggle to separate four clusters (LFADS and pi-VAE; Appendix G). Quantitatively, the reconstruction scores also suggest that our model outperforms all alternative models (Figure 2E).

For the original temporal dataset, TE-ViDS performs significantly better than those models that process sequential spikes at each time point independently, and moderately better than LFADS

---

[2]pi-VAE incorporates the label prior during training, but the inferred latent variables are built without the label prior at the evaluation stage. This way is used in all experiments.

Table 1: The decoding scores (%) for 118 natural scenes on the mouse visual neural dataset. The standard error is computed on 10 runs with different random initializations.

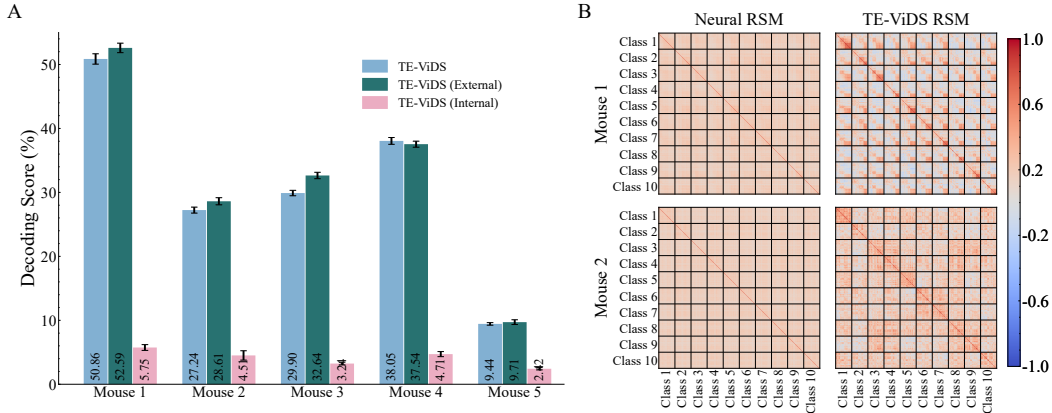| Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
PCA	0.59	1.53	1.53	0.80	0.85
LFADS	30.76±0.65	16.46±0.49	22.20±0.26	19.69±0.38	4.69±0.22
pi-VAE	7.49±0.77	19.42±0.62	22.92±0.37	13.71±1.39	2.22±0.64
Swap-VAE	32.81±1.47	24.34±0.57	14.36±1.31	14.85±1.29	3.92±0.37
CEBRA	1.53±0.15	3.42±0.07	4.86±0.19	2.81±0.23	1.08±0.10
TE-ViDS-small	47.08±2.86	23.95±1.39	29.08±1.47	34.95±0.90	**9.93±0.27**
TE-ViDS	**50.86±0.81**	**27.24±0.47**	**29.90±0.43**	**38.05±0.53**	9.44±0.20

Figure 3: Results on the mouse neural dataset under natural scene stimuli. **A**. The decoding scores (%) of the *full*, *external* and *internal* latent representations of TE-ViDS for 118 natural scenes. **B**. RSMs computed on the original neural representations and TE-ViDS's latent representations, respectively (Mouse 1 and Mouse 2). Each element in a matrix is the similarity between two trials' representations. Each small square involves comparisons between two natural scenes, containing 50 trials.

which also uses RNNs to process sequential data (Figure 2E). Moreover, when we shuffle the time dimension for each trial data in the original dataset to obtain a dataset without temporal dependency, the performance of our model and LFADS shows a drastic degradation. However, for models utilizing time-jittered positive samples, the performance reduction observed in Swap-VAE and CEBRA was comparatively less severe. These results demonstrate the superiority of our model in dealing with temporal data and its sensitivity to temporal relationships.

4.5 Results on the Mouse Neural Dataset under Natural Scene Stimuli

As shown in Table 1, TE-ViDS achieves the highest decoding scores for all mice with a noticeable improvement over other models. In particular, CEBRA, which can encode temporal relationships, instead performs poorly in this downstream task, suggesting that using time filters to extract temporal neural features is less suitable in this case. In contrast, the time-evolving latent representations constructed by our model reliably capture temporal relationships encoded in the mouse visual cortex under different static scene stimuli, which may be helpful for natural scene discrimination. We further evaluate the decoding performance of the *external* and *internal* latent variables separately (Figure 3A). We find that the *external* latent representations significantly outperform the *internal* latent representations, supporting our assumption that the former are tuned to stimulus-relevant components of neural activity, while the latter point to stimulus-irrelevant internal states.

Although our model achieves the highest performance, we find that the decoding scores for different mice show large differences. We further apply Representational Similarity Analysis (RSA) [33, 32] to the original neural activity and TE-ViDS's latent representations. Specifically, we calculate the representational similarity between the representations to each pair of trials using the Pearson correlation coefficient, yielding representational similarity matrices (RSMs). We select ten scenes that elicit the strongest average responses for visualization. By comparing the RSMs of the original

Table 2: The decoding scores (%, in 1s window) for natural movie frames on the mouse visual neural dataset. The standard error is computed on 10 runs with different random initializations.

| Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
PCA (baseline)	8.44	28.77	25.42	21.56	11.69
LFADS	8.94±0.25	26.57±2.46	26.77±2.23	24.76±1.80	12.69±1.38
pi-VAE	10.24±0.31	42.51±0.65	36.96±0.60	38.31±0.52	18.08±0.59
Swap-VAE	12.19±0.20	51.31±0.73	45.96±0.34	41.53±0.63	22.70±0.42
CEBRA	10.62±0.18	52.76±0.89	**61.01±0.76**	42.11±0.73	22.33±0.31
TE-ViDS-small	13.23±0.25	64.09±0.41	59.36±0.30	53.46±0.58	29.60±0.26
TE-ViDS	**13.88±0.19**	**65.38±0.36**	59.88±0.72	**54.33±0.54**	**30.18±0.40**
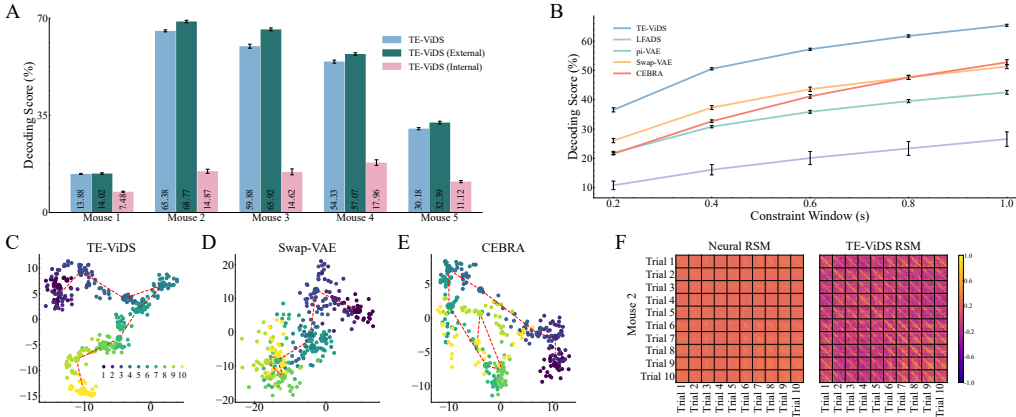
Figure 4: Results on the mouse neural dataset under natural movie stimuli. **A**. The decoding scores (%) of the *full*, *external* and *internal* latent representations of TE-ViDS. **B**. The decoding scores (%) for movie frames under different constraint windows of predicted frames and the true frames. **C-E**. Visualization results of latent trajectories (Mouse 2). Each color corresponds to all frames within 1s. Small dots denote one frame. Large dots denote the average among a group of frames. The red dashed line connects all averages. **F**. RSMs computed on the original neural representations and TE-ViDS's latent representations (Mouse 2). Each element is the similarity between two frames' representations. Each small square involves comparisons between two trials, containing 900 frames.

neural activity and TE-ViDS (Figure 3B), we show that our model can extract clearer intrinsic patterns of neural activity and reduce the impact of useless noise, facilitating the unraveling of information processing mechanisms in the visual cortex. Moreover, we observe that the RSMs of TE-ViDS between two mice exhibit significantly different patterns. For Mouse 1, there are two redder blocks, top left and bottom right, in each of the small squares, suggesting that the neural representations of closer trials in time are more similar and divided into two periods, even under different visual stimuli. This phenomenon may be due to differences in the internal states of this mouse during the two periods, and the effect of such states on neural activity is even stronger than that of visual stimuli to some extent. For Mouse 2, there is no such clear change in state. These results echo some previous studies, showing that sensory performance in mice is strongly influenced by their internal state [4], which may also be an important reason for the variability in neural activity between subjects.

In addition to tasks on "held-out" trials, we perform a more challenging task on "held-out" stimuli. All models are trained with neural responses to 95 natural scenes (approximately 80% of all scenes), and their decoding performance is evaluated on the remaining 23 scenes. The results (see Appendix H) show that TE-ViDS outperforms other models, even for these unseen stimuli.

In summary, these results demonstrate the advantages of our model in decoding natural scene stimuli and extracting fine neural dynamics (see latent trajectories over time in Appendix I). The latent representations constructed by our model help to explain potential differences in visual information processing between individuals, which deserves further exploration.
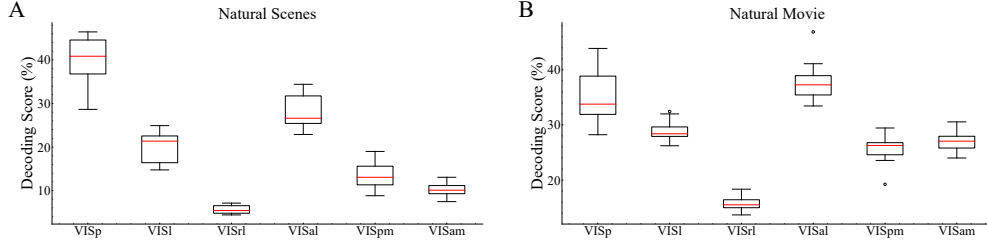
8

Figure 5: The decoding scores (%) of TE-ViDS for natural scenes/natural movie frames on six mouse visual cortical region datasets. The box plots are based on 10 runs with different random initializations.

4.6 Results on the Mouse Neural Dataset under Natural Movie Stimuli

TE-ViDS performs best for four mice (Table 2). Specifically, our model gains more advantage over LFADS than over the others. We also show that the *external* latent representations are more stimulus-relevant (Figure 4A). The results of Figure 4B show that TE-ViDS consistently outperforms alternatives across a broad range of constraint windows, especially in finer-grained frame decoding.

For visualization, we reduce the dimensions of the latent representations of each movie frame using tSNE. We set all frames within 1s as a group and show the trajectories of latent representations for the middle 10s of the movie (Figures 4C-E. We focus on Mouse 2, for which most models achieve the highest scores. See Appendix I for the results of the other parts of the movie and the other models). Compared to alternative models, the representations of TE-ViDS show clear temporal structures along movie frames with less overlap and entanglement between different time groups.

For further analysis, we also compute RSMs of the original neural activity and TE-ViDS on Mouse 2 (Figure 4F). Different from the RSMs in Figure 3B, each element in these matrices is the representational similarity between the representations to each pair of movie frames and each small square involves comparisons between two trials. The comparison between neural RSM and TE-ViDS RSM provides the same evidence that our model extracts clearer patterns and reduces noise. In each small square of TE-ViDS RSM, brighter regions are near the diagonal, showing that the neural representations of neighboring frames are more similar, even for different trials. This pattern suggests that under continuous and long-duration visual stimuli, this mouse has a stronger neural response to stimuli and is less affected by its internal state. This may reveal differences in the mouse's visual encoding ability under short-duration static stimuli and long-duration continuous stimuli.

To conclude, our model achieves the highest decoding scores for natural movie frames. Furthermore, we also perform experiments on "held-out" movie frame stimuli, in which our model performs best on decoding movie frames (Appendix H). Importantly, our model constructs meaningful latent representations related to the content and temporal structure of movie stimuli at large time scales, providing insights into visual information processing mechanisms for long-duration stimuli.

4.7 Experiments for Mouse Cortical Regions

To explore the ability of six mouse visual cortical regions to process visual stimuli, we randomly sampled 150 neurons from each of the six regions of five selected mice to construct visual neural activity datasets, respectively. We train TE-ViDS on these six datasets and evaluate its decoding performance for natural scenes/natural movie frames. The experiment is repeated ten times.

As shown in Figure 5, the decoding performance varies considerably across regions and the performance trends across regions between decoding the two types of stimuli are essentially the same. The anatomical hierarchy of the mouse visual cortex [51] shows that VISp is the primary region, while VISl, VISrl and VISal are the mid-level regions, and VISpm and VISam are the high-level regions. Additionally, VISrl is a multi-sensory region [14] that receives multi-modal sensory input. Our results show that the decoding performance is higher in the primary and mid-level regions than in the high-level regions, with the lowest performance observed in VISrl. First, the poor performance of VISrl may be due to the fact that, as a multi-sensory region, it is not well driven by visual stimuli alone. Second, while some previous studies have suggested that the mouse visual cortex is organized in a parallel structure [49, 24], the differences in decoding performance across regions may provide

evidence that there is heterogeneous and specialized visual encoding information for each brain region. These results echo the anatomical work [51] to some extent and provide new insights into the functional hierarchy of the mouse visual cortex.

5   Discussion

This work presents a novel latent variable model, TE-ViDS, by introducing temporal structures to explicitly establish temporal relationships and constructing two parts of latent representations to disentangle the stimulus-relevant components from visual neural activity. The results of synthetic and mouse neural datasets demonstrate that our model outperforms alternative models and builds latent representations that are strongly correlated with visual input, revealing intrinsic correlations between visual neural activity and visual stimuli. A series of ablation studies demonstrates the effectiveness of our model's components (Appendix J). Furthermore, TE-ViDS aids in explaining the variability in visual information processing between subjects and provides computational evidence for the functional hierarchy of the mouse visual cortex, which may contribute to the computational neuroscience community.

There is a paucity of research analyzing visual neural activity with LVMs. CEBRA has built consistent latent representations from multimodal visual neural activity [48]. Our work takes a further step in the development of powerful LVMs to yield stimulus-correlated latent representations and capture neural dynamics. However, there are still some limitations. First, in the absence of recorded internal states or behavioral information, we are unable to quantitatively assess the interpretability of *internal* latent variables and find it difficult to interpret the functional role of inferred *internal* latent representations. Second, our model exhibits substantial variability in decoding performance across individual mice. Subsequent analyses suggest that the variability in neural activity between subjects and trials [60, 38] may be caused by the animal's internal states. This could represent a potential breakthrough for LVMs to consistently construct high-quality stimulus-correlated latent representations across conditions, but further exploration is required.

Last but not least, our approach is not limited to studying visual neural spikes from mice and can be extended to neural data from other species and modalities to investigate broader principles of biological coding mechanisms, facilitating the development of computational neuroscience.

References

[1] Hamidreza Abbaspourazad, Eray Erturk, Bijan Pesaran, and Maryam M Shanechi. Dynamical flexible inference of nonlinear latent factors and structures in neural population activity. *Nature Biomedical Engineering*, 8(1):85–108, 2024.

[2] Parima Ahmadipour, Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Multimodal subspace identification for modeling discrete-continuous spiking and field potential population activity. *Journal of Neural Engineering*, 21(2):026001, 2024.

[3] Mikio C Aoi, Valerio Mante, and Jonathan W Pillow. Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. *Nature neuroscience*, 23(11):1410–1420, 2020.

[4] Zoe C Ashwood, Nicholas A Roy, Iris R Stone, International Brain Laboratory, Anne E Urai, Anne K Churchland, Alexandre Pouget, and Jonathan W Pillow. Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, 25(2):201–212, 2022.

[5] Giwon Bahg, Daniel G Evans, Matthew Galdo, and Brandon M Turner. Gaussian process linking functions for mind, brain, and behavior. *Proceedings of the National Academy of Sciences*, 117(47):29398–29406, 2020.

[6] 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. In *Advances in Neural Information Processing Systems 34*, pages 25164–25178, 2021.

[7] Mohammad Bashiri, Edgar Walker, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. In *Advances in Neural Information Processing Systems 34*, pages 15801–15815, 2021.

[8] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks, 2015.

[9] Kevin L Briggman, Henry DI Abarbanel, and WB Kristan Jr. Optical imaging of neuronal populations during decision-making. *Science*, 307(5711):896–901, 2005.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607. PMLR, 2020.

[11] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[12] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28*, 2015.

[13] 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.

[14] Saskia EJ de Vries, Jerome A Lecoq, Michael A Buice, Peter A Groblewski, Gabriel K Ocker, Michael Oliver, David Feng, Nicholas Cain, Peter Ledochowitsch, Daniel Millman, et al. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature neuroscience*, 23(1):138–151, 2020.

[15] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

[16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

[17] Changde Du, Kaicheng Fu, Jinpeng Li, and Huiguang He. Decoding visual neural representations by multimodal learning of brain-visual-linguistic features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10760–10777, 2023.

[18] Eva L Dyer, Mohammad Gheshlaghi Azar, Matthew G Perich, Hugo L Fernandes, Stephanie Naufel, Lee E Miller, and Konrad P Körding. A cryptography-based approach for movement decoding. *Nature biomedical engineering*, 1(12):967–976, 2017.

[19] Alexander S Ecker, Philipp Berens, R James Cotton, Manivannan Subramaniyan, George H Denfield, Cathryn R Cadwell, Stelios M Smirnakis, Matthias Bethge, and Andreas S Tolias. State dependence of noise correlations in macaque primary visual cortex. *Neuron*, 82(1):235–248, 2014.

[20] Otto Fabius and Joost R. van Amersfoort. Variational recurrent auto-encoders, 2015.

[21] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems 29*, 2016.

[22] Joshua Glaser, Matthew Whiteway, John P Cunningham, Liam Paninski, and Scott Linderman. Recurrent switching dynamical systems models for multiple interacting neural populations. In *Advances in Neural Information Processing Systems 33*, pages 14867–14878, 2020.

[23] Rabia Gondur, Usama Bin Sikandar, Evan Schaffer, Mikio Christian Aoi, and Stephen L Keeley. Multi-modal gaussian process variational autoencoders for neural and behavioral data. In *International Conference on Learning Representations*, 2024.

[24] Liwei Huang, Zhengyu Ma, Liutao Yu, Huihui Zhou, and Yonghong Tian. Long-range feedback spiking network captures dynamic and static representations of the visual cortex under movie stimuli. In *Advances in Neural Information Processing Systems 37*, pages 11432–11455, 2024.

[25] Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Targeted neural dynamical modeling. In *Advances in Neural Information Processing Systems 34*, pages 29379–29392, 2021.

[26] Mehrdad Jazayeri and Srdjan Ostojic. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current opinion in neurobiology*, 70:113–120, 2021.

[27] Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix Pei, Richard Gao, and Jakob H. Macke. Latent diffusion for neural spiking data, 2024.

[28] Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, 2008.

[29] Mohammad Reza Keshtkaran and Chethan Pandarinath. Enabling hyperparameter optimization in sequential autoencoders for spiking neural data. In *Advances in Neural Information Processing Systems 32*, 2019.

[30] Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19(12):1572–1577, 2022.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[32] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4, 2008.

[33] Nikolaus Kriegeskorte, Marieke Mur, Douglas A Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141, 2008.

[34] Christopher Langdon, Mikhail Genkin, and Tatiana A Engel. A unifying perspective on neural manifolds and circuits for cognition. *Nature Reviews Neuroscience*, 24(6):363–377, 2023.

[35] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 914–922, 2017.

[36] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. In *Advances in Neural Information Processing Systems 34*, pages 10587–10599, 2021.

[37] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

[38] Tyler D Marks and Michael J Goard. Stimulus-dependent representational drift in primary visual cortex. *Nature communications*, 12(1):5169, 2021.

[39] João C Marques, Meng Li, Diane Schaak, Drew N Robson, and Jennifer M Li. Internal state dynamics shape brainwide activity and foraging behaviour. *Nature*, 577(7789):239–243, 2020.

[40] Jeremiah B Palmerston and Rosa HM Chan. Latent variable models reconstruct diversity of neuronal response to drifting gratings in murine visual cortex. *IEEE Access*, 9:75741–75750, 2021.

[41] Matthijs Pals, A Erdem Sağtekin, Felix Pei, Manuel Gloeckler, and Jakob H Macke. Inferring stochastic low-rank recurrent neural networks from neural data. In *Advances in Neural Information Processing Systems 37*, pages 18225–18264, 2024.

[42] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[43] Felix Pei, Joel Ye, David Zoltowski, David Zoltowski, Anqi Wu, Raeed Chowdhury, Hansem Sohn, Joseph O' Doherty, Krishna V Shenoy, Matthew Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee Miller, Jonathan Pillow, Il Memming Park, Eva Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[44] Patrick T Sadtler, Kristin M Quick, Matthew D Golub, Steven M Chase, Stephen I Ryu, Elizabeth C Tyler-Kabara, Byron M Yu, and Aaron P Batista. Neural constraints on learning. *Nature*, 512(7515):423–426, 2014.

[45] Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021.

[46] Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Dissociative and prioritized modeling of behaviorally relevant neural dynamics using recurrent neural networks. *Nature neuroscience*, 27(10):2033–2045, 2024.

[47] Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

[48] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

[49] Jianghong Shi, Eric Shea-Brown, and Michael Buice. Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In *Advances in Neural Information Processing Systems 32*, 2019.

[50] 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, 2022.

[51] Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, 2021.

[52] Jonnathan Singh Alvarado, Jack Goffinet, Valerie Michael, William Liberti III, Jordan Hatfield, Timothy Gardner, John Pearson, and Richard Mooney. Neural dynamics underlying birdsong practice and performance. *Nature*, 599(7886):635–639, 2021.

[53] Jimmy Smith, Scott Linderman, and David Sussillo. Reverse engineering recurrent neural networks with jacobian switching linear dynamical systems. In *Advances in Neural Information Processing Systems 34*, pages 16700–16713, 2021.

[54] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. Lfads - latent factor analysis via dynamical systems, 2016.

[55] Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature neuroscience*, 25(1):11–19, 2022.

[56] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43(1):249–275, 2020.

[57] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 2056–2066, 2022.

[58] Haiguang Wen, Junxing Shi, Yizhen Zhang, Kun-Han Lu, Jiayue Cao, and Zhongming Liu. Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral cortex*, 28(12):4136–4160, 2018.

[59] Anqi Wu, Nicholas A. Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. In *Advances in Neural Information Processing Systems 30*, 2017.

[60] Ji Xia, Tyler D Marks, Michael J Goard, and Ralf Wessel. Stable representation of a naturalistic movie emerges from episodic activity with gain variability. *Nature communications*, 12(1):5170, 2021.

[61] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in Neural Information Processing Systems 21*, 2008.

[62] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316, 2017.

[63] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. In *Advances in Neural Information Processing Systems 33*, pages 7234–7247, 2020.

A Detailed Structure of TE-ViDS

The encoder and decoder of TE-ViDS are derived from Swap-VAE. The encoder consists of three blocks, the first two of which are sequentially stacked with a linear layer, a batch normalization, and a ReLU activation. The last block differs from Swap-VAE in that it additionally introduces the hidden states of the GRU as input. The output dimensions of the three blocks are N, M, and M, where N is the number of neurons and M is the number of latent variables. The decoder is a symmetric structure of the encoder, where the first two blocks are also sequentially stacked with a linear layer, a batch normalization, and a ReLU activation, and the last block is a linear layer followed by a SoftPlus activation. We set the dimensions of the two latent variables to be equal and use a one-layer GRU for each of them, where the dimensions of the hidden states are equal to the dimensions of the latent variables. The operations of each module are shown in Figure 6.
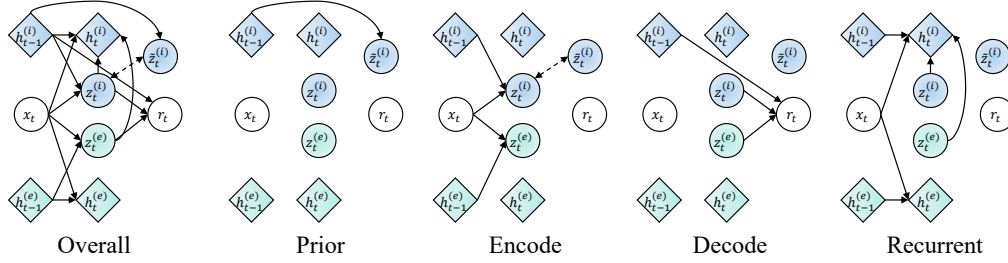

Figure 6: The operations of each module in TE-ViDS.

B Derivation of the Loss Function of TE-ViDS

The loss function of TE-ViDS obeys the strategy to maximize the likelihood of the joint sequential distribution $p(\mathbf{x}_{1:T})$. Involving the latent variables $\mathbf{z}_{1:T}$, we have the variational lower bound:

$$
\begin{aligned}
\log p(\mathbf{x}_{1:T}) &= \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log p(\mathbf{x}_{1:T}) d\mathbf{z}_{1:T} \\
&= \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T} \\
&= \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}{p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T} + \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T} \\
&= \mathrm{KL}(q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \| p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})) + \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T} \\
&\geq \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T}, \\
&\geq \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) \log \frac{p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})} d\mathbf{z}_{1:T} - \mathcal{L}_{\mathrm{contrast}},
\end{aligned}
\tag{7}
$$

where $p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})$ is the joint distribution as well as $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$ and $q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$ is the true posterior and the variational approximate posterior, respectively. The true posterior is intractable.

Considering Equations 1 an 5, we know that $\mathbf{z}_t^{(e)}$ is deterministic values and $\mathbf{h}_t^{(i)}$ is a function of $\mathbf{x}_{1:t}$ and $\mathbf{z}_{1:t}^{(i)}$. Therefore, we have the factorization:

$$
p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)}, \mathbf{z}_{1:t}^{(e)}, \mathbf{x}_{1:t-1}) p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}^{(i)}),
\tag{8}
$$

$$
q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{(i)}),
\tag{9}
$$

15

where $q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})$, $p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)})$ and $p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})$ are the distributions defined by Equations 2, 3 and 4, respectively. Based on the above factorization, we decompose the variational lower bound as:

$$
\int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})\log\frac{p(\mathbf{x}_{1:T},\mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}d\mathbf{z}_{1:T}
$$

$$
= \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})\sum_{t=1}^{T}\left(\log\frac{p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)})}{q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})}\right)d\mathbf{z}_{1:T} \qquad (10)
$$

$$
= \sum_{t=1}^{T}\left(\int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})\log\frac{p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)})}{q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})}d\mathbf{z}_{1:T}\right).
$$

When we simplify the above log-likelihood to a function $g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})$, we have:

$$
\int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})d\mathbf{z}_{1:T}
$$

$$
= \int\left(\int q(\mathbf{z}_{1:T-1}|\mathbf{x}_{1:T-1})q(\mathbf{z}_T^{(i)}|\mathbf{x}_{1:T},\mathbf{z}_{1:T-1}^{(i)})g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})d\mathbf{z}_T\right)d\mathbf{z}_{1:T-1}
$$

$$
= \int\left(q(\mathbf{z}_{1:T-1}|\mathbf{x}_{1:T-1})g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})\int q(\mathbf{z}_T^{(i)}|\mathbf{x}_{1:T},\mathbf{z}_{1:T-1}^{(i)})d\mathbf{z}_T\right)d\mathbf{z}_{1:T-1} \qquad (11)
$$

$$
= \int q(\mathbf{z}_{1:T-1}|\mathbf{x}_{1:T-1})g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})d\mathbf{z}_{1:T-1}
$$

$$
= \cdots = \int q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})g(\mathbf{x}_{1:t},\mathbf{z}_{1:t})d\mathbf{z}_{1:t}.
$$

Therefore, we further decompose Equation 10 as:

$$
\int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})\log\frac{p(\mathbf{x}_{1:T},\mathbf{z}_{1:T})}{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}d\mathbf{z}_{1:T}
$$

$$
= \sum_{t=1}^{T}\left(\int q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\log\frac{p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)})}{q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})}d\mathbf{z}_{1:t}\right)
$$

$$
= \sum_{t=1}^{T}\left(\int q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\log p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})d\mathbf{z}_{1:t}+\right.
$$

$$
\left.\int q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\log\frac{p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)})}{q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})}d\mathbf{z}_{1:t}\right)
$$

$$
= \sum_{t=1}^{T}\left(\int q(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\log p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})d\mathbf{z}_{1:t}-\right. \qquad (12)
$$

$$
\left.\int q(\mathbf{z}_{1:t-1}|\mathbf{x}_{1:t-1})\mathrm{KL}(q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})\|p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)}))d\mathbf{z}_{1:t-1}\right)
$$

$$
= \int q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})\sum_{t=1}^{T}\left(\log p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})-\right.
$$

$$
\left.\mathrm{KL}(q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})\|p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)}))\right)d\mathbf{z}_{1:T}
$$

$$
= \mathbb{E}_{q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})}\left[\sum_{t=1}^{T}\log p(\mathbf{x}_t|\mathbf{z}_{1:t}^{(i)},\mathbf{z}_{1:t}^{(e)},\mathbf{x}_{1:t-1})-\right.
$$

$$
\left.\mathrm{KL}(q(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t},\mathbf{z}_{1:t-1}^{(i)})\|p(\mathbf{z}_t^{(i)}|\mathbf{x}_{1:t-1},\mathbf{z}_{1:t-1}^{(i)}))\right].
$$

Finally, for a given sequential data x, we have the loss function:

$$\mathcal{L} \simeq \sum_{t=1}^{T} \left(\underbrace{-\log p(\mathbf{x}_t | \mathbf{z}_{1:t}^{(i)}, \mathbf{z}_{1:t}^{(e)}, \mathbf{x}_{1:t-1})}_{\text{reconstruction loss}} + \underbrace{\mathrm{KL}(q(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{(i)}) \| p(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}^{(i)}))}_{\text{regularization loss}} \right) + \mathcal{L}_{\text{contrast}},$$

(13)

where the first and second terms correspond to $\mathcal{L}_{\mathrm{P}}$ and D_{KL} in the main text, respectively.

Since we assume a Poisson distribution for the inferred neural activity, $\mathcal{L}_{\mathrm{P}}$ is the Poisson negative log-likelihood:

$$\begin{aligned}
\mathcal{L}_{\mathrm{P}}(\mathbf{x}_t, \mathbf{r}_t) &= -\log \frac{\mathbf{r}_t^{\mathbf{x}_t}}{\mathbf{x}_t!} e^{-\mathbf{r}_t} \\
&= -\mathbf{x}_t \log \mathbf{r}_t + \mathbf{r}_t + \log \mathbf{x}_t! \\
&\approx -\mathbf{x}_t \log \mathbf{r}_t + \mathbf{r}_t + \mathbf{x}_t \log \mathbf{x}_t - \mathbf{x}_t + \frac{1}{2} \log(2\pi \mathbf{x}_t).
\end{aligned}$$

(14)

As for D_{KL}, under the assumption that both the prior and the approximate posterior are Gaussian, we have:

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(\mathbf{z}_t^{(i)} \| \tilde{\mathbf{z}}_t^{(i)}) &= \int q(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{(i)}) \log \frac{q(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{(i)})}{p(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t-1}^{(i)})} d\mathbf{z}_t^{(i)} \\
&= \int q(\mathbf{z}_t^{(i)} | \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{(i)}) \log \frac{\frac{1}{\sqrt{2\pi \boldsymbol{\sigma}_{z,t}^2}} \exp\left(-\frac{\left(\mathbf{z}_t^{(i)} - \boldsymbol{\mu}_{z,t}\right)^2}{2\boldsymbol{\sigma}_{z,t}^2}\right)}{\frac{1}{\sqrt{2\pi \tilde{\boldsymbol{\sigma}}_{z,t}^2}} \exp\left(-\frac{\left(\mathbf{z}_t^{(i)} - \tilde{\boldsymbol{\mu}}_{z,t}\right)^2}{2\tilde{\boldsymbol{\sigma}}_{z,t}^2}\right)} d\mathbf{z}_t^{(i)} \\
&= -\frac{\mathbb{E}_{q(\mathbf{z}_t^{(i)})}\left[\left(\mathbf{z}_t^{(i)} - \boldsymbol{\mu}_{z,t}\right)^2\right]}{2\boldsymbol{\sigma}_{z,t}^2} + \frac{\mathbb{E}_{q(\mathbf{z}_t^{(i)})}\left[\left(\mathbf{z}_t^{(i)} - \tilde{\boldsymbol{\mu}}_{z,t}\right)^2\right]}{2\tilde{\boldsymbol{\sigma}}_{z,t}^2} - \log \boldsymbol{\sigma}_{z,t} + \log \tilde{\boldsymbol{\sigma}}_{z,t} \\
&= -\frac{1}{2} + \frac{\boldsymbol{\sigma}_{z,t}^2 + \boldsymbol{\mu}_{z,t}^2 - 2\boldsymbol{\mu}_{z,t}\tilde{\boldsymbol{\mu}}_{z,t} + \tilde{\boldsymbol{\mu}}_{z,t}^2}{2\tilde{\boldsymbol{\sigma}}_{z,t}^2} - \log \boldsymbol{\sigma}_{z,t} + \log \tilde{\boldsymbol{\sigma}}_{z,t} \\
&= \frac{1}{2}\left(-1 + \frac{(\boldsymbol{\mu}_{z,t} - \tilde{\boldsymbol{\mu}}_{z,t})^2 + \boldsymbol{\sigma}_{z,t}^2}{\tilde{\boldsymbol{\sigma}}_{z,t}^2} - \log \boldsymbol{\sigma}_{z,t}^2 + \log \tilde{\boldsymbol{\sigma}}_{z,t}^2\right).
\end{aligned}$$

(15)

Finally, we apply NT-Xent loss as the contrastive loss:

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp\left(\mathrm{sim}\left(\mathbf{z}^{(e)}, \mathbf{z}_{\text{pos}}^{(e)}\right) / \tau\right)}{\exp\left(\mathrm{sim}\left(\mathbf{z}^{(e)}, \mathbf{z}_{\text{pos}}^{(e)}\right) / \tau\right) + \sum \exp\left(\mathrm{sim}\left(\mathbf{z}^{(e)}, \mathbf{z}_{\text{neg}}^{(e)}\right) / \tau\right)},$$

(16)

where $\mathrm{sim}(*, *)$ is the cosine similarity and τ is the temperature coefficient.

C Generating Procedure of Synthetic Datasets

Non-temporal dataset. First, we generate labels u_i from four uniform distributions on $[\frac{2i \times \pi}{4}, \frac{(2i+1) \times \pi}{4}], i \in \{0, 1, 2, 3\}$, in preparation for building four clusters. Second, for each cluster, we sample 2-dimensional latent variables $\mathbf{z}$ from independent Gaussian distribution with $(5 \sin u_i, 5 \cos u_i)$ as mean and $(0.6 - 0.5|\cos u_i|, 0.5|\cos u_i|)$ as variance. Third, we feed sampled

latent variables into a RealNVP network [16] to form firing rates of 100-dimensional observations and generate the synthetic neural activity from the Poisson distribution. Each cluster contains 4,000 samples. This dataset, being label-dependent, is used to evaluate the models' ability to construct discriminative latent variables.

Temporal dataset. We generate three dynamic latent variables from the Lorenz system, consisting of a set of nonlinear equations. The firing rates of 30 simulated neurons are then computed by randomly weighted linear readouts from the Lorenz latent variables. Synthetic neural activity is also generated from the Poisson distribution. The hyperparameters of the Lorenz system follow a previous work [54]. We run the Lorenz system for 1s (1ms for a time point) from five randomly initialized conditions. Each condition contains 20 trials. This dataset is for assessing the models' ability to represent temporal information.

D Characters of Mouse Visual Neural Dataset

The full names and abbreviations of all cortical regions are listed in Table 3 and the number of neurons for all chosen mice is also presented.

Table 3: Characters of the neural dataset.

Cortical Region	Abbreviation	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
primary visual cortex	VISp	75	51	93	63	52
lateromedial area	VISl	39	30	56	38	20
rostrolateral area	VISrl	49	24	58	44	41
anterolateral area	VISal	42	51	43	71	46
posteromedial area	VISpm	62	90	17	19	64
anteromedial area	VISam	94	72	49	60	64

E Number of Trainable Parameters of All Models

The number of model parameters is roughly proportional to the number of input neurons. We present the number of parameters for the Mouse 1 dataset in Table 4.

Table 4: The number of model parameters for the Mouse 1 dataset.

	LFADS	pi-VAE	Swap-VAE	CEBRA	TE-ViDS-small	TE-ViDS
Number of parameters	0.45M	0.49M	0.38M	0.71M	0.29M	0.68M

F Training Setup

We list the information of training datasets and the hyperparameters of the model training in Table 5. For each model, we perform the grid search for the learning rate (1.0×10^{-5} to 1.0×10^{-3}) and the weights of each loss term (0.01 to 10) to achieve optimal performance on the validation set.

Table 5: The information of each dataset and the hyperparameters of the training.

Dataset	Training Size	Validation Size	Test Size	Latent Dimension	Learning Rate	Optimizer
non-temporal synthetic dataset	12800	—	3200	32	0.0005	Adam
temporal synthetic dataset	80000	—	20000	8	0.001	Adam
visual neural dataset under natural scenes	118000	14750	14750	128	0.0001	Adam
visual neural dataset under natural movie	28800	3600	3600	128	0.0001	Adam

Furthermore, we present the preprocessing implementation of training samples for all models.

Visual neural dataset. For the neural activity under natural scenes, each sample input to TE-ViDS is sequential data from 5 time points and the offset of positive samples from target samples is within ± 3

18

time points. For the neural activity under the natural movie, each sample consists of 4 time points and the maximum absolute offset is 2. As for alternative models, pi-VAE and Swap-VAE take the neural activity of an independent time point as an input sample. LFADS processes samples with the same length as our model. For CEBRA, following the original approach [48], we take the surrounding points centered on the target point to form a sequence (5 time points for natural scenes and 4 for the natural movie).

Non-temporal dataset. Since this dataset does not involve temporal relationships, each sample can be conceptualized as a sequence comprising a single time step. In the context of contrastive learning for such data, the approach of defining positive pairs through temporal shifting is inapplicable. Instead, we adopt a strategy provided in CEBRA, selecting positive samples based on the distance of their respective labels, thereby implementing a form of supervised contrastive learning.

Temporal dataset. TE-ViDS and LFADS use 50ms of data as input, while the other models take data at one time point. The offset of positive samples is set to 5ms for our model.

All models for all datasets are trained on 1 GPU (NVIDIA A100).

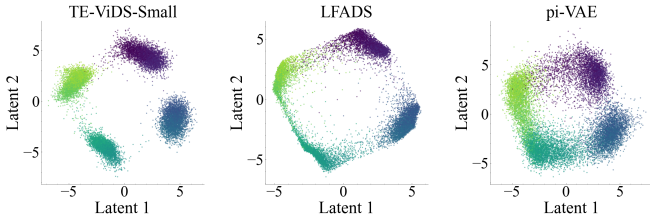# G Additional Results of Alternatives on Synthetic Non-Temporal Dataset

Figure 7: The inferred latent variables of alternatives on the non-temporal dataset.

H Additional Experiment on the Visual Neural Datatsets

We train models with neural activity under approximately 80% natural scene/movie stimuli and evaluate them using "held-out" stimuli, i.e., completely unseen scenes/movies. As Tables 6 and 7 show, TE-ViDS consistently achieves the best performance on such challenging tasks.

Table 6: The decoding scores (%) for "held-out" natural scenes.

Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
LFADS	48.96±1.13	34.78±1.83	39.22±2.47	38.17±1.60	15.74±1.14
pi-VAE	23.57±3.17	37.04±2.42	41.91±1.90	20.70±2.05	10.35±1.10
Swap-VAE	55.48±2.34	17.65±1.94	36.61±3.21	44.09±2.39	14.17±1.81
CEBRA	5.83±0.58	8.87±0.80	13.04±1.40	9.39±0.77	4.96±0.69
TE-ViDS-small	62.61±2.50	**41.30±1.72**	**47.57±3.06**	50.52±2.77	23.30±1.20
TE-ViDS	**65.91±1.31**	38.70±1.71	46.52±2.09	**53.57±1.88**	**24.52±1.17**

Table 7: The decoding scores (%, in 1s window) for "held-out" natural movie frames.

Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
LFADS	38.89±0.98	51.78±1.28	49.06±2.04	46.28±1.89	39.22±1.42
pi-VAE	38.89±1.34	65.39±0.95	66.56±1.31	59.11±1.34	46.17±1.13
Swap-VAE	36.22±1.83	59.72±1.35	58.33±1.62	57.72±1.77	44.50±1.19
CEBRA	37.44±1.04	54.39±1.41	52.89±0.84	48.89±0.86	37.89±1.60
TE-ViDS-small	39.28±2.15	**68.06±1.09**	**72.50±0.81**	65.50±0.78	**47.83±1.47**
TE-ViDS	**42.00±0.52**	66.28±1.07	69.06±0.84	**66.28±1.43**	47.00±1.80

I Additional Results of Latent Trajectories on the Visual Neural Datatsets

For the mouse neural dataset under natural scene stimuli, we visualize the latent representations by embedding them in two dimensions using tSNE (Figure 8). We focus on Mouse 1, for which most of the models achieve the highest scores. We select ten scenes that elicit the strongest average responses for visualization. Specifically, we reduce the dimensions of latent representations at a single time point and take the average across all trials, to show the latent trajectories over time for each scene. The result of TE-ViDS exhibits a clear temporal structure for different natural scenes. For LFADS, its ability to encode temporal features of sequential neural activity results in clear temporal structures, but the latent representations of different classes are largely intermingled. For pi-VAE, Swap-VAE, and CEBRA, their latent trajectories show varying degrees of entanglement over time. These results suggest that our model effectively distinguishes between category information and captures temporal information from neural dynamics well.

For the mouse neural dataset under natural movie stimuli, in addition to Figures 4C-E, we visualize the results of all models and all three parts of the movie for Mouse 2 (Figure 9).
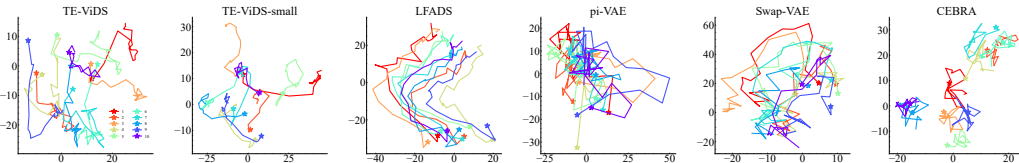

Figure 8: Visualization results of latent trajectories on the mouse visual neural dataset under natural scene stimuli (Mouse 1).
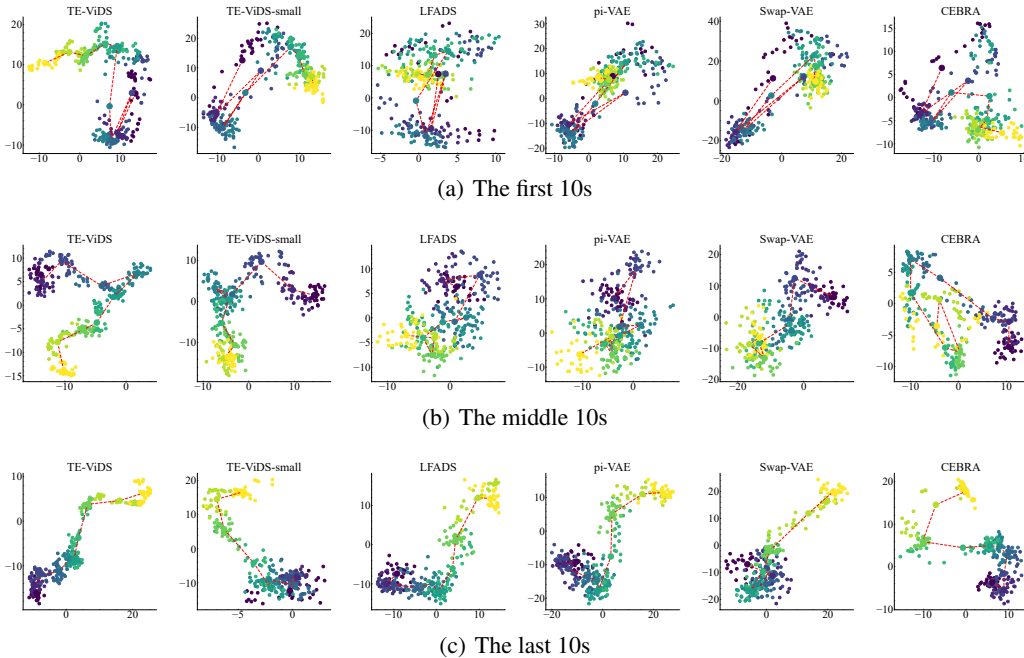

(a) The first 10s

(b) The middle 10s

(c) The last 10s

Figure 9: Visualization results of latent trajectories on the mouse visual neural dataset under natural movie stimuli (Mouse 2).

J Ablation Studies

We perform ablation studies for several aspects of TE-ViDS's components and neural activity input dimensions to explore their impact on performance.

The loss function and the recurrent module of TE-ViDS. To show the effectiveness of the components of our model, we conduct some ablation studies on the loss function and the recurrent

Table 8: The decoding scores (%) of ablation studies for the loss function, the recurrent module and the split design of TE-ViDS on the mouse visual neural dataset under natural scene stimuli.

Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
TE-ViDS	**50.86±0.81**	**27.24±0.47**	**29.90±0.43**	**38.05±0.53**	**9.44±0.20**
w/o negative samples	21.85±1.86	9.71±0.72	12.32±1.22	20.03±1.26	6.20±0.43
w/o contrastive loss	30.17±1.74	8.24±0.33	13.76±0.87	23.42±0.79	7.19±0.28
w/o swap operation	28.80±0.86	14.27±0.42	18.92±0.93	15.97±0.85	4.68±0.24
w/o contrastive loss and swap operation	6.88±0.64	3.92±0.22	4.17±0.72	6.58±0.32	2.56±0.12
with temporal independent prior	28.97±1.16	9.05±0.64	15.10±0.98	15.54±0.69	3.97±0.32
GRU→Vanilla RNN	47.64±0.78	25.61±0.68	28.58±0.66	34.15±0.43	7.17±0.30
GRU→LSTM	49.97±0.50	26.92±0.59	29.31±0.23	36.73±0.41	9.24±0.27
Non-recurrent	33.32±0.99	21.78±0.37	22.63±0.47	27.27±0.64	7.69±0.40
External-Only	45.41±1.16	21.49±0.46	23.83±0.73	27.39±0.67	6.80±0.23
Internal-Only	2.10±0.21	1.78±0.19	1.44±0.16	2.49±0.22	1.46±0.12

Table 9: The decoding scores (%, in 1s window) of ablation studies for the loss function, the recurrent module and the split design of TE-ViDS on the mouse visual neural dataset under natural movie stimuli.

Models	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5
TE-ViDS	**13.88±0.19**	**65.38±0.36**	59.88±0.72	54.33±0.54	30.18±0.40
w/o negative samples	11.27±0.36	49.59±1.18	45.67±0.60	44.17±0.42	19.93±0.35
w/o contrastive loss	11.24±0.23	47.98±0.90	44.09±0.67	44.02±0.47	18.24±0.41
w/o swap operation	10.22±0.31	49.39±0.57	45.30±0.41	43.12±0.57	21.83±0.39
w/o contrastive loss and swap operation	9.16±0.37	24.84±1.10	22.33±0.81	26.49±0.66	12.02±0.49
with temporal independent prior	12.09±0.16	57.74±0.62	53.87±0.49	46.90±0.48	22.92±0.43
GRU→Vanilla RNN	13.08±0.31	63.19±0.55	59.13±0.42	53.83±0.41	29.62±0.40
GRU→LSTM	12.77±0.25	64.69±0.53	**60.00±0.60**	**54.37±0.41**	28.50±0.61
Non-recurrent	11.14±0.30	53.26±0.48	48.31±0.47	43.81±0.40	22.98±0.44
External-Only	12.16±0.24	63.33±0.29	58.64±0.37	52.11±0.39	**30.24±0.54**
Internal-Only	7.57±0.24	11.73±0.43	12.86±0.35	16.10±0.33	9.89±0.22

module (Tables 8 and 9). In terms of contrastive learning, we first exclude negative samples from the computation of the contrastive loss and use only the cosine distance between the *external* latent variables of positive sample pairs as the loss function. In other words, we only bring the positive pairs closer. This results in a decrease in performance, suggesting that negative samples are useful. Next, we remove either the contrastive loss or the swap operation and observe a similar impact for both. Finally, we remove both, meaning that there are no more objectives related to contrastive learning. The significant decrease in performance suggests that contrastive learning plays a crucial role in our models. In terms of the regular loss of the *internal* latent variables, we originally assumed that the prior distribution is time-dependent. When we assume that it is an independent standard normal distribution at each time step, the model's performance degrades, demonstrating that time-dependent assumptions about the prior are also important. In terms of the recurrent module, the results suggest that GRU is a better choice when considering the trade-off between performance and computational efficiency. Besides, we evaluate a non-recurrent version of our model by setting the time steps of GRU to 1, which demonstrates that the recurrent module plays a critical role.

The split design of latent variables. We build two models based on TE-ViDS, one with *external* variables only (External-Only) and the other with *internal* variables only (Internal-Only). Both models perform worse than TE-ViDS, confirming the split design's importance and effectiveness.

The dimension of latent variables and the number of input neurons. We perform ablation studies on the number of latent variables and input neurons. As shown in Figures 10 and 11, first, the performance saturates gradually as the dimension of the latent variables increases, which suggests that it is sufficient to choose a dimension in a reasonable range (not much fewer than input neurons). Second, performance decreases as the number of sampled neurons decreases, suggesting that for each mouse, all recorded neurons contribute to the representation of visual stimuli.
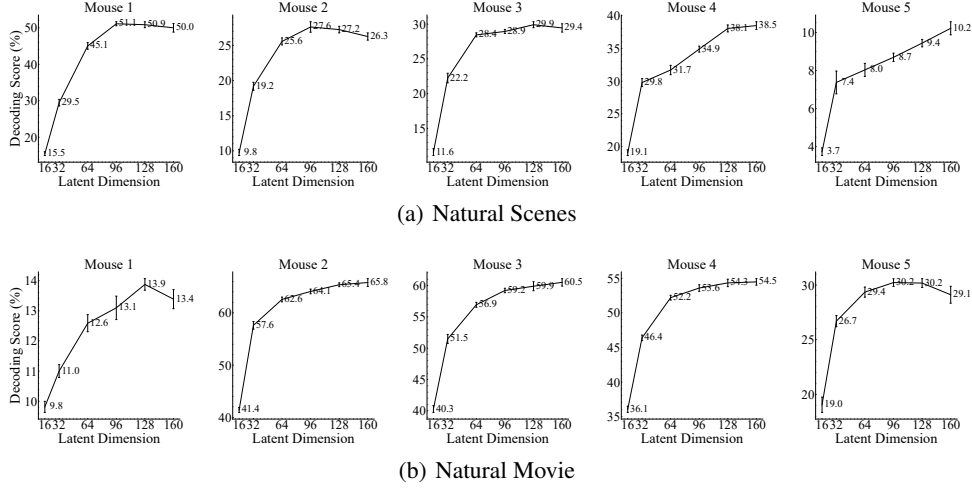
(a) Natural Scenes

(b) Natural Movie

Figure 10: The results of ablation studies on the dimension of latent variables.

(a) Natural Scenes
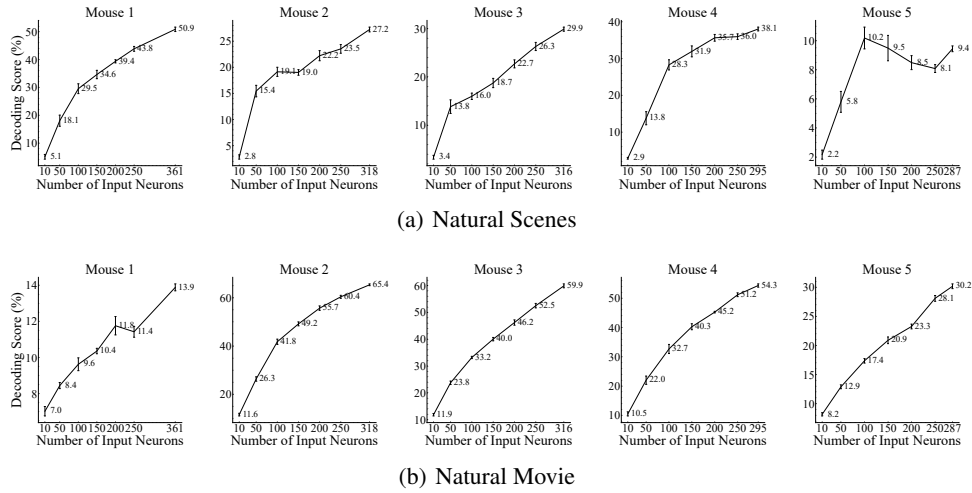

(b) Natural Movie

Figure 11: The results of ablation studies on the number of input neurons.

K   Additional Experiment on the Mouse Visual Neural Dataset from CEBRA

The neural dataset used in CEBRA is also preprocessed from the Allen Brain Observatory Visual Coding dataset. The dataset is generalized by sampling different numbers of neurons from the same visual cortical region of all mice, without taking into account the variability between subjects. We evaluate our model on this dataset. The results (Figure 12) show that our model performs better on the primary and mid-level regions, while CEBRA performs better on the high-level regions. Moreover, in the case of a sample with 40 time steps (CEBRA reported their highest performance in this setting), our model can achieve a performance of more than 95% with fewer trainable parameters (TE-ViDS: 0.44M; CEBRA: 1.09M).
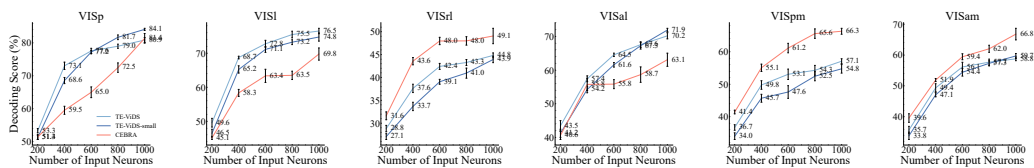

Figure 12: The decoding scores (%, in 1s window) on the mouse neural dataset used in CEBRA.

22

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see abstract and introduction.

   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The work does not involve proof of theory.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We detail all the techniques for reproducing the results of our work in Section 3 and Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The codes and data are available at *https://github.com/Grasshlw/Time-Evolving-Visual-Dynamical-System*.

   Guidelines:
   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please see Section 4 and Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Please see the results in all Tables and Figures.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Please see Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our work conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our work does not involve social impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not release data or models with a high risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets used in our paper are publicly available.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our work does not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing experiments nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The LLM is used only for writing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.