Switching Subspace Model for Neural Population Analysis

Mithilesh Vaidya
Georgia Institute of Technology
mithilesh.vaidya@gatech.edu

Chris Rodgers
Emory University
christopher.rodgers@emory.edu

Anqi Wu
Georgia Institute of Technology
anqiwu@gatech.edu

Abstract

Neural population data usually contains a mixture of various signals distributed across both neurons and time. The goal of this work is to develop a latent variable model that disentangles such signals and gives an interpretable single-trial dynamic. We hypothesize that the dynamic for each signal evolves separately throughout the trial and the observed neural activity is a linear combination of subspaces corresponding to these signals. Priors such as Unimodal GP and supervision using a simple convolutional neural network demonstrate interpretable single-trial results, good data explanation performance, and high decoding accuracy for task-relevant behavioral signals.

1 Introduction

With advancements in neural recording equipment, it is now possible to simultaneously measure the neural spike data of populations of neurons. This enables us to measure the neural activity of subjects carrying out a wide variety of complex tasks such as decision making [Harvey et al., 2012, Rodgers, 2022b, Luo et al., 2023] and motor tasks [Yu et al., 2008, Pandarinath et al., 2018]. Given high-dimensional neural population data, researchers have developed latent variable models to map such data to low-dimensional representations for better interpretation [Pandarinath et al., 2018, Wu et al., 2017]. Among these methods, switching linear dynamic systems (SLDS) are a popular class of models [Zoltowski et al., 2020, Linderman et al., 2016, 2017]. These models employ a piece-wise linear model to capture the non-linear neural dynamics in the low-dimensional space. Each linear model is a linear dynamic and the dynamic switches over time.

In this paper, we make a substantially different assumption from SLDS: in many neural datasets, what switches is not the dynamic but instead the neural subspace. This means there are multiple subspaces underlying the same neural population, each potentially encoding different types of information such as stimulus, choice, or motor actions [Kobak et al., 2016, Aoi and Pillow, 2018, Stringer et al., 2019, Hasnain et al., 2023]. We propose that multiple subspaces can be active simultaneously, but their activation may switch from one to another over time within the same trial. Such switches can differ across trials. This assumption is supported by many existing works [Hasnain et al., 2023, Luo et al., 2023].

Therefore, in this paper, (1) We introduce a novel switching subspace model, based on the assumption that neural data consists of multiple subspaces, each encoding unique types of information like stimulus and choice. The neural data is a combination of these subspaces, with varying contributions of each subspace over time. We can identify these switching subspaces on a trial-by-trial basis. (2) To

Preprint. Under review.

identify the subspaces encoding different experimental variables, we incorporate these variables into the subspace model as supervision. Similar models [Aoi and Pillow, 2018, Kobak et al., 2016] have used such supervision to learn subspaces, but they assume a linear summation of all subspace signals without differentiating their contributions at different time points, thus lacking switching information. (3) Another contribution of our work is moving beyond the use of a hidden Markov model (HMM) to model switching, as used in SLDS. This is based on the observation that some subspace signals may be transient while others may persist. Since multiple subspaces can contribute simultaneously, it might be suboptimal to assume only one subspace is active at a time (like HMM or SLDS models). Thus, we introduce a soft subspace indicator, where each subspace has a continuous subspace indicator with values between 0 and 1 (termed assignment latent). If two subspaces coexist, both subspace indicators will be non-zero, providing greater flexibility compared to the hard switching model used in SLDS, which assumes only one dynamic is active at a time. In our model, multiple subspaces can be active simultaneously, each with its own dynamic corresponding to specific experimental variables such as stimulus or decision-making. Therefore, we are indirectly capturing a mixture of dynamics. (4) Finally, we regularize the subspace indicators with smoothness and unimodality assumptions. The smoothness assumption is based on the expectation that subspace transitions should be gradual. It is unlikely that the stimulus subspace would dominate at one instant, choice subspace would dominate the next, and suddenly revert back to the stimulus subspace immediately afterward. The unimodality assumption applies primarily to the choice subspace indicator. The rationale is that as evidence accumulates and saturates, it typically leads to a steady decision state, as modeled by the drift-diffusion model [Luo et al., 2023, Zoltowski et al., 2020]. Consequently, the choice assignment latent across time is expected to be unimodal, with the peak occurring toward the end of the trial. These assumptions are incorporated as priors over the assignment latent, serving as soft regularizers. They help guide the learning of the assignment latent without imposing overly strict constraints.

We apply our proposed switching subspace model to a somatosensory cortex dataset [Rodgers, 2022b]. This dataset consists of neural spike recordings from rodents actively moving their whiskers to discriminate between convex and concave shapes, termed *stimulus*. They are trained to lick either the left or right pipe based on the shape of the stimulus, with their decision termed *choice*. Previous analyses have shown that somatosensory neurons encode both stimulus and choice information with a timing difference, where stimulus encoding precedes choice encoding, an observation found in many brain areas [Rodgers, 2022b, Luo et al., 2023]. We demonstrate that our method better explains the data, compared to SLDS. We present the stimulus and choice subspaces and the subspace assignment latents for single trials, showing that these discovered subspaces and assignments are interpretable, given the whisking decision-making task.

2 Method

2.1 Switching subspace model

For a single trial, let $Y \in \mathbb{R}^{T \times N}$ be the observed spikes of N neurons across T time bins, and y_t denote the slice of Y at time t. We build a variational autoencoder (VAE) [Kingma and Welling, 2013], consisting of a decoder, an encoder, and a prior to model the neural data. Refer to Figure 1 for the entire architecture.

Data generation from the latent space (decoder): We model the observations as a linear combination of $K(\ll N)$ subspaces. Each subspace corresponds to a continuous latent $x_t^i \in \mathcal{R}^{m^i}$ which evolves across time. Here, m^i is the dimensionality of the i^{th} subspace and $M = \sum_{i=1}^{K} m^i$. The subspace mapping function is $f_{\theta_i}^i(\cdot) : \mathbb{R}^{m_i} \to \mathbb{R}^N$ parameterized by θ_i. The contribution of each subspace to y_t is modeled by the assignment latent $z_t^i \in [0, 1]$. We impose $\sum_{i=1}^{K} z_t^i = 1$ for all t. We achieve these constraints on z using a softmax function, resulting in an output where the values are between 0 and 1 and sum to 1. In models such as HMM or SLDS, $z^i \in \{0, 1\}$, while we allow z^i to be continuous so as to allow multiple signals to coexist in time. The generative model for the firing rate $\hat{y}_t$ is defined as $\hat{y}_t = \sum_{i=1}^{K} z_t^i f_{\theta_i}^i(x_t^i)$. The neural spikes are generated from $y_t \sim \text{Poisson}(g(\hat{y}_t))$ where g is a simple non-negative function such as Softplus.

Let $Z_t \in \mathbb{R}^K$ denote the concatenation of all z_t^i and $X_t \in \mathbb{R}^M$ denote the concatenation of all x_t^i. We want to infer the latents $X = [X_1, X_2, ..., X_T]^\top \in \mathcal{R}^{T \times M}$, $Z = [Z_1, Z_2, ..., Z_T]^\top \in \mathcal{R}^{T \times K}$ on a trial-by-trial basis using Y.

Posterior (encoder): We encode the posterior $q_\phi(X, Z|Y)$ using a bidirectional GRU [Chung et al., 2014]. One can alternatively use LSTMs, Transformers or any flexible architecture but GRU suffices for our setting. Here, ϕ denotes the parameters of the GRU. Following standard VAE-based approaches, we parameterize the mean and covariance of X and Z and draw samples from the distribution using the reparameterization trick. Both posteriors are assumed to be diagonal. We can optionally increase complexity by introducing additional variational parameters to capture off-diagonal entries. The posterior is computed as follows:

$$q_\phi(X, Z|Y) = \underbrace{\Pi_{t=1}^{T} \mathcal{N}(\mu_{x,t}, \Sigma_{x,t})}_{\text{Posterior at each time for } x} \cdot \underbrace{\Pi_{i=1}^{K} \mathcal{N}(\mu_z^i, \Sigma_z^i)}_{\text{Posterior for each } z \text{ across time}} \tag{1}$$

where $\mu_{x,t} = [\mu_{x,t}^1, \mu_{x,t}^2, ..., \mu_{x,t}^M] \in \mathcal{R}^M, \mu_z^i = [\mu_{z,1}^i, \mu_{z,2}^i, ..., \mu_{z,T}^i] \in \mathcal{R}^T$ and $\Sigma_{x,t} \in \mathcal{R}^{M \times M}$, $\Sigma_z^i \in \mathcal{R}^{T \times T}$. Denote the output of the GRU at time t by o_t. We train separate MLPs on top of the GRU output in order to extract the variational parameters:

$$[\mu_{z,t}^i, \sigma_{z,t}^i] = \text{MLP1}^i(o_t); \quad \Sigma_z^i = [\text{diag}(\sigma_{z,t}^i)]_{T \times T} \tag{2}$$

$$[\mu_{x,t}^i, \sigma_{x,t}^i] = \text{MLP2}^i(o_t); \quad \Sigma_{x,t} = [\text{diag}(\sigma_{x,t}^i)]_{M \times M} \tag{3}$$

Since o_t has access to the entire trial, it can jointly model both x and z across time and latent dimensions.

Prior: We impose GP priors over time for all z^i in order to ensure smoothness. This is crucial because our interpretation of assignment latent is that it captures the *phase* of the experiment e.g. stimulus collection or decision making. These cannot evolve rapidly. This method of smoothing the latent in a sequential VAE model has been previously explored in Fortuin et al. [2020]. We do not impose any such smoothing over x^i so that it can capture rapid local variations in the observations. Formally, the prior over X and Z is as follows:

$$P(X, Z) = \underbrace{\Pi_{t=1}^{T} \mathcal{N}(0_M, I_M)}_{\text{Identity prior over X}} \cdot \underbrace{\Pi_{i=1}^{K} \mathcal{GP}(0_T, \mathcal{K})}_{\text{GP prior over Z}} \tag{4}$$

where $\mathcal{K} \in \mathcal{R}^{T \times T}$ is the RBF kernel with length scale ℓ, scaling factor σ_f, and noise variance σ_n^2. Each element in $\mathcal{K}$ is defined as $k(m, n) = \sigma_f^2 \exp\left(-\frac{\|m-n\|^2}{2\ell^2}\right) + \sigma_n^2$.

Loss: We maximize the standard ELBO in VAE to learn the parameters ϕ and $\Theta = \cup_{i=1}^{K} \{\theta_i\}$:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi(X,Z|Y)} [\log p_\Theta(Y|X, Z)] - \text{KL}(q_\phi(X, Z|Y) \parallel p(X, Z)). \tag{5}$$

2.2 Unimodality prior

We found that the above model overfits to the data in order to maximize data reconstruction performance despite the GP priors over the latent z. This hurts single-trial interpretability. An example of this is given in Column 1 of Figure 2C. We observe that the z, despite weak GP smoothing, can rapidly change over time. Moreover, it can peak multiple times over the course of a trial. Therefore, there is a necessity to impose some additional prior assumptions to regularize the latent z.

Our goal is to capture the high-level *phase* of the experiment using z. As a concrete example, consider the somatosensory dataset. Each trial begins with some task-irrelevant activity. As the shape slowly moves into the vicinity of the whiskers, the subject begins to whisk and gather sensory information. Once enough contacts have been made, the choice has been made. Say we model this experiment using 3 subspaces, with their assignment latents z^1, z^2, z^3 corresponding to the stimulus subspace, choice subspace, and the null subspace (not encoding stimulus nor choice). In this case, we want z^3 to be high in the beginning. Then, z^1 should ramp up as contacts are made and either saturate if contacts persist or die down if the decision dominates the activity. z^2 should ramp up at the end and persist, as often modeled by the classic drift-diffusion model for decision-making. Given such an intuition, we want z^2 to be unimodal.

We impose this unimodal prior using a penalty term inspired by Andersen et al. [2017]. We briefly review the idea in this section and refer to the cited paper for more details. Suppose we aim to impose

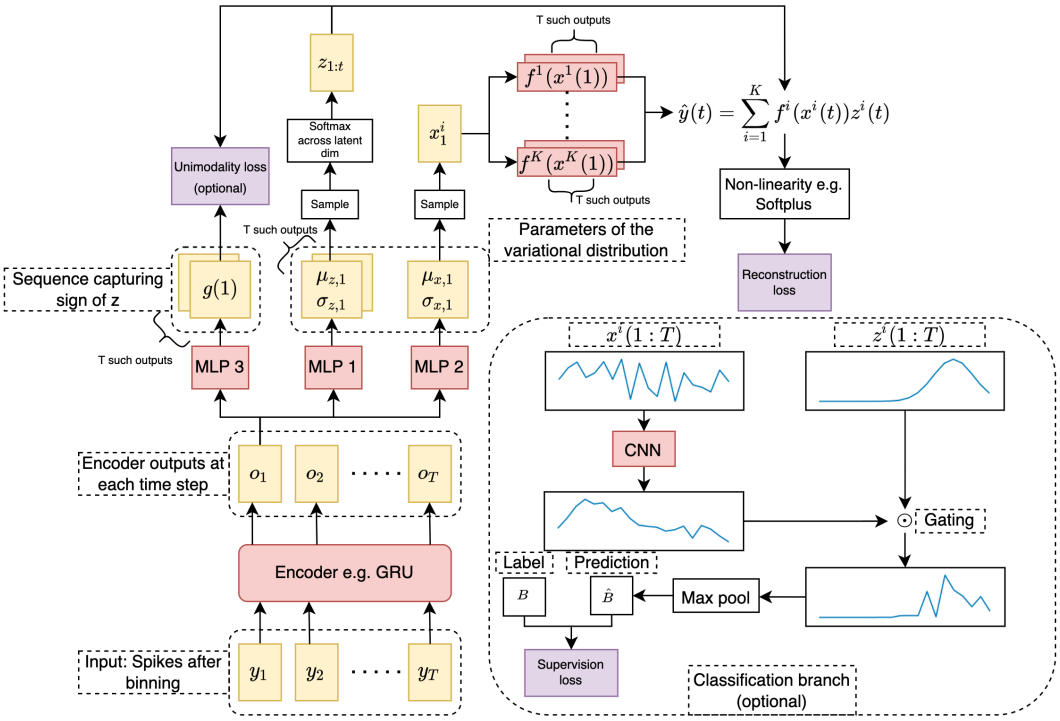

Figure 1: Architecture of the model described in Section 2. The dotted box in the bottom right corner is an example of a CNN which can be used for decoding any behavioral variable of interest using a combination of the latents x^i, z^i.

unimodality over a GP function f with derivative f'. We introduce another sequence g which models the sign of f'. Let $\Phi(x)$ denote the cdf of a standard Normal distribution. Then, maximizing the following term penalizes any shape of f which deviates from the unimodal assumption:

$$L_U = \prod_{t=1}^{T} \left[\underbrace{\Phi(-\nu_f f'_t)\Phi(-\nu_g g_t)}_{\text{f is decreasing when g < 0}} + \underbrace{\Phi(\nu_f f'_t)\Phi(\nu_g g_t)}_{\text{f is increasing when g > 0}} \right] \prod_{t=1}^{T} \underbrace{\Phi(-\nu_g g'_t)}_{\text{Ensures g is non-increasing}} \tag{6}$$

Here, ν_f and ν_g are scaling parameters which control the sharpness of the GPs. To adapt it to our model, we replace f in Eq. 6 with z^i for the i^{th} subspace. We also introduce a new latent variable g according to Eq. 6. Therefore, we need to augment the latent variable set into $\{x, z, g\}$. We parameterize g_t using a simple linear mapping from the output of the GRU (MLP3 in Figure 1) i.e. $g_t = G \cdot o_t + d$ where o_t is the GRU output at time t and G, d are jointly learned with the rest of the network. We have one set of G, d, and g for every unimodal latent.

2.3 Supervision

Neural population encodes a multitude of signals. As an example: in the somatosensory cortex, both choice and stimulus are encoded. However, poor decoding of stimulus using top PCA components reveals that stimulus information is encoded very weakly. Fitting the above models using ELBO would lead to the elimination of stimulus information. To overcome this, we propose the inclusion of a lightweight classifier into our model. By decoding the variable of interest (using B to indicate stimulus or choice for the somatosensory dataset) from our latents X, Z, we can preserve it in the latent space.

The design of the classifier and the loss function $\mathcal{L}_C$ depends on the nature of B. In order to preserve a trial-level variable such as stimulus or choice, we pass the corresponding latent x^i through a convolutional neural network (CNN), gate the output of the CNN by z^i, and mean pool it across time to output the final prediction $\hat{B}$. We minimize the standard Binary Cross Entropy loss $\mathcal{L}_B$ between B and $\hat{B}$ to learn a good X, Z. The exact operation is depicted in Figure 1.

Gating by z^i is important because it allows the CNN to decode from only the portion of the trial which is relevant for the corresponding behavioral variable e.g. choice appears later on in the trial

after enough whisking. The x^i in the first few time bins will be very noisy, thereby making it difficult for the CNN to learn. If z^i indeed captures choice, it will be low early on and hence such x^i will not hamper the learning dynamics of the CNN.

The final loss, integrating all three terms, can be written as:

$$\mathcal{L} = -\mathcal{L}_{ELBO} - c_U \log(\mathcal{L}_U) + c_B \mathcal{L}_B \tag{7}$$

where c_U and c_B are hyperparameters which control the contribution of each term.

3 Experimental Setup

3.1 Dataset

We focus our experiments on the somatosensory dataset by Rodgers [2022a,b]. The data consists of 161 trials and 35 recorded neurons. We focus on a 2.5 second window from -2s to 0.5s where t = 0 refers to the opening of the response window. We bin the neural responses in 100 ms bins. Both these choices are in line with the exploratory work in Nogueira et al. [2023]. Hence, each observation Y is of shape 25 (time bins) x 35 (number of neurons). We split the trials into train-test subset in a 80-20 ratio. We ensure that no subset has a lopsided distribution of either stimulus or choice.

We flatten each trial across time and neurons to train a simple SVM classifier to obtain a baseline number for stimulus and choice decoding from raw spikes. We use L2 regularisation since feature dimension (25 bins * 14 neurons = 350) > number of trials (161). Choice can be correctly decoded in 88% of the test trials while stimulus accuracy on the test set is lower, at 72%. We chose to use SVM instead of CNN here because SVM is simpler to train and provides a basic understanding of the level of stimulus and choice information encoded in this neural population. However, integrating SVM into our end-to-end framework is challenging. Therefore, for easier training and evaluation of the latent variables, we use a CNN instead, which is more flexible.

3.2 Evaluation metrics

We evaluate the effectiveness of our model using the following metrics:

- Bits/Spike [Pei et al., 2021]: it measures the data reconstruction performance of the model. The observation model is a Poisson distribution. Higher is better.
- Behavior Decoding: To gauge the effectiveness of latents z^i, x^i in capturing the desired behavioral variables, we decode stimulus and choice from the x^i/z^i pair using a separate CNN. Refer to Section A.1 for more details.

Apart from the numbers above, we also plot both single-trial and trial-averaged latents in order to visualize and understand their behavior. An interpretable latent is as important as the above metrics to understand the underlying sensorimotor strategy.

3.3 Setup

A 2-layer, 8-unit bidirectional GRU was found to be sufficient for encoding the posterior. We chose a 1-d subspace for each of the 3 latents z^1, z^2, z^3 i.e. $m_1 = m_2 = m_3 = 1$ since increasing the size of the subspace leads to marginal gain in the reconstruction performance. For the GP prior over each assignment latent z^i, we use a length scale $l = 3$, scale factor $\sigma_f^2 = 0.5$, and noise variance $\sigma_n^2 = 0.01$.

The mapping from latent to observation $f^i(\cdot)$ was chosen to be a linear mapping for all 3 latents since it has few parameters and gives interpretable results [Zoltowski et al., 2020, Luo et al., 2023].

MLP1 and MLP2, which map the output o_t of the GRU to the variational parameters, consist of a single hidden layer with 8 units and a ReLU non-linearity.

For unimodality, $\nu_g = \nu_f = 1$. Also, we impose it only on z^2, which corresponds to choice, because we know from previous analysis [Rodgers, 2022b] that choice information persists even without contact. Stimulus, on the other hand, is transient, and hence z^1 need not be unimodal.

Table 1: Evaluation metrics for 4 variants of our model. Decoding performance is obtained by training a separate CNN for each model on the latent space z^i, x^i and we report results on the **test** set (Check Section A.1 for exact methodology). Since M1 and M2 have no supervision, we report the best result on using any one of the x^i/z^i pairs. In M3 and M4, stimulus and choice supervision is added on top of latents z^1, x^1 and z^2, x^2 resp. and hence results are reported using these latents. Terms in brackets indicate the latent which gave the best performance.

Model	Uni.	CNN	Bits/Spike (Train)	Bits/Spike (Test)	Stimulus Decoding (%)	Choice Decoding (%)
M1	✗	✗	0.59	**0.58**	42.4 (x^1, z^1)	66.6 (x^1, z^1)
M2	✓	✗	0.59	0.57	51.5 (x^1, z^1)	72.7 (x^2, z^2)
M3	✗	✓	0.55	0.53	72.7 (x^1, z^1)	81.8 (x^2, z^2)
M4	✓	✓	0.57	0.55	**87.8** (x^1, z^1)	**93.9** (x^2, z^2)

For the CNN in Section 2.3, we use a 1-layer 8-channel CNN with a kernel of length 5 and padding of length 2 to preserve the shape of the output across time, thereby allowing gating by z. Separate CNNs are trained for stimulus and choice. The same architecture is used for training a CNN during evaluation.

The coefficients of loss terms are set to the following: $c_U = 10$ and $c_B = 10$ for choice while $c_B = 20$ for stimulus since stimulus is harder to extract from the data.

The entire network was trained end-to-end using the Adam optimizer [Kingma and Ba, 2014], with a learning rate of 0.01, on CPU. Since our datasets and models are small, we do not need a GPU.

3.4 Baseline

SLDS is the baseline model we compare against in this work.[1] We fit parameters for a model with 3 discrete states and 3 continuous latents. For a fair comparison, we report results on also including stimulus and/or choice as input to the SLDS model. We input the stimulus throughout the trial and input the choice when the response window opens. We use the Laplace EM method [Zoltowski et al., 2020] to do the inference and train for 20 iterations to ensure that the optimization converges.

4 Results

We first discuss the behavioral decoding and spike reconstruction performance of our proposed switching subspace model. Then, we visualize both trial-averaged and single-trial results for our model. Finally, we compare the results with SLDS.

4.1 Reconstruction and decoding performance

First, we separately study the effectiveness of the two orthogonal contributions: CNN and unimodality. Table 1 contains the reconstruction performance for each variant of the model. As expected, enforcing unimodality or adding a CNN degrades reconstruction performance but the drop is minimal (5% drop for the model with both unimodality and CNN). Stimulus and choice decoding performance without unimodality or CNN already match that of SVM. However, the single-trial results for this model are not interpretable (as discussed in the subsequent section). Adding unimodality or a CNN enhances decoding performance (models M2 and M3) but adding both significantly improves the decoding performance. Visualizing the trial-averaged plots next throws light on this result.

4.2 Trial-averaged interpretation

We plot the trial-averaged z^1, z^2, z^3 for all 4 models in Figure 2A. Although subspace switching may vary from trial to trial, we expect the trial-averaged plot to reveal an overall trend, providing an average representation of the majority of trials. It can be observed that even without unimodality or CNN, our model recovers a good subspace corresponding to the stimulus (red), choice (blue), and null (green) subspace. The timing of the peak for z^1 (stimulus) appears around t = -1, which happens

[1] We use the SLDS implementation provided at: *https://github.com/lindermanlab/ssm*.

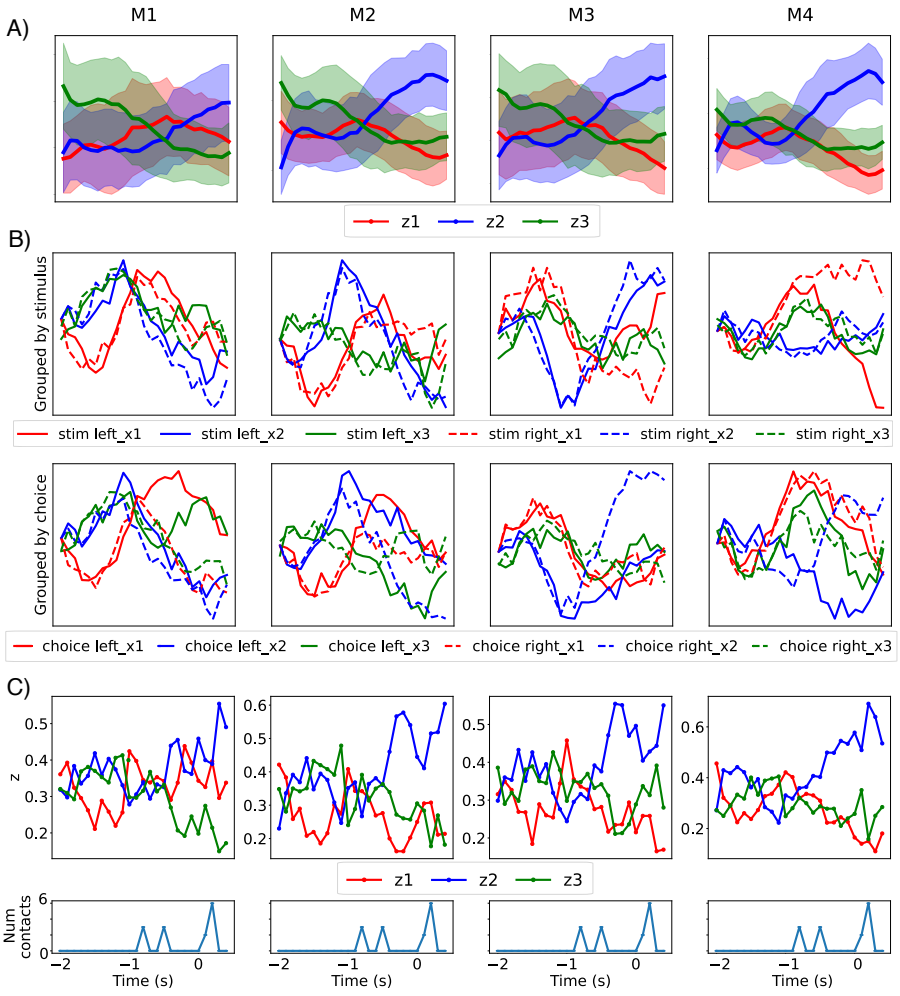

Figure 2: A) Trial-averaged assignment latents z^1, z^2, z^3 for all 4 models. The x-axis indicates the time in seconds. The solid line indicates the mean while the shaded portion indicates one standard deviation across all 161 trials. B) Trial-averaged continuous latents x^1, x^2, x^3 for all 4 models. Each column indicates one model. The top row contains trial-averaged x^i, separately averaged for stimulus left and right. Similarly, the bottom row contains x^i trial-averaged over left choice and right choice. C) Single-trial results for all 4 models for a particular trial. The first row contains inferred z and the second row contains the aggregate number of contacts made by all 4 whiskers.

to be near the time when most contacts take place [Rodgers et al., 2021]. z^2 (choice) ramps up at the end as expected.

Visualizing the trial-averaged continuous latents x^i throw light on the discriminative information encoded in them (Figure 2B). Each dotted and solid line of the same color represents the trial-averaged latent for binary behavior (convex vs concave for stimulus, top row; left vs right for choice, bottom row). Hence, the higher the separation between the two, the higher the discriminative information encoded by the continuous latent for the corresponding behavior.

Referring back to Table 1, we infer the following: For M1, we can see that some choice information is present in x^1 since the red curve separates out for left and right choice (Column 1, Row 2 in Fig. 2B). On the other hand, no such separation is seen for stimulus in the case of any of the latents (Column 1, Row 1). This is reflected in their poor decoding performances. Enforcing unimodality in M2 leads to a marginal improvement in the decoding performance (Table 1) due to a more structured latent space. M3, which adds a CNN, substantially improves both choice and stimulus performance as expected but the drop in bits/spike is noticeable. This could be because the CNN forces the model to learn a latent specifically tuned for decoding, hurting reconstruction. Surprisingly, further adding unimodality to CNN (M4) improves both: reconstruction and decoding. This is much higher than the

Table 2: Evaluation metrics for SLDS models (S1-4) considering all 4 input combinations. Similar to the previous section, we train a CNN on pairs of x^i, z^i to extract behavioral information; refer to Section A.1 for methodology. Terms in brackets indicate the latent which gave the best performance.

Model	Stimulus in u	Choice in u	Bits/Spike	Stimulus Decoding (%)	Choice Decoding (%)
S1	✗	✗	**0.33**	66.6 (x^2)	63.6 (x^1)
S2	✓	✗	0.31	69.6 (x^3)	63.6 (x^3)
S3	✗	✓	0.32	63.6 (x^1)	**72.7** (x^2)
S4	✓	✓	0.31	**72.7** (x^1)	66.6 (x^2
M4 (ours)	-	-	**0.55**	**87.8**	**93.9**

SVM decoding from raw spikes. Bits/Spike also goes up marginally, proving that both unimodality and CNN help discover a better latent space, which also captures stimulus and choice.

4.3 Single-trial interpretation

In Figure 2C, we examine the inferred z, along with the contacts for a single trial. It can be observed that for M1, the z is very noisy. In M2, choice (z^2) peaks and saturates at the end. However, as seen in Table 1, the inferred z and x in M2 are not very helpful for decoding stimulus or choice, since there is no supervision. In M3, z^2 dips and peaks again later while stimulus latent z^1 barely crosses the choice latent z^2 during contacts. M4 demonstrates all desired characteristics: stimulus peak during contacts and choice saturation at the end, all while preserving reconstruction performance and high decoding accuracy.

Another interesting discovery is that, in both trial-averaged and single-trial plots, there is a bump in z^2 at the beginning of the trial, indicating that the neural data already encodes some choice information. Our hypothesis is that this encodes choice information from the previous trial. To test this, we decoded spike trains to predict both the previous and current trials' choices. We found that the accuracy for predicting the previous choice (72%) was higher than for the current choice (48%). This indicates that our method can detect choice-encoding behavior from the previous trial and identify when it fades, being overtaken by the stimulus subspace as the new trial progresses. Later in the trial, choice encoding re-emerges as new evidence accumulates. We can discover this phenomenon for each single trial separately. Due to space constraints, we plan to explore this in more depth in future work.

4.4 SLDS results

Reconstruction and decoding performance on using SLDS can be found in Table 2. Following the same convention, we refer to them as S1, S2, S3, and S4 where each has a different input u. Firstly, we observe that Bits/Spike for all 4 variants is much worse compared to any of our models (0.33 vs 0.55). Although our model has more parameters, reconstruction is still bottlenecked by the 3-dimensional latent X, similar to SLDS. This shows that modeling the observations as switching subspaces is more appropriate in this case.

For the discrete latent z, we observe that switching in SLDS mainly occurs near t = -1, which happens to be the point where contacts begin to take place (Figure 3A). Moreover, among all 4 models, only S3 captures 3 distinct dynamics, and others only capture 2. This implies that SLDS is mainly capturing two dynamics: the null dynamic before contacts and a mixed choice/stimulus dynamic post contacts. For S3, state 1 captures the null dynamic while both states 2 and 3 capture the stimulus dynamic as they both ramp up. No separate choice dynamic can be seen, evolving between -1 and 0 and saturating at the end. This is the **key shortcoming of SLDS**: it captures a switching dynamic instead of a switching subspace. We hypothesize that the choice subspace dominates the observations at the end of the trial but SLDS fails to discover it.

Similar to our model, we also plot the trial-averaged x in Figure 3B. For behavior decoding results in Table 2, we observe that the best stimulus decoding is 72.7% while best choice decoding is also 72.7%. In Figure 3B, we see that the trial-averaged x is well separated across left (solid) and right (dashed) for x^2 in S3, giving rise to non-trivial choice decoding performance. Similarly, x^1 separates

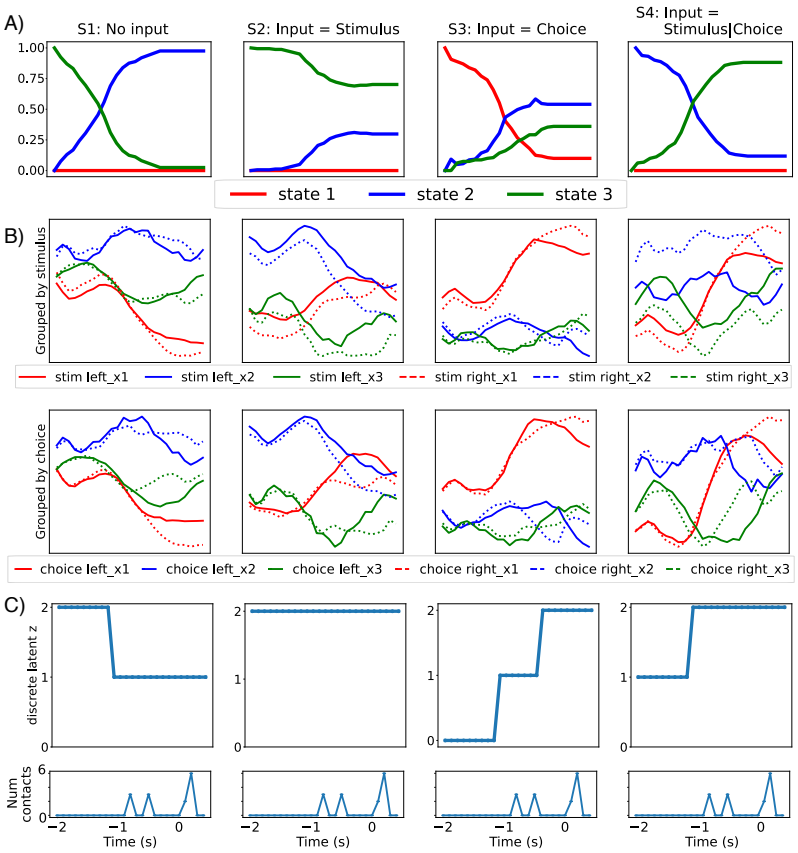

Figure 3: A) Trial-averaged discrete latent z in SLDS, computed by individually computing the averaged assignment of z^i across trials at each time bin. B) Trial-averaged continuous latent x in SLDS. It is averaged separately for each label for choice and stimulus. C) Single-trial results for all 4 models of SLDS for the same trial. The first row contains the discrete state z while the second row contains the aggregate of contacts made by all 4 whiskers.

out at the end for S4 (Column 4, Row 1 in Figure 3B), giving 72% stimulus decoding. However, both are much less than our best model (M4).

The single-trial results for SLDS in Figure 3C corroborate the story: switching (if it occurs) happens due to the change from trial-irrelevant activity to contacts in models S1 and S4 (no switching in S2). The emergence of the choice dynamic at the end is captured only by S3 for this particular trial. The trial-averaged continuous latent x shows some separation across choice but it is minimal (Column 3, Row 2 in Figure 3B). This is reflected in the 72% choice decoding performance (Column 4, Row 3 in Table 2).

5 Conclusion

In this work, we proposed a novel switching subspace model that captures trial-level dynamics of multiple variables of interest. Such a model allows multiple dynamics to coexist in time, as opposed to the popular SLDS approach of modeling a switch in the dynamics. We enforced a unimodality prior to obtain a more interpretable latent and proposed a simple approach to include supervision in the model in order to preserve weak but important signals. We demonstrated a much higher bits/spike than SLDS, interpretable single-trial results, and high decoding accuracy for both choice and stimulus.

One key limitation of our work is the simple prior (Standard Normal) assumption over the continuous dynamic latent x. Previous works have shown that imposing a dynamic or temporal dependence assumption over x is helpful to capture interpretable latent dynamics with good data explanation [Wu et al., 2017, She and Wu, 2020]. In future work, we plan to explore a latent dynamic prior over x to discover more interpretable latents.

References

Michael Riis Andersen, Eero Siivola, and Aki Vehtari. Bayesian optimization of unimodal functions. In *NIPS workshop on Bayesian optimization*, 2017.

Mikio Aoi and Jonathan W Pillow. Model-based targeted dimensionality reduction for neuronal population data. *Advances in neural information processing systems*, 31, 2018.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020.

Christopher D Harvey, Philip Coen, and David W Tank. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 484(7392):62–68, 2012.

Munib A Hasnain, Jaclyn E Birnbaum, Juan Luis Ugarte Nunez, Emma K Hartman, Chandramouli Chandrasekaran, and Michael N Economo. Separating cognitive and motor processes in the behaving mouse. *bioRxiv*, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *eLife*, 5:e10989, apr 2016. ISSN 2050-084X. doi: 10.7554/eLife.10989. URL https://doi.org/10.7554/eLife.10989.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial intelligence and statistics*, pages 914–922. PMLR, 2017.

Scott W Linderman, Andrew C Miller, Ryan P Adams, David M Blei, Liam Paninski, and Matthew J Johnson. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.

Thomas Zhihao Luo, Timothy Doyeon Kim, Diksha Gupta, Adrian G Bondy, Charles D Kopec, Verity A Elliot, Brian DePasquale, and Carlos D Brody. Transitions in dynamical regime and neural mode underlie perceptual decision-making. *bioRxiv*, pages 2023–10, 2023.

Ramon Nogueira, Chris C Rodgers, Randy M Bruno, and Stefano Fusi. The geometry of cortical representations of touch in rodents. *Nature Neuroscience*, 26(2):239–250, 2023.

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.

Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V Shenoy, Matthew T Kaufman, Mark Churchland, et al. Neural latents benchmark'21: evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

Chris Rodgers. A detailed behavioral, videographic, and neural dataset on object recognition in mice, 2022a. URL https://dandiarchive.org/dandiset/000231/0.220904.1554.

Chris C. Rodgers. A detailed behavioral, videographic, and neural dataset on object recognition in mice. *Scientific Data*, 9(1):620, Oct 2022b. ISSN 2052-4463. doi: 10.1038/s41597-022-01728-1. URL https://doi.org/10.1038/s41597-022-01728-1.

Chris C Rodgers, Ramon Nogueira, B Christina Pil, Esther A Greeman, Jung M Park, Y Kate Hong, Stefano Fusi, and Randy M Bruno. Sensorimotor strategies and neuronal representations for shape discrimination. *Neuron*, 109(14):2308–2325, 2021.

Qi She and Anqi Wu. Neural dynamics discovery via gaussian process recurrent neural networks. In *Uncertainty in Artificial Intelligence*, pages 454–464. PMLR, 2020.

Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D Harris. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 364(6437):eaav7893, 2019.

Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

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. *Advances in neural information processing systems*, 21, 2008.

David Zoltowski, Jonathan Pillow, and Scott Linderman. A general recurrent state space framework for modeling neural dynamics during decision-making. In *International Conference on Machine Learning*, pages 11680–11691. PMLR, 2020.

A Appendix

A.1 Decoding Choice and Stimulus from latents

To decode a behavioral variable from the latents $x^i \in \mathcal{R}^{\mathcal{T}}, z^i \in \mathcal{R}^{\mathcal{T}}$, we use a 1-layer, 5-kernel CNN with ReLU activation. The hidden layer has 8 channels while the output layer has 1 channel, denoting the scalar behavior prediction at each time point. Input is padded on both sides to preserve dimensionality over time axis. Refer to Algorithm 1 for more details.

Algorithm 1 Decoding a behavioral variable

Require: $x^i \in \mathbb{R}^T$ ▷ Continuous latent variable
Require: $z^i \in \mathbb{R}^T$ ▷ Assignment latent variable
$o \leftarrow \text{CNN}(x^i); o \in \mathcal{R}^T$ ▷ Apply CNN filter over x^i
$o_t \leftarrow o_t \cdot z_t^i \quad \forall t$ ▷ Gate the output with z^i
$p \leftarrow \sum_{t=1}^{T} o_t/T; p \in \mathcal{R}$ ▷ Compute the mean across time
return p ▷ Return the predicted logit

A.2 Pre-training of GRU

We found that training was sensitive to the seed. In order to work around this, we first pre-trained only the encoder (GRU) and MLP 1 (which captured z) to output an assignment latent with the following properties: z^1, which is supposed to capture stimulus, peaked around the start of contacts while z^2, which is supposed to capture choice, peaked midway through the contacts. This is possible since we have access to the contacts for every trial. Training of the end-to-end network continued from this pre-trained state.

A.3 Importance of GP prior

We add a GP prior over the assignment latent z in order to improve the interpretability of z as discussed previously. Results **without** such a GP prior are given in Figures 4, 5, 6. We use the same model as M4 i.e. with unimodality on z^2 and supervision using a CNN for both choice and stimulus. The bits/spike for test is 0.49, which is much poorer than the same model with GP prior (0.55). The key takeaway is that GP smoothing significantly improves interpretability.

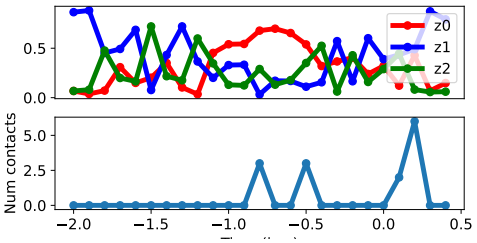

Figure 4: Single-trial plot for the same trial examined previously. We can observe that z is very noisy in the beginning of the trial, making interpretation difficult.

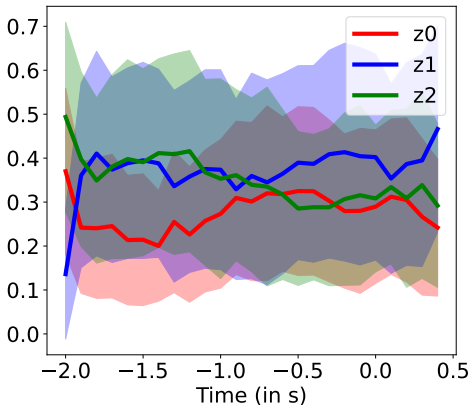

Figure 5: Trial-averaged z looks almost flat, which is not desirable.

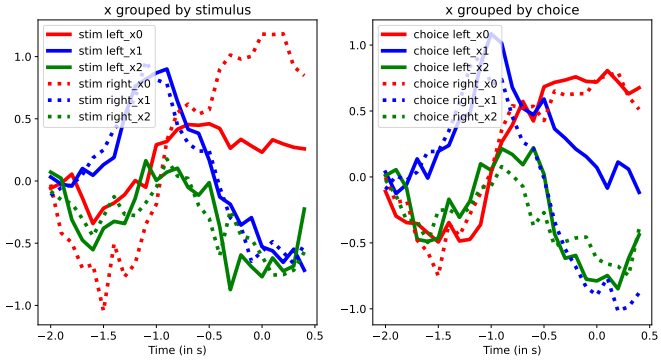

Figure 6: For the trial-averaged case: x^1 and x^2 do show trial-averaged separation for stimulus and choice respectively, indicating decent subspace recover due to the presence of supervision. However, single-trial results are uninterpretable.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The experimental results show that the proposed approach better models the observed data.

   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: Limitations of our work are discussed in the final section.

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: No theoretical contributions in this paper.

   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 provide the hyperparameter values for all components of our model in Section 3.3 and the evaluation methodoloy in section A.1.

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 cited dataset is openly available. We plan to release our code upon acceptance.

   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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameter values are provided in the paper. Exact implementation can be found in the code.

   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: [No]

   Justification: Since we intialized our GRU from a good pre-trained checkpoint as discussed in Section A.2, varying random seeds does not have a significant effect on the final model.

   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: We mention these details in Section 3.3.

   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: We have followed the 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: [NA]

    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: [NA]

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: We cite the dataset and the license is respected.

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: [NA]

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: [NA]

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: [NA]

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.