Drop, Swap, and Generate: A Self-Supervised Approach for Generating Neural Activity

Anonymous Author(s)
Affiliation
Address
email

Abstract

With meaningful and simplified representations of neural activity, we often are afforded insights into how and what information is being processed within a neural circuit. However, without labels, the process of finding representations that reveal the link between the brain and behavior can be challenging. Here, we introduce a novel unsupervised approach for learning representations of neural activity called *Swap-VAE*. Our approach combines a generative modeling framework with an *instance-specific alignment* loss that tries to maximize the representational similarity between transformed views of the input (brain state). By creating these views through dropping out neurons and jittering samples in time, we essentially ask the network to find a representation that maintains both temporal consistency and invariance to the neurons used to represent the specific brain state. We then couple this with a novel block swapping latent augmentation and a generative model to simulate new high-dimensional neural activities. Through evaluations on both synthetic and real neural datasets from hundreds of neurons in different primate brains, we show that by combining our self-supervised alignment loss with a generative model, we can build representations that disentangle neural datasets along relevant latent dimensions linked to behavior.

1 Introduction

In the brain, the coordinated action of groups of neurons is responsible for behaviors as diverse as encoding sensory inputs and generating movements. Although neurons are powerful, the schema by which they encode information are highly distributed, and the activity of individual neurons is often highly variable, even in constrained circumstances [1, 2, 3, 4, 5, 6]. As a result, it is challenging to decode complex behavior from the activity of a population of neurons given only the raw records of their spike times [7]. Remarkably, even when the individual responses of neurons vary, the population-level responses, which are considered to be the basis of sensation, perception, cognition, and behavior, remain stable (to some extent) [8, 9, 10]. This suggests that extracting stable representations from neural datasets might provide a decoder for information encoded in the brain, which would be a massive leap forward for neuroscience.

With the rapid growth of deep learning, there has been a great deal of interest in applying these methods to extract representations of neural activity that reveal underlying links to behavior or decision making [11, 12, 13, 14, 10, 15, 16]. The success of such approaches often grants insights into how neural circuits work to encode their inputs and drive decisions. However, without labels, learning latent spaces that can appropriately reveal and *disentangle different sources of variability* is still a major challenge [17, 18]. Moving forward, we need new tools to decompose the latent factors that drive neural activity.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

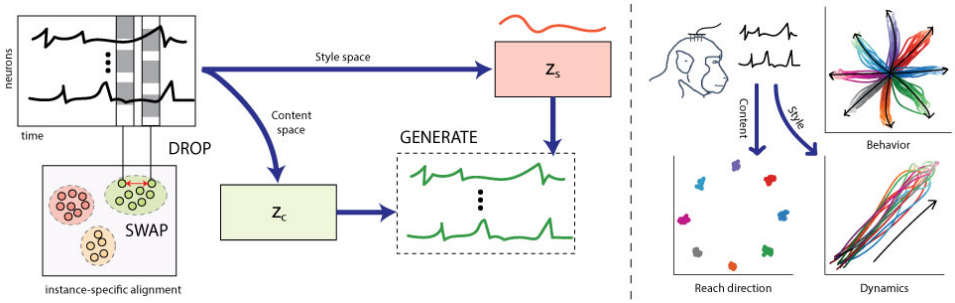

Figure 1: *Overview of approach.* The model's latent variables are partitioned into a content space and a style space. An instance-specific loss is applied to the content representations of augmented views of a brain state (shifting in time and dropping out neurons) to encourage alignment, while views are reconstructed at the output of the decoder using both parts of the latent space. To further enhance the clustering in the content space, we introduce *BlockSwap*: two augmented views are projected through the encoder and in the content space their representations are swapped before being passed through the decoder. To the right, we show an example of applying this approach to disentangle the neural representation of movement.

Here, we develop a novel self-supervised approach for generative modeling of neural activity called *Swap-VAE*. Our approach is loosely inspired by methods used in computer vision that aim to decompose images into their *content* and *style* [19, 20, 21, 22, 23]: the representation of the content should give us the abstract "gist" of the image (what it is), and the style components are needed to create a realistic image (or, equivalently, they capture the variation in images with the same content). To map this idea onto the decomposition of brain states, consider the execution of movements and their representation within the the brain (Figure 1, Right). The content in this case may be *knowing where to go* (target location) and the style would be the *exact execution of the movement* (the movement dynamic). We ask whether the neural representation of movement can be disentangled into these components using our proposed conceptual framework.

To build the content space, we seek to minimize an instance-specific alignment loss that effectively penalizes the deviation between *transformed views* of the input in the latent space of the encoder. In other words, we apply a variety of transformations to the input data in an effort to learn the consistent "truth" that persists despite manipulation. These transformed (or augmented) views are created by dropping out neurons and jittering samples in time, which intuitively should lead the network to a representation that maintains both temporal consistency and invariance to the specific neurons used to represent the neural state. In addition to this alignment loss, we use a separate set of latent variables to reconstruct the original inputs using a regularized variational autoencoder (β-VAE [24, 25]). To further encourage disentanglement, we introduce a novel latent space augmentation called *BlockSwap*, where we swap the content components of two examples and ask the network to predict the original view. We show that through combining our self-supervised alignment loss with a generative model, we can build good representations that disentangle neural datasets.

We apply our method to synthetic data and non-human primate (NHP) reaching datasets from two different individuals (using the same datasets as in [15] from [26]). To quantify the disentanglement of our representations of these data, we compute a number of measures of representation quality and also compute linear decoding accuracies across both spaces to test whether more information about the target (what) and time (how) are encoded in the two different spaces. We show that by using our approach, we can effectively disentangle the behavior and dynamics of movement without any labels. Our model strikes a nice balance between both view-invariant representation and generation.

Our specific contributions are as follows:

- In Section 3, we propose a generative method, *Swap-VAE*, that can both (i) learn a representation of neural activities that reveal meaningful and interpretable latent factors and (ii) generate realistic neural activities.
- In Section 3.3, we propose a novel latent representation augmentation procedure called *BlockSwap* which swaps variables in the latent space to improve disentanglement.
- In Section 3.4, we introduce metrics to quantify the disentanglement of our representations of behavior and apply them to neural datasets from different non-human primates (Section 4) to gain insights into the link between neural activity and behavior.

2 Background and Related Work

2.1 Variational autoencoders and their application in neural data analysis

Variational auto-encoders (VAEs) [27] are a popular deep generative learning framework due to their ability to generate high-quality data. Denote the data and the latent variables by $\mathbf{x}$ and $\mathbf{z}$, where $\mathbf{z} = q_\phi(\mathbf{x})$ is the latent representation extracted from $\mathbf{x}$ by the encoder q_ϕ. The usual objective of probabilistic generative models is to maximize the log evidence of the observed data $\max_\theta \log p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ based on the parameterized model p_θ (called the decoder). VAEs and their variants instead optimize a tractable lower bound on the original objective, which is also well-known as the evidence lower bound (ELBO) in variational Bayesian inference [28],

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{x})}[p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) =: \mathcal{L}^{\text{VAE}}_{\theta,\phi}, \tag{1}$$

where the encoder q_ϕ is trained to approximate the Bayes posterior $p_\theta(\mathbf{z}|\mathbf{x})$, $p(\mathbf{z})$ is the prior over the latent variables, D_{KL} is the Kullback–Leibler divergence, and $\beta \geq 1$ is a trade-off parameter. Specifically, the standard VAE corresponds to $\beta = 1$ while $\beta > 1$ gives the β-VAE [24]. A larger value of β imposes implicit regularization on the posterior of latent variables to align with the prior, which empirically induces more disentanglement and thus interpretability in the learned representations [25, 29, 30].

Recently, a number of generative modeling techniques based upon VAE have been applied to neural data. LFADS and more recent extensions of this model [10, 13], use a sequential VAE [31, 32] to estimate neural population dynamics and show that this model can faithfully reconstruct single trial firing rates. More recently, pi-VAE [11] was proposed to learn representations of neural activity using a simple MLP encoder to capture information in each neural state vector (firing rate of d neurons) independently. They show that, even with a simplified architecture that treats each sample independently, it is possible to learn an identifiable model that can directly link behaviors to neural responses. In our work, we also use a MLP encoder similar as [11], where we do not explicitly incorporate temporal structure into our architecture. Instead, we ask the model to disentangle the content (target) from the dynamics without regularization from a more complex architecture.

2.2 Instance-specific alignment and self-supervision

Recent SSL methods have made impressive advances, now rivaling (or in some cases surpassing) supervised methods [33, 34, 35, 36]. To build representations, these approaches aim to maximize the similarity across multiple augmented "views" of the same data (positive pairs) [37, 38, 35]. Thus, instead of providing labels to regularize the latent representations, we instead supply augmented versions of the same example and align their latent representations. Intuitively, by *aligning the instance-specific views* where the semantic content is preserved, the network will learn meaningful representations, which can then be used to solve downstream tasks [39].

Building on these successes, [15] recently showed how instance-specific alignment can be applied to multi-neuron recordings. They demonstrated that through the combination of simple augmentations, including: dropout, temporal shift (selecting nearby points in time), and additive noise to neural activity, with a dual network approach, it is possible to learn representations from neural datasets. In this work, we further show that coupling this type of alignment approach with a generative model, we can build networks that achieve both good approximation quality and disentanglement, all without the need for a dual network approach. This greatly simplifies our architecture and optimization approach.

3 Methods

In this section, we introduce our self-supervised approach for generative modeling of neural data. A PyTorch [40] implementation and demos will be made available here: [url hidden for review].

3.1 Model for neural datasets

Throughout, we consider collections of d neurons that have been spike sorted and binned to compute an estimate of the firing rate of all neurons at N distinct time points (bins). This results in an input vector $\mathbf{s}_i \in \mathbb{R}^d$ as an observation at each time point. Let $\mathcal{D} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ denote the neural state vectors generated by this binning process. Let k denote the dimension of the latent space.

3.2 Unifying instance-specific alignment and generative modeling

As we describe in the introduction, our aim is to build a decomposable picture of brain states. To do this, we will leverage the principles of *self-supervision* to build a view-invariant representation and then use this as the building block for our generator.

Our goal is to learn two functions, an *encoder* $f : \mathbb{R}^d \to \mathbb{R}^k$ and *decoder* $g : \mathbb{R}^k \to \mathbb{R}^d$. Let $\mathbf{x}_1 = t_1(\mathbf{s})$ and $\mathbf{x}_2 = t_2(\mathbf{s})$ denote the views generated after applying two random transformations $t_1, t_2 \sim \mathcal{T}$ to a sample $\mathbf{s}$ from our dataset $\mathcal{D}$. Let $\mathbf{z}_1 = f(\mathbf{x}_1), \mathbf{z}_2 = f(\mathbf{x}_2)$ denote the representations of both views in the network. To decouple the factors of our latent representations, we divide the latent space into two parts, $\mathbf{z}_1 = [\mathbf{z}_1^{(c)}, \mathbf{z}_1^{(s)}]$ and $\mathbf{z}_2 = [\mathbf{z}_2^{(c)}, \mathbf{z}_2^{(s)}]$, with $\mathbf{z}_1^{(c)}$ and $\mathbf{z}_1^{(s)}$ modeling the behaviour styles and intrinsic neural contents, respectively. $\widehat{\mathbf{x}}_1 = g(\mathbf{z}_1)$ and $\widehat{\mathbf{x}}_2 = g(\mathbf{z}_2)$ are the reconstructions of both views obtained after passing them through the decoder.

To encourage alignment of the views through the encoder while also solving our generative modeling objective, we propose the following loss:

$$\min_{f,g} \sum_{i=1,2} \underbrace{\mathcal{L}_{\text{rec}}(\mathbf{x}_i, g(\mathbf{z}_i))}_{\text{Reconstruction loss}} + \beta \sum_{i=1,2} \underbrace{D_{KL}(\mathbf{z}_i^{(s)} \parallel \mathbf{z}_{i,\text{prior}}^{(s)})}_{\text{Regularization - style space}} + \alpha \underbrace{\mathcal{L}_{\text{align}}(\mathbf{z}_1^{(c)}, \mathbf{z}_2^{(c)})}_{\text{Alignment - content space}}, \tag{2}$$

where the alignment loss $\mathcal{L}_{\text{align}}$ encourages two views to be close (here we used a normalized L2-distance), α and β are hyperparameters that determine the tradeoff between alignment and reconstruction, and the KL divergence terms measure the deviation between the style latent variables and the prior $\mathbf{z}_{i,\text{prior}}^{(s)}$ which we set to be the isotropic Gaussian $\mathcal{N}(\mathbf{0}, I)$. In our experiments on neural datasets, we choose the reconstruction loss $\mathcal{L}_{\text{rec}}$ to be the Poisson loss [10, 41]. Further details on the method and our implementation is provided in Appendix A.

3.3 *BlockSwap*: A novel latent space augmentation for disentanglement

Our previous objective provides a general-purpose strategy for combining instance-specific alignment with a VAE. To further improve disentanglement, we propose the following novel *latent space augmentation*, which basically swaps the content information (block of variables) between two augmented views while keeping their style constant. We refer to this latent augmentation as *BlockSwap* as it holds one part of the representation consistent and then swaps a different subset of latent variables between two augmentations. Specifically, after generating the representations $\mathbf{z}_1 = [\mathbf{z}_1^{(c)}, \mathbf{z}_1^{(s)}]$ and $\mathbf{z}_2 = [\mathbf{z}_2^{(c)}, \mathbf{z}_2^{(s)}]$ for each view, we also generate their content-swapped versions $\widetilde{\mathbf{z}}_1 = [\mathbf{z}_2^{(c)}, \mathbf{z}_1^{(s)}]$ and $\widetilde{\mathbf{z}}_2 = [\mathbf{z}_1^{(c)}, \mathbf{z}_2^{(s)}]$. To encourage disentanglement, we propose to replace the previous reconstruction term by adding the loss over the swapped representations:

$$\mathcal{L}_{\text{rec}}^{\text{swap}} = \sum_{i=1,2} \underbrace{\mathcal{L}_{\text{rec}}(\mathbf{x}_i, g(\widetilde{\mathbf{z}}_i))}_{\text{Swapped content}} + \underbrace{\mathcal{L}_{\text{rec}}(\mathbf{x}_i, g(\mathbf{z}_i))}_{\text{Original}}. \tag{3}$$

When we use this loss, we can also consider removing the alignment term in Equation 2 and simply couple this reconstruction loss with the regularization KL term on the style space (see Section 4.3). In practice, the reparameterization trick is performed before the swapping of content and style latent.

3.4 Representational quality and disentanglement metrics

Knowing when we have a good latent space that captures the underlying factors of interest is, in general, very challenging [17, 42, 43]. In neuroscience this is certainly true. In this work, we will consider two main measures of representation quality and disentanglement to guide our investigation.

Multi-task disentanglement score. To confirm that our method effectively disentangles the represented behavior of neural signals from the dynamics, we need to measure the extent to which latent variables respond to one or the other with specificity. In other words, it should be possible to divide the latent variables into behavior-encoding and dynamic-encoding sets, so that the value of a variable in one of the two sets only changes significantly when the associated parameter (i.e. reaching direction or dynamics) is changed. Concretely, let $\mathbf{z}$ denote the latent representation, and y_c, y_s discrete variables parameterizing the reaching direction and dynamics, respectively. Computing the covariance score for a particular latent variable $\mathbf{z}_i$ consists of three steps:

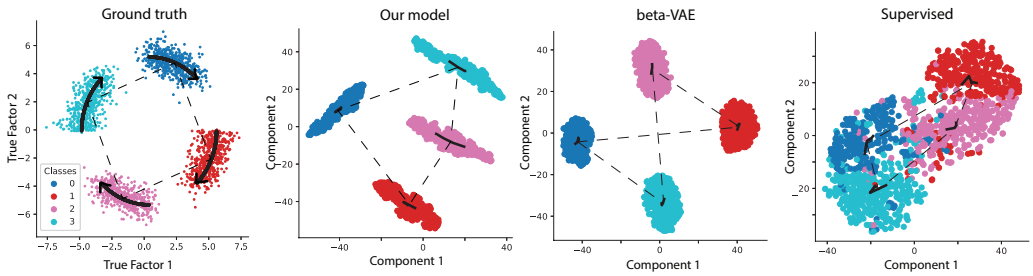

Figure 2: *Synthetic Experiments.* On the left, we show the ground truth latent space and their dynamics, from which the firing rate is generated in a clockwise manner. To the right, we show the results of our model, a beta-VAE, and a supervised model. Our model recovers both the discrete classes and the sequential structure present within this synthetic dataset.

1. Compute the variance when changing y_c with fixed y_s, and average over values of y_s.
2. Compute the variance when changing y_s with fixed y_c, and average over values of y_c.
3. Compute the absolute difference of the two variances. This is the score for $\mathbf{z}_i$.

Intuitively, if the score is large, then $\mathbf{z}_i$ changes more dramatically in response to one parameter than the other, so it displays specificity, while if it is low then the amount that $\mathbf{z}_i$ changes is nearly the same. Averaging across all latent variables after normalization gives a final score, which provides a measure of how disentangled the entire representation is.

Linear readout from representation layer. To further quantify the representation quality and stability of representations in downstream decoding, we use a linear readout strategy employed frequently in self-supervised learning approaches [33, 35]. In particular, we will train the model on our training dataset, freeze the weights in the network, and then train a linear layer to decode the reach directions from the output of the encoder. However, because we are also interested in disentanglement, we will consider the prediction of two different class labels from either the full, content, or style factors in the network. When decoding either reach directions y_c or temporal structure y_s, we will retrain the linear weights but keep the representation fixed.

Similar to [15], we have two scores, acc and delta-acc, for the linear decoding accuracy on reach direction. Consider the reaching task as a regression over a circle with a total of l discrete labels, we count the decoded angle that falls within $[(2i-1)\pi/l, (2i+1)\pi/l]$ as the correct classification in acc, and that falls within $[(2i-1.5)\pi/l, (2i+1.5)\pi/l]$ as the correct classification in delta-acc. The two scores jointly evaluate the representation quality in terms of the reach direction decoding precision.

4 Experiments

The usefulness of this method depends on its ability to decompose neural data into meaningful latent factors. To quantify this, we devised experiments to reveal three desirable properties:

1. Performance on downstream classification tasks (linear separability of the representations).
2. Faithfulness of the latent space structure to the ground-truth structure of the task.
3. Separability of the content and style components across the respective latent subspaces.

4.1 Synthetic experiments

We first trained and evaluated our model on an artificially generated data that was designed resemble our neural datasets of interest. The data is generated and the experiments are designed following the approach used in [44, 11].

Synthetic reaching dataset. We generated latent variables from a 2-dimensional independent Gaussian distribution with its mean being $(5\sin u, 5\cos u)$ and variance being $(0.6 - 0.3|\sin u|, 0.3|\sin u|)$, where the u is uniformly sampled from 4 clusters $[\frac{i\times\pi}{4}, \frac{(i+1)\times\pi}{4}], i \in \{0, 2, 4, 6\}$. For each sequence, we randomly sampled $l = 4$ data points within each cluster and rank them in a clockwise manner to form a sequence to capture dynamics within each behavior in the latent space (as shown in Figure 2 on the left). The formed sequences were fed to a RealNVP network [45] to generate 100-dimensional Poisson observations as the synthetic firing rate. The generated

synthetic firing rate is shuffled and split into 80% training set and 20% test set. The ground truth latent distributions and the generated results are shown in Figure 2. Ideally, we want the model to learn both the discrete groups/clusters y_c, and also learn the dynamics of the sequence that is implicitly encoded via the RealNVP network.

Results on synthetic datasets. To study into the representational power of our approach, we applied our model, the β-VAE, and a supervised decoder (trained to predict y_c) to synthetic datasets. All models have the same backbone with a latent space of 32-dim, while our model has a content space of 16-dim and a style space of 16-dim. All models are trained on the training set for 100,000 iterations, with Adam optimizer with a learning rate of 0.0005 (More details on model selection and hyperparameter optimization in Appendix A). When we examined the representations formed by each of these models, we found that *Swap-VAE* was very effective at both preserving the sequence dynamics (as highlighted by the connection between class centroids) as well as separating the different target classes (Figure 2). From the figure, we can see that while the β-VAE successfully separated different clusters, the dynamics are barely encoded (the black line), while the elongated distribution formed by our model more accurately reflected the true distribution of each component, regardless of the noise. Using the metrics described in Section (3.4), we further confirmed that our model provides good disentanglement, producing a multi-task disentanglement score of 0.93. The corresponding score for the beta-VAE and supervised model were 0.46 and 0.12, respectively.

4.2 Experiments on neural datasets

After testing the model on synthetic datasets, we applied our model to datasets collected from the primary motor cortex of non-human primates performing a reaching task with two different settings. The dataset is used in recent studies of deep representation learning [15] and interpretable generative modeling [11].

Motor cortex reaching datasets. We use reaching as a simplified laboratory task to test our hypothesis that the *what* and *how* of movements could be disentangled. We consider spike sorted datasets from two individuals, Monkey C and Monkey M, both trained to perform a reaching task with 8 different directions. The reaching task has two different settings: Monkey C performs the reaching task directly, while Monkey M performs the reaching task with a waiting period of time (between 500-1500 ms) before receiving an auditory 'Go' cue. While carrying out these movement tasks, neural activities in primary motor cortex (M1) were recorded. In these examples, the summed activity of a population of roughly one hundred single neurons was binned into 100ms intervals to generate approximately 1.3k data points per dataset. For each direction, there are multiple trials/repeats. For each monkey, 2 days of neural recordings are considered, where different groups of neurons are recorded on different days.

Experiment setup. In our model, we apply a combination of two augmentations: (i) *spatial augmentations*, where we randomly dropout neurons from the input (with $p = 0.6$), and (ii) *temporal augmentations* that select a nearby point in time (randomly in a window, ± 5 samples from target sample) as a positive example. All models have a 128-dim latent space, where for our model the style and content space are both 64-dim. All models are trained using one Nvidia Titan RTX GPU for 200 epochs with the Adam optimizer with a learning rate of 0.0005 (Further details can be found in Appendix A). With d as the total number of neurons, all generative models have an encoder and a symmetric decoder, where the encoder has three linear layers with size $[d, 128, 128]$, batch normalization, and the ReLU activation. All discriminative models have an encoder of 4 linear layers with size $[d, 128, 128, 128]$, which is a more optimal encoder architecture for discriminative models. In our experiments, we split the dataset into 80% for train and 20% for test.

Investigating disentanglement in neural representations of movement. After training our model, we examined the latent space structure by applying tSNE [46] to the Full space (considering both Content and Style jointly), as well as the Content and Style spaces individually (see Figure 3 for Chewie-1, Mihi-2 and Appendix B.1 for visualizations of the remaining datasets). When compared with a β-VAE and Supervised decoder trained on the reach direction task, we observe that the Content space in our model and the beta-VAE have similar overall structure, with our model providing further separability and preservation of the task structure (circular positioning of targets). The Style space provides a good embedding of the entire dataset along an axis where reach direction has been collapse but time is nicely organized. These results suggested that our model is good at separating semantic meanings *without any labels* while also preserving the overall structure of the behavior.

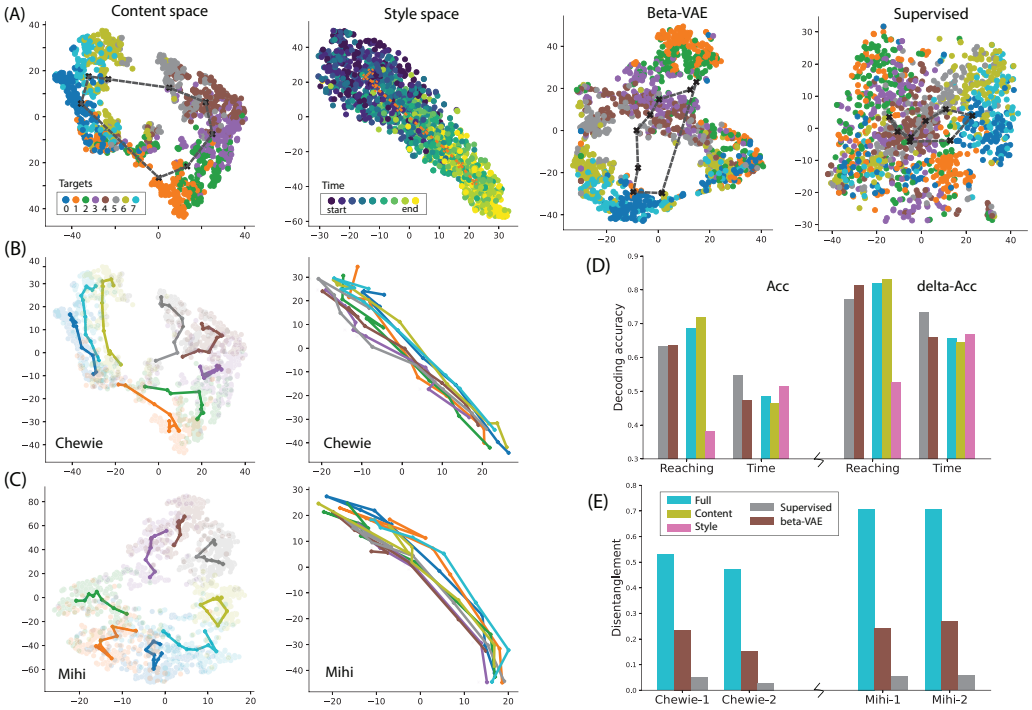

Figure 3: *Disentangling neural representations of movement in the primate motor cortex.* Along top in (A), we show the representations formed by our model's content and style space when compared with the β-VAE and a supervised network trained to decode reach direction. All of the representations are embedded into 2D using tSNE. Below, we decompose the content and style space further by averaging over all trials towards a specific reach and visualizing their trial-averaged trajectory for Chewie (Day 1) in (B) and Mihi (Day 2) in (C). In (D), we compare the decoding accuracies over both reach direction and time for Chewie-1 for our Full, Content, and Style spaces, the beta-VAE, and supervised decoders trained on either task (see legend in E). In (E), we show the results of our disentanglement score on all four reaching datasets. In this case, we compare the disentanglement over all of our latent space (Full) with the beta-VAE and supervised model trained on reach direction.

To understand how much information our model has about the two different downstream tasks, we examined the decoding accuracies in our reach and temporal decoding tasks on Chewie-1. We examined the Full, Content, and Style spaces for our model on both tasks, and compared with beta-VAE and supervised models trained on two tasks as the upper bounds. These measures provided further evidence of disentanglement as our Content spaces provide good decoding accuracies on reach decoding while the Style space has little predictive power over reach direction (as anticipated). The reverse is true for the temporal decoding in the Content space. These results are promising indicators that disentanglement is indeed possible with our approach and that these decoding measures capture what we observe in our visualization.

We next measured the multi-task disentanglement scores across all four datasets. When examining the separability of our latent space across the two individuals, we found that the disentanglement (i.e., separation between the reach direction and the dynamics of the movement) for Monkey C is on average lower than the Monkey M in both cases; this observation may be interesting given the fact that M needs to wait before making a reach and has to delay their movement at the beginning. While the results of this analysis need to be studied further, this result provides initial evidence that our unsupervised method for disentanglement provides a useful lens into the distinction between the neural representation of these two different movement tasks.

Stability of representations as measured through linear readouts. Next, we conducted a comprehensive evaluation of the decoding of the reach direction and the dynamics. In this case, we compared our model with a supervised decoder trained on either task, a supervised and an unsupervied disentanglement generative model (pi-VAE [11] and beta-VAE [24, 25]), and two self-supervised methods for general representation learning (BYOL [47], MYOW [15]). As the pi-VAE provides a state-of-the-art supervised baseline for disentangled generative modeling that has been applied on these same tasks (in fact from the same individual at different points in time), we consider these mod-

els to be a comprehensive collection of competitors for our tasks of interest. A table of comparison on target decoding is shown in Table 1, the comparison on temporal decoding is shown in Appendix B.2.

Through our analysis of decoding reach direction, we have interesting findings as follows. First, we find that our model is competitive with MYOW on the decoding task and outperforms this approach on a subset of the datasets. Both approaches outperform supervised decoders due to their strong regularization from augmentations. The power of our model shines in our ability to also decode the temporal structure from the data, through the use of a novel generative backbone with SSL.

Table 1: *Accuracy (in %) for reach direction classification on neural datasets.*

		Supervised	pi-VAE	beta-VAE	BYOL	MYOW	Ours
Chewie-1	acc	63.29	65.63	63.73	63.80	70.41	73.44
	delta-acc	77.22	82.62	81.36	81.90	86.24	85.38
Chewie-2	acc	72.29	60.60	58.07	57.17	60.95	66.06
	delta-acc	81.51	74.64	80.79	77.36	81.36	82.26
Mihi-1	acc	63.64	62.44	59.06	59.50	70.48	65.15
	delta-acc	79.02	77.12	75.04	79.78	83.24	81.16
Mihi-2	acc	61.49	63.26	58.95	60.82	64.35	67.78
	delta-acc	68.44	77.58	76.76	78.30	80.58	84.05

Testing the generative quality of the model. A key component of *Swap-VAE* is the integration of SSL with a generative modeling framework. Thus, we needed to test how well our model could generate neural activity. As shown in Figure 4, the direct reconstruction of the neuron firing rate is realistic in terms of both the class-conditioned firing rates, and the dynamics of individual neuron's firing rates. When we analyzed the RMSE of the fitted rate for all neurons in our model against those from a VAE, we found our model has a lower error and could reconstruct data more faithfully than the VAE. We obtain a good denoised estimate of neural activity that is more indicative of what aspects of neural responses are stable and related to the movement tasks (rather than noise).

Also, to show that our generated neuron activities are useful for downstream tasks, we performed generative data augmentation to supervised classifier, where we generate new samples and mix them with original training samples to train a supervised model. The supervised model takes class-conditioned generated neuron activities as the new data points, and fed into a supervised model. All supervised models are trained for 400 epochs, with the same model settings as mentioned in experiment setup. We tested on both the Chewie dataset and the Mihi dataset and computed the accuracy (gain and improvement) as we increase the number of generated samples included in the training set (50%, 100%, 200%). We showed that our generated neuron activities are useful such that it can enhance the performance of the supervised model. However, we note that the supervised models trained with generated samples still do not surpass the *Swap-VAE* (Details in Appendix C).

4.3 Model ablations: Testing our *BlockSwap* augmentation

In this section, we study different variants of the proposed loss functions and different data augmentation operations. Ablation of different dimensions of the content and style space is in Appendix D.

To examine how the different components of our loss play a role, we tested variants of our objectives in Equations (2) and (3). Specifically, we test our model performance in three cases: (i) when removing the alignment term in Eq. 2 but including the *BlockSwap* (no L2), (ii) when removing the alignment term and the original reconstruction term (Swap-only), and (iii) when keeping the alignment loss and the original reconstruction term but removing the *BlockSwap* augmentation (No-Swap). We also report the results obtained with a vanilla-VAE (beta-VAE results can be found in Table 1). The results are reported in Table 2. In our experiments, we find that almost all of the variants of our decomposed loss functions performs better than a vanilla VAE or a beta-VAE. Adding the *BlockSwap* loss term improves performance overall, with our highest decoding accuracies being obtained with this model. When we use the content swapping technique, we can remove the L2 alignment loss with minimal change in performance, but it in general, including this alignment terms provides an additional parameter to give more flexible control. This shows that our proposed model is stable and that our proposed swapping loss provides a strong boost in performance.

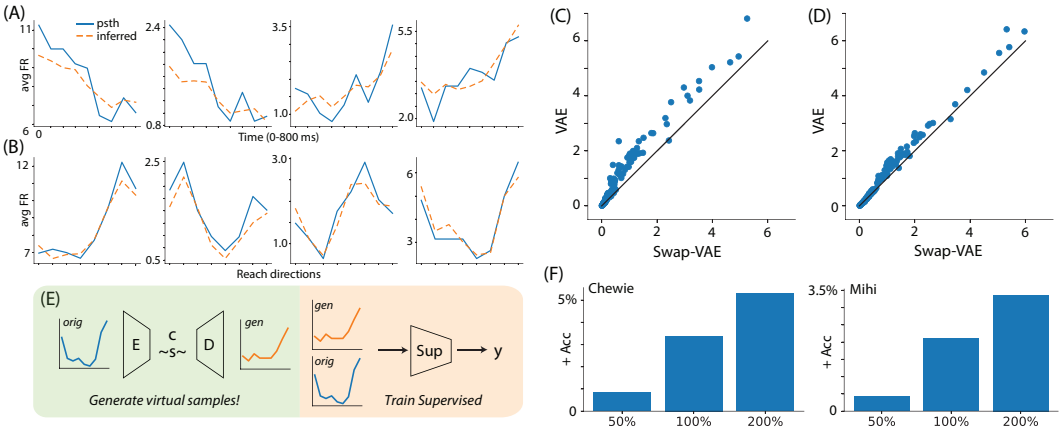

Figure 4: *Testing Swap-VAE's ability to reconstruct and generate new neural activity.* We show the reconstruction of firing rates from example neurons over time (A) and across different reaching directions in (B). We further validate the reconstruction accuracy of our model by comparing the RMSE obtained with our model vs. VAE for different reaching directions (C) and over time (D). (E) shows a sketch of how we generate virtual samples (green) and use them to train a supervised classifier (orange). (F) shows the increase of the classification accuracy as we increase the amount of generated data on Chewie-1 and Mihi-2.

Table 2: *Model ablations.* Accuracy (in %) of different variants of our proposed model.

	no L2	swap-only	no-swap	vanilla-VAE	S-Aug	T-Aug	Ours
acc	71.63	68.36	63.17	63.79	66.77	71.15	73.44
delta-acc	85.62	83.02	83.20	79.10	81.03	83.34	85.38

Next we studied how different augmentations impact overall performance. We tested the spatial-only (S-Aug) and temporal-only (T-Aug) condition separately in Table 2. In this case, we can see that they all perform reasonably well, although they are both worse than our final model where we combine both. As we know, the selection of the data augmentations is critical for the performance of a representation learning model [33, 48, 49]. Our model needs even fewer data augmentation operations than MYOW to achieve a good performance, highlighting the power of our approach.

5 Conclusion

This paper introduces a new self-supervised approach, *Swap-VAE*, for generative modeling of neural activity. Our proposed method leverages a self-supervised alignment strategy to decompose neural activity into different representations that each reveal distinct information about the relationship between neural activity and animal behavior.

In particular, our analysis of neural activity patterns across two different individuals revealed interesting outcomes. We found that the disentanglement in Monkey M was more pronounced than C, both in terms of their decoding accuracies across the content and style spaces and multi-task covariance scores. As we point out in Section 4.2, C and M were trained to make reaches differently: with M being forced to wait and receive an auditory cue before being able to make a movement. In this case, we find that the reach direction and movement are also more decoupled and in Monkey C, where the cue is given and they do not wait, both pieces of information are more entangled. We find that our multi-task covariance score can reveal these differences across generative and discriminative models.

Currently, we only use simple augmentations of neural states like dropout and local temporal shifts. However, other works like [15] use a nearest-neighbor approach to link brain states that are nonlocal or may span different trials. Through combining our approach with this nonlocal view mining strategy, we may be able to build even further invariance into our model's content space. Combining our SSL-backed approach with a sequential encoder model is another exciting line of future research that can further help to extract structure over longer timescales.

References

[1] S. E. de Vries, J. A. Lecoq, M. A. Buice, P. A. Groblewski, G. K. Ocker, M. Oliver, D. Feng, N. Cain, P. Ledochowitsch, D. Millman, *et al.*, "A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex," *Nature neuroscience*, vol. 23, no. 1, pp. 138–151, 2020.

[2] V. Talebi and C. L. Baker, "Natural versus synthetic stimuli for estimating receptive field models: a comparison of predictive robustness," *Journal of Neuroscience*, vol. 32, no. 5, pp. 1560–1576, 2012.

[3] M. N. Shadlen and W. T. Newsome, "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding," *Journal of neuroscience*, vol. 18, no. 10, pp. 3870–3896, 1998.

[4] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of neural science*, vol. 4. McGraw-hill New York, 2000.

[5] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[6] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, no. 5456, pp. 1273–1276, 2000.

[7] C. Pandarinath, K. C. Ames, A. A. Russo, A. Farshchian, L. E. Miller, E. L. Dyer, and J. C. Kao, "Latent factors and dynamics in motor cortex and their application to brain–machine interfaces," *Journal of Neuroscience*, vol. 38, no. 44, pp. 9390–9401, 2018.

[8] J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller, "Long-term stability of cortical population dynamics underlying consistent behavior," *Nature neuroscience*, vol. 23, no. 2, pp. 260–270, 2020.

[9] A. Afshar, G. Santhanam, M. Y. Byron, S. I. Ryu, M. Sahani, and K. V. Shenoy, "Single-trial neural correlates of arm movement preparation," *Neuron*, vol. 71, no. 3, pp. 555–564, 2011.

[10] C. Pandarinath, D. J. O'Shea, J. Collins, R. Jozefowicz, S. D. Stavisky, J. C. Kao, E. M. Trautmann, M. T. Kaufman, S. I. Ryu, L. R. Hochberg, *et al.*, "Inferring single-trial neural population dynamics using sequential auto-encoders," *Nature methods*, vol. 15, no. 10, pp. 805–815, 2018.

[11] D. Zhou and X.-X. Wei, "Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae," *arXiv preprint arXiv:2011.04798*, 2020.

[12] L. Y. Prince, S. Bakhtiari, C. J. Gillon, and B. A. Richards, "Parallel inference of hierarchical latent dynamics in two-photon calcium imaging of neuronal populations," *bioRxiv*, 2021.

[13] M. R. Keshtkaran, A. R. Sedler, R. H. Chowdhury, R. Tandon, D. Basrai, S. L. Nguyen, H. Sohn, M. Jazayeri, L. E. Miller, and C. Pandarinath, "A large-scale neural network training framework for generalized estimation of single-trial population dynamics," *bioRxiv*, pp. 2021–01.

[14] J. Ye and C. Pandarinath, "Representation learning for neural population activity with neural data transformers," *bioRxiv*, 2021.

[15] M. Azabou, M. G. Azar, R. Liu, C.-H. Lin, E. C. Johnson, K. Bhaskaran-Nair, M. Dabagia, K. B. Hengen, W. Gray-Roncal, M. Valko, and E. Dyer, "Mine your own view: Self-supervised learning through across-sample prediction," *arXiv preprint arXiv:2102.10106*, 2021.

[16] G. Loaiza-Ganem, S. M. Perkins, K. E. Schroeder, M. M. Churchland, and J. P. Cunningham, "Deep random splines for point process intensity estimation of neural population data," *arXiv preprint arXiv:1903.02610*, 2019.

[17] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *international conference on machine learning*, pp. 4114–4124, PMLR, 2019.

[18] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[19] M. Mathieu, J. Zhao, P. Sprechmann, A. Ramesh, and Y. LeCun, "Disentangling factors of variation in deep representations using adversarial training," *arXiv preprint arXiv:1611.03383*, 2016.

[20] R. Zhang, S. Tang, Y. Li, J. Guo, Y. Zhang, J. Li, and S. Yan, "Style separation and synthesis via generative adversarial networks," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 183–191, 2018.

[21] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

[22] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

[23] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

[24] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[25] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[26] E. L. Dyer, M. G. Azar, M. G. Perich, H. L. Fernandes, S. Naufel, L. E. Miller, and K. P. Körding, "A cryptography-based approach for movement decoding," *Nature biomedical engineering*, vol. 1, no. 12, pp. 967–976, 2017.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[29] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh, "Disentangling disentanglement in variational autoencoders," in *International Conference on Machine Learning*, pp. 4402–4412, PMLR, 2019.

[30] R. T. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," *arXiv preprint arXiv:1802.04942*, 2018.

[31] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *arXiv preprint arXiv:1506.02216*, 2015.

[32] O. Fabius and J. R. Van Amersfoort, "Variational recurrent auto-encoders," *arXiv preprint arXiv:1412.6581*, 2014.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[35] Z. D. Guo, B. A. Pires, B. Piot, J.-B. Grill, F. Altché, R. Munos, and M. G. Azar, "Bootstrap latent-predictive representations for multitask reinforcement learning," in *International Conference on Machine Learning*, pp. 3875–3886, PMLR, 2020.

[36] X. Chen and K. He, "Exploring simple siamese representation learning," *arXiv preprint arXiv:2011.10566*, 2020.

[37] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

[38] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[39] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[41] Y. Gao, E. Archer, L. Paninski, and J. P. Cunningham, "Linear dynamical neural population models through nonlinear embeddings," *arXiv preprint arXiv:1605.08454*, 2016.

[42] S. Van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem, "Are disentangled representations helpful for abstract visual reasoning?," *arXiv preprint arXiv:1905.12506*, 2019.

[43] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, "On the fairness of disentangled representations," *arXiv preprint arXiv:1905.13662*, 2019.

[44] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, "Variational autoencoders and nonlinear ica: A unifying framework," in *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217, PMLR, 2020.

[45] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[46] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[47] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[48] H. Lee, S. J. Hwang, and J. Shin, "Rethinking data augmentation: Self-supervision and self-distillation," 2019.

[49] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning," *arXiv preprint arXiv:2005.10243*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] To our knowledge, this work will have no negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] After review, a URL to all relevant code and a demo will be included at the beginning of Section 3
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix A
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] After review, a URL to all relevant code and a demo will be included at the beginning of Section 3
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]