Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Complex time-varying systems are often studied by abstracting away from individual components and their dynamics and building a model of the population-level dynamics from the start. However, when building a collective description, it can be easy to lose sight of each individual and how different individuals contribute to the larger picture. Here, we present a novel transformer architecture for learning from time-varying data by building descriptions of both the individual as well as the collective population dynamics. Rather than combining many individuals into our model at the onset, we develop a separable architecture that operates on individual time-series first before passing them forward; this induces a permutation-invariance property and can be used to transfer across systems of different size and order. After demonstrating that our model can be applied to successfully recover complex interactions and dynamics in many-body systems, we apply our approach to populations of neurons in the nervous system. On neural activity datasets, we show that our multi-scale transformer not only yields robust decoding performance, but also provide impressive performance in transfer. Our results show that it is possible to learn from neurons in one animal's brain and transfer the model on neurons in a different animal's brain, with interpretable neuron correspondence across sets and animals. This finding opens up a new path to decode from and represent large collections of neurons.

1 Introduction

Complex systems (such as the brain) contain multiple individual elements (e.g. neurons) that interact dynamically to generate their outputs. The process by which these local interactions give rise to large-scale behaviors is important in many domains of science and engineering, from ecology [1, 2] and social networks [3, 4, 5] to microbial interactions [6] and brain dynamics [7].

A natural way to model the activity of a system is to build a collective or population-level view, where we consider individuals (or channels) jointly to determine the dynamics of the population (Figure 1(A)). In many cases, studying systems from this population-level perspective has provided important insights into collective computations and emergent behaviors [7]; however, it is also possible to lose sight of the contributions of different individuals' dynamics to the final prediction or inference. This is important in many settings where the dynamics of individuals are significant, either due to their different functional roles in the system [8, 9, 6], or due to shift in their dynamics because of sensor displacement or corruption [10, 11, 12]. Moving forward, we need methods that can build good population-level representations while also providing an interpretable view of the data at the individual level.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Here, we present a new framework for modeling time-varying observations of a system which uses dynamic embeddings of individual channels to construct a population-level view (Figure 1(B-C)). Our model, which we dub *Embedded Interaction Transformer* or `EIT`, decomposes multi-channel time-series by first learning rich features from individual time-series before incorporating information and learned interactions across different individuals in the population. One critical benefit of our model is *spatial/individual separability*: it builds a population-level representation from embeddings of individual channels, which naturally leads to channel-level permutation invariance. In domain generalization tasks, this means a trained model can be tested with permuted channels or entirely different number of channels.

To first understand how the model captures different types of interactions and dynamics, we experiment with synthetic many-body systems. Under different types of interactions, we show that our model can be applied to successfully recover the dynamics of known systems.

Next, we turn our attention to recordings from the nervous system [13], where we have access to readouts from populations of neurons in the primary motor cortex of two rhesus macaques that are performing the same underlying center-out reaching motor task. The stability of the neural representations and collective dynamics found in data from these animals [13, 14, 15] makes them an ideal testbed to examine the generalization of our approach. We show that, remarkably, generalization not only occurs across recording sessions (with different sets of neurons) from the same animal, but that we can also transfer across animals through only a simple linear readout. The performance of the linear decoder based on pre-trained weights in the across-animal transfer outperforms many models that are trained and tested on the same animal. This result provides an exciting path forward in neural decoding and highlights the utility of our architecture.

The main contributions of this work are:

- We proposed a novel framework that considers individual dynamics before collective dynamics, and realized it with a multi-stage transformer: *Embedded Interaction Transformer* (`EIT`). `EIT` learns both population representation and individual representation through decoupling the dynamics of individuals and their interactions in multi-variate time-series.
- With this new framework, we proposed methods for generalization across datasets of different input size and ordering. We further proposed to measure the alignments of unknown individuals through the Wasserstein divergence.
- We applied `EIT` to both many-body systems and neural systems recordings. After demonstrating our model's robust decoding ability, we validated its ability to transfer individual dynamics by performing domain generalization across different populations of neurons and finding neuron alignments.

2 Background and Related Work

2.1 Background on Transformers

Self-attention. Transformers have revolutionized natural language processing with the mechanism of self-attention, which helps the model to learn long-range interactions across different elements or tokens in a sequence. Consider a data sequence $X = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N] \in \mathbb{R}^{N \times d}$, consisting of N tokens of d-dimensions. The self-attention mechanism is built on the notion of queries, keys, and values. Let $Q = XW_Q, K = XW_K, V = XW_V$, where $Q \in \mathbb{R}^{N \times d_q}$, $K \in \mathbb{R}^{N \times d_k}$, $V \in \mathbb{R}^{N \times d_v}$. The attention operation can be written as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{1}{\sqrt{d_k}} Q K^T\right) V, \tag{1}$$

where $\mathrm{softmax}(\cdot)$ denotes a row-wise softmax normalization function.

Multi-head self-attention (MSA). Rather than only learning one set of keys, queries, and values for our tokens, MSA allows each head to find different patterns in the data that are useful for inference. Each of the h heads provides a d_v dimensional output, and the outputs from all heads are concatenated into a $d_v \cdot h$ vector, which is further projected to produce the final representation.

The full operation of a transformer of L layers can be written as below:

$$\begin{aligned} Z_0 &= \left[\mathrm{Embed}(\mathbf{x}^1), \mathrm{Embed}(\mathbf{x}^2), \cdots \mathrm{Embed}(\mathbf{x}^N)\right] + \mathbf{E}_{\text{pos}} \\ Z_{\ell+1} &= Z_\ell + \mathrm{MSA}(Z_\ell) + \mathrm{FF}(Z_\ell + \mathrm{MSA}(Z_\ell)),\ \ell = \{0, \ldots, L-1\} \end{aligned} \tag{2}$$

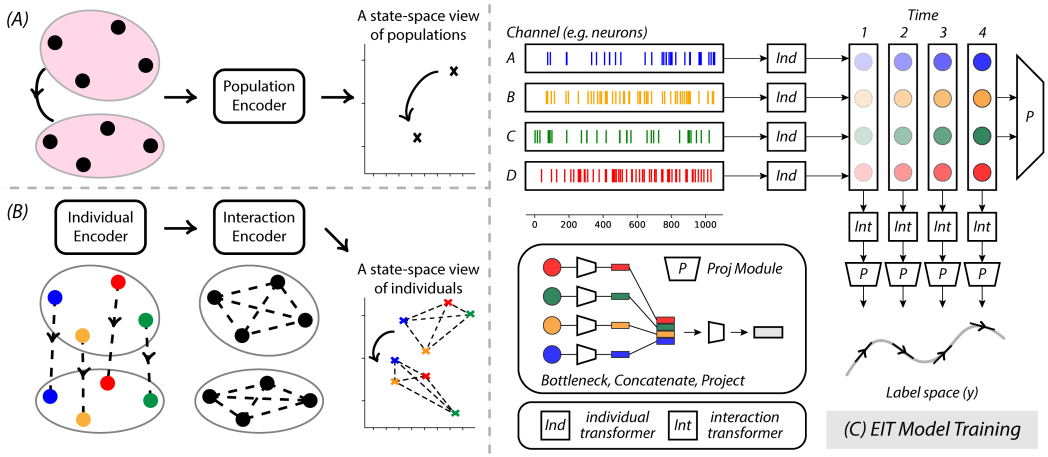

Figure 1: *Embedded Interaction Transformer (`EIT`).* **(A)** A traditional state-space view would treat the collective dynamics as a population right from the beginning, and use a population encoder to learn how the dynamics progress along time, which creates a highly abstracted latent space. **(B)** `EIT` learns individual dynamics with an individual encoder at the beginning. After establishing individual representation, we feed them into an interaction encoder to build a population representation. The two encoders work together to build a representation space that is richer than that of the traditional method. **(C)** The detailed architecture: `EIT` consists of an individual transformer that processes data for each individual, an interaction transformer that processes embeddings at each timepoint, and two projection modules at the end of both transformers.

where Z_0 is the summation of the individual embedding of each data token and the positional embedding $\mathbf{E}_{\text{pos}}$ that helps to retain the positional information, and each transformer layer is the combination of the MSA operation ($\text{MSA}(\cdot)$), the projection ($\text{FF}(\cdot)$), and residual connections.

Transformers for multi-channel time-series. Multi-channel time-series, as a data format that consists of sequences of samples, are a natural choice to model with the transformer architecture.

One common approach of modeling many time-varying data streams with transformers is to combine all channels into a collective representation at the beginning of the model. The population dynamics is thus learnt with the resulting embedding via a temporal transformer for the purpose of inference. There are many complex ways to create this embedding: [16] and [17] extracted embeddings from multivariate physical systems with Koopman operators before feeding the resulting representation into a temporal transformer; [18] used a graph neural network before transformers to embed interconnected-structures to perform skeleton-based action recognition; [19] used convolutional architecture to extract image embeddings before feeding them into transformer.

Another approach is to re-design the attention block such that the attention operation is computed both along the temporal and spatial dimension. Many variants of this 'spatial-temporal' attention block have been shown effective: [20] designed a non-autoregressive module to generate queries for time series forecasting; [21] used stacked spatial-temporal modules for traffic flow forecasting; GroupFormer [22] embed spatial and temporal context in parallel to create a clustered attention block for group activity recognition. While our approach also makes use of the high-level idea of separating spatial (individual) and temporal information, crucially, we restrained the direct computation between the spatially-related attention map and the temporally-related attention map, which yields a representation of the individual which is completely free of spatial interactions.

Spacetime attention in video transformers. With the advances of Vision Transformer [23] as a new way to extract image embeddings, many 'spatial-temporal transformer' architectures are developed in the video domain [24, 25, 26]. Such works provided interesting explorations on how to organize spatial attention and temporal attention with coupled attention blocks [26] or factorized attention blocks [24], and how to create better tokens for videos with tubelet tokenizations [24]. However, as image patches do not possess the channel-specific dynamics that we are interested in, processing videos with spacetime attention presents a very different problem.

2.2 Neural decoding and the challenge of generalization

The brain is an incredibly complex system. Neural activity is guided both by a tight interplay of membrane dynamics and ionic currents [27, 28, 29, 30] as well as interactions with other neurons over larger distributed circuits [31, 32, 33]. Understanding how neurons work together to represent their inputs is an important objective in modern neuroscience.

Accordingly, constructing representations that can explain neural population activity is an active area of research with many existing approaches. Discriminative models produce a representation which is a good predictor of external covariates (e.g. the subject's perception or behavior) or temporally-adjacent activity, and include work such as MYOW [34] and the Neural Data Transformer [35]. Generative approaches instead attempt to model neural activity as driven by latent factors (which might include external covariates, but could be unknown) [36] and make use of switching linear dynamical systems [37] and recurrent neural networks [38]. Recently, SwapVAE [39] emerged as a hybrid of these two approaches, combining a latent factor model of neural activity with self-supervised learning techniques to predict across data augmentations. However, all of these methods produce representations exclusively at the population level.

Learning across recordings or individuals. Generalizing across different neural recordings is a critical but challenging task [40, 41]. The challenge is two-fold: 1) Tracking individual neurons over a long-term recording is nearly impossible, as the detected subset of neurons, even at the same physical location, could shift from time to time [10, 11, 12]; 2) We labels to map the recorded neurons from different subsets or different individuals, as the specific neurons might be completely different.

Many existing approaches attempt to conquer this challenge through the population point of view: [42] proposed a manifold alignment method to continuously update a neural decoder over time as neurons may shift in and out of the recording; [43] proposed an adversarial domain adaptation for stable brain-machine interfaces; [44] proposed a robust alignment of cross-session recordings of neural population activity through unsupervised domain adaptation; [38] fit each set of neurons with its own encoder and used dynamics to inform alignment in the latent space during training. Crucially, these works all rely on a subset of common neurons across recordings which can be matched [42]. However, such a subset of course does not exist across animals. Although there are works assume that the neural activities are intrinsically stochastic [45, 46], we believe it is reasonable to expect that there exists individuals that share similar representations when producing similar behaviour.

3 Methods

3.1 Formulation and Setup

In this work, we consider time-varying prediction from multi-variate time-series observations. Consider $X \in \mathbb{R}^{N \times T}$ as an input datum that measures the dynamics of N individual channels over T points in time. Let x_{ij} denote the value of channel i at the j-th time point, where $1 \leq i \leq N$ and $1 \leq j \leq T$, and let $X^{(i)}$ denote the time series of the i-th channel. We consider time-varying labels $y = \{y_1, \ldots, y_T\}$ where $y_j \in \mathbb{R}^d$ is the label at the j-th time point.

Our aim is to find a function which takes input X to predict target variables y. Instead of jointly considering individual time-series immediately, we instead decompose the model into two functions $f : \mathbb{R}^T \to \mathbb{R}^T$ and $g : \mathbb{R}^N \to \mathbb{R}^N$, so that the optimization problem becomes:

$$\min_{f,g,\mathrm{Proj}} \; \mathbb{E}_X \left[\sum_{j=1}^{T} \ell \Big(\; \mathrm{Proj} \left[g \left(f(X^{(1)})_j, \ldots, f(X^{(N)})_j \right) \right], \; y_j \; \Big) \right] \tag{3}$$

where Proj projects latent representations to predict the labels, and $\ell(\cdot, \cdot)$ is a loss measuring the discrepancy between the labels and the time-varying output from the decoder. In words, f is applied to each time-series independently, while g combines individual embeddings across the population at a single time step. Crucially, this decomposition provides f invariance to the number of channels fed into the model. In practice, we map each x_{ij} to a higher dimension to increase capacity for inference.

3.2 Approach

As shown in Figure 1(B), EIT decomposes collections of temporal data through two transformers to build the representations of the individual time-series and also the population-level interactions.

A temporal-spatial module that disentangles individual and interactions. To separate interactions at the population level from the individual dynamics, we start by building representations from individual time-series. For the i-th channel, we obtain an initial embedding of T temporal tokens as $Z_0^{(i)} = [\text{Embed}(x_{i1}), \text{Embed}(x_{i2}), ..., \text{Embed}(x_{iT})]$, where Embed : $\mathbb{R} \to \mathbb{R}^M$ and $Z_0^{(j)} \in \mathbb{R}^{T \times M}$. We apply a transformer with multi-head attention to the resulting sequence using Equation (2). Let $\widehat{Z}^{(i)} \in \mathbb{R}^{T \times M}$ denote the final embeddings with elements $\hat{z}_{ij}$ obtained from the first transformer in our model and $\widehat{Z} \in \mathbb{R}^{N \times T \times M}$ be the combined embedded output.

After building representations of dynamics at the scale of individual time-series, we then pass the representations into a population-level transformer to capture interactions between all of the individual time-series. To do this, we take the output of the first module $\widehat{Z}$ and slice it along the temporal axis to yield an embedding of each time point in terms of the different channels. Denote this new sequential reslicing of the data at the j-th timepoint as $V_0^{(j)} = [\hat{z}_{1j}, \hat{z}_{2j}, ..., \hat{z}_{Nj}] + \mathbf{E}_{\text{pos}}$, where $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{N \times M}$ is the fixed positional embedding that is dependent on neuron i's identity. After applying the attention block operations in Equation (2) again, we then arrive at our final embedding of our time-series as $\widehat{V} \in \mathbb{R}^{N \times T \times M}$. Let $\hat{v}_{ij}$ be the final representation of the observation x_{ij}.

A projection module that preserves individual identity. To form a population representation with individual representation, we design a projection module that preserves individual identity. The projection module $\text{Proj}(\cdot)$ consists of two parts: 1) For representations of individuals $z_{ij} \in \mathbb{R}^M$ for the i individual and j-th time-point, Proj first bottleneck the individual latents with a function *bottleneck* : $\mathbb{R}^M \to \mathbb{R}$ to reduce the dimensions. 2) To form a population representation, Proj then concatenates the representation across individuals at the j-th time point, and learns another projection *project* : $\mathbb{R}^N \to \mathbb{R}^d$ to inference of label y_j. Through this operation, we could obtain both the population representation that could be used for inference and the individual representation that is affected in the minimal level.

Learning representations through a multi-stage loss. To build an unified representation that can provide both a description at the individual-level and the population-level, we compose a loss function as follows:

$$\mathcal{L}_{total} = \sum_{j=1}^{T} \ell(\text{Proj-Int}([\widehat{v}_{1j}, \ldots, \widehat{v}_{Nj}]), y_j) + \alpha \cdot \ell(\text{Proj-Ind}([\widehat{z}_{1j}, \ldots, \widehat{z}_{Nj}]), y_j), \tag{4}$$

where $\text{Proj-Int}(\cdot)$ and $\text{Proj-Ind}(\cdot)$ denote two projection modules after the population and individual transformers, respectively; ℓ can be set to either classification loss (CrossEntropy) or regression loss (MSE), depending on the type of prediction target we consider; α is a scaling factor that determines how much emphasis is placed on prediction from individual representations in the first half of the network.

Remark: By setting $\alpha > 0$, we train a projection module on the intermediate individual-level embeddings that provides a purely individual-based representation of the population. Empirically, this more flexible architecture also seems to improve stability of training when compared with setting $\alpha = 0$.

3.3 Generalization across domains through linear decoding and functional alignment

A key element of our framework is that, after training, we can use the first part of the network (f) that we have learned for individual-level dynamics, on a new dataset of arbitrary size (number of neurons) and ordering. Here, we describe: (i) the linear probing approach for transfer across different sized or shuffled populations, and (ii) ways to characterize alignment of functional properties of different individual channels that are studied.

Decoding from a new population of different dimension and ordering. In many domains, it is difficult to reliably preserve channels across datasets. Our model instead provides a flexible

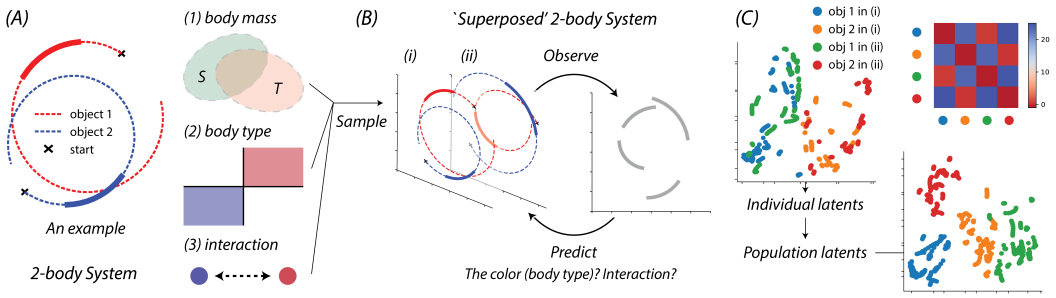

Figure 2: Synthetic Experiments. (A) We show an example of the two-body system, where the system is roughly controlled by three factors: the body mass, the body type, and the body interaction. (B) We create the 'superposed' two-body system by sampling initial conditions from the distribution, observing the produced trajectories over shorter time intervals, and predicting the body type and their interaction from the trajectories. (C) We visualize both the individual latent space and the population latent space. Upper right corner shows how our OT evaluation metric can align and distinguish bodies of the same type (red denotes lower discrepancy).

framework for domain generalization as the first transformer (f) operates only on individual time-series without considering the entire population. Specifically, to transfer to a new dataset, we can learn a new projection $h(\cdot)$ that acts on the concatenated pre-trained embeddings (outputs of f) to build predictions about downstream target variables y:

$$h^* = \arg\min_h \ell(h([\widehat{z}_{1j}, \ldots, \widehat{z}_{Nj}]), y_j), \tag{5}$$

where h is a decoder mapping representations to predicted labels, $\widehat{z}_{ij}$ are the representations formed for the i individual at the j-th time point, and y_j is the label at j-th time point. When we restrict h to be linear, this approach is similar to the evaluation methods in self-supervised learning [47, 48], where a linear decoder is used to evaluate the quality of the frozen representation based on various different downstream tasks.

Functional alignment through representational similarity analysis. Our model also provides a flexible framework to understand correspondences that might exist between channels in different recordings. Recall that for each individual time-series $X^{(i)} \in \mathbb{R}^T$, our model builds a representation space of size $\widehat{Z}^{(i)} \in \mathbb{R}^{T \times M}$. To characterize the similarity across different individual time-series in our datasets, we pass many samples through the encoder to estimate the distribution of $\widehat{Z}^{(i)} \in \mathbb{R}^{T \times M}$. We denote this distribution of individual i as R_i.

We used the Wasserstein divergence ($\mathcal{W}$), a measure of distributional distance motivated by optimal transport (OT) [49, 50] to obtain robust measures of similarity (More details in Appendix 1.1). For one set of distributions $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_{N_1}^{(1)}\}$ from domain (1) and another set of distributions $\{R_1^{(2)}, R_2^{(2)}, \ldots, R_{N_2}^{(2)}\}$ from domain (2), we compute the divergence between all pairs of individuals, which yields a matrix $D \in \mathbb{R}^{N_1 \times N_2}$ where $D[i][j] = \mathcal{W}(R_i^{(1)}, R_j^{(2)})$. When correspondence between channels is known, we can compare against the distance matrix to quantify how well our encoder preserves functional properties.

4 Experiments

4.1 Synthetic experiment: observing superposed many-body systems

We first tested our model on synthetic many-body systems [51]. As shown in Figure.2(A), for a many-body system, a body's observed trajectory is decided by its own properties (its mass and starting point), and its own intended dynamics are further affected by its interaction with other bodies. Furthermore, for certain many-body systems (e.g. k-body systems for $k \geq 3$) the trajectories are chaotic in nature, which is aligned with the stochasticity of neural activity [45, 46]. To test whether our model is able to capture, disentangle, and decode the body dynamics, we create an experiment of 'superposed' many-body systems (as shown in Figure.2(B)), where two separate many-body systems are overlaid with each other as to create a system where some connections exists (bodies within the same system), while some do not (bodies from different systems).

Generating non-chaotic and chaotic systems. We generated the non-chaotic two-body systems and chaotic three-body systems in a relatively stable near-circular way [52], where the trajectories are obtained with the Explicit Runge-Kutta method [53]. We refer the readers to Appendix 2.1 for the details of the set of second-order differential equations that guide the movement.

As shown in Figure.2(B), we create the 'superposition' of many-body systems in the following pipeline: two independent many-body systems are *sampled*, and then their trajectories without additional type or interaction information are *observed*. When training the model, we aim to generate a latent representation for each body at each time that *predicts* the underlying factors that guide the body movements. Note that we use a different distribution of body mass to separate the training set (Source) and testing set (Target), which makes the generated body trajectories purely controlled by two factors: 1) where the body initially starts (as the body **types**); and 2) which body is this body interacting with (as the **interaction**). The model is trained to recover these two factors.

We sample 10 consecutive observations with a gap of 0.2 and 50 consecutive observations with a gap of 0.05, correspondingly, for the two-body system and the three-body system, within time span $[0, 10]$. For each system that consists of different body types and interactions, we generate 500 trials to provide large variability of the movements. We performed a 80/20% train/test split throughout, where the testing set contains unseen combinations of body weights that determine the trajectory.

Model performance. We evaluate the performance of `EIT` by investigating its ability to decode the body types ('Type'), body interaction ('Int'), as well as all possible combinations ('All') of them. The experiments are performed under the domain generalization scheme where the body mass distribution of the testing set (T) is different from that of the training set (S).

Table 1: *Decoding performance of `EIT` on many-body systems when compared to a baseline model (BM).*

	2-body		3-body	
	BM	EIT	BM	EIT
'Int'	93.97	97.28	19.99	91.70
'Type'	97.40	99.93	4.03	37.76
'All'	87.90	94.72	2.03	41.99

We benchmark `EIT` with a transformer with the same depth and amount of heads that considers the population (create the population representation) right from the start of the model. As shown in Table 1, `EIT` consistently outperforms the baseline model (BM) in all cases, where our model showed significant advantage over the baseline when predicting ≈ 100 classes (the 'Type' and 'All' in three-body setting are 90 and 360 classes, respectively). The advantages of the architecture are especially apparent when it comes to the three-body chaotic systems: as the system trajectories provide much higher variability and are highly sensitive to the initial conditions (as body types), our model provide a significantly better inference performance by analyzing the individual dynamics separately to provide a stable representation space for each individuals.

We visualized both the individual and population latent space in Figure 2(C) for the two-body systems. Our individual-module successfully distinguishes different types of bodies, while the population-module further separates different systems by learning the interaction of bodies. We refer readers to Appendix 2.1 for visualizations of the three-body systems. On the top-right corner, we show the OT distance matrix that is produced by our individual-based evaluation method on the test set. The matrix can clearly quantify the similarity between bodies, which tells the different types of individuals.

4.2 Decoding behavior from neural populations

Datasets. The Mihi-Chewie reaching dataset [13] consists of stable behavior-based neural responses across different neuron population groups and animals, and thus is used in previous methods for across-animal decoding [44, 43, 13] and neural representation learning [34, 39]. Mihi-Chewie is a spike sorted dataset where two rhesus macaques, Chewie ('C') and Mihi ('M'), are trained to perform a simplified reaching task to one of eight targets, while their neural activities in the primary motor cortex were simultaneously recorded. The dataset contains two recordings of different sets of neurons for each of the subjects, for a total of four sub-datasets. The Jenkins' Maze dataset [54] contains the primary motor and dorsal premotor cortices activities of a rhesus macaque named Jenkins ('J'). In this dataset, J reaches towards targets while avoiding the boundaries of maze that appears on the screen. Since this dataset provides more complex behaviour trajectories with a complete collection of continuous labels (movement velocity and hand position), we used this dataset as to examine if our model creates an individual representation space that is rich enough to decode continuous targets. All neural datasets are sorted and binned into 100 ms bins by counting the number of spikes each neuron emits during that time frame to generate a time-series for each neuron.

Table 2: *Performance on behavioral decoding from populations of neurons in the motor cortex.*

I. Decoding performance on neural datasets

	Mihi-Chewie ($T = 2$)				*Mihi-Chewie* ($T = 6$)				*Maze* (R^2x100)	
	C-1	C-2	M-1	M-2	C-1	C-2	M-1	M-2	Vel	Pos
MLP	74.22	74.54	78.17	74.42	78.91	90.74	87.90	84.11	60.64	78.59
GRU	75.78	75.46	79.96	74.22	84.72	90.12	85.98	78.81	79.97	94.01
NDT	59.90	58.56	73.02	70.74	64.93	63.73	80.56	71.96	76.01	90.07
NDT-Sup	**80.47**	80.56	83.93	80.79	87.33	**94.29**	**96.83**	91.47	82.02	**94.88**
EIT (T)	76.04	81.25	81.15	71.71	83.33	88.27	93.65	86.82	71.18	87.09
EIT	79.69	**82.41**	**86.51**	**81.61**	**88.36**	92.59	95.24	**91.57**	**82.15**	94.77

II. Generalization performance - trained on one population, tested on another (more in Appendix.3.2)

$\text{NDT}_{\text{retrain}}$(C-2)	75.52	-	77.78	73.06	85.24	-	92.46	86.18	×	×
$\text{NDT}_{\text{retrain}}$(M-2)	74.48	70.37	76.78	-	**86.28**	89.20	91.99	-	×	×
EIT (C-2)	**79.17**	-	82.94	**75.24**	81.42	-	92.33	**91.34**	×	×
EIT (M-2)	78.13	**81.02**	**84.13**	-	84.72	**91.06**	**93.25**	-	×	×

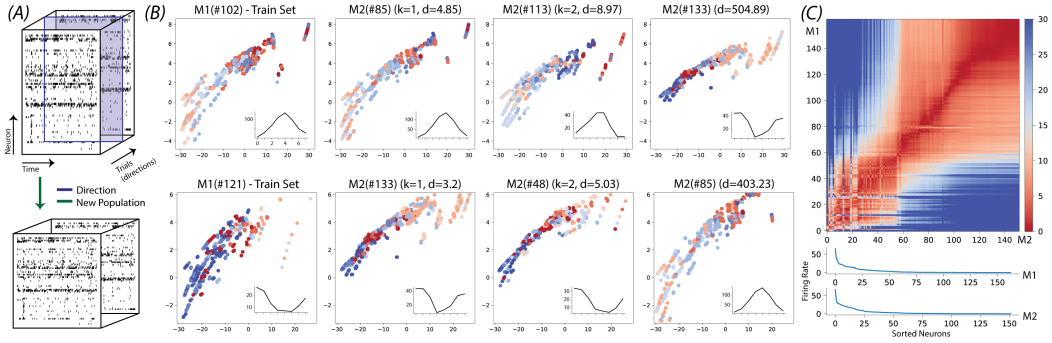

Figure 3: *Experiments on multi-neuron recordings from motor cortex.* (A) A schematic of the two transfer conditions considered for neural recordings. (B) Along each row we display a query neuron, its first and second nearest neighbor, and a high firing neuron that has large divergence. The average firing rate per reach direction are shown in the bottom right of each embedding. (C) The neuron-to-neuron OT distance matrix for M1-M2 which encodes the similarity between the neuron-level embeddings (red denotes similar and blue is dissimilar). Below, we show the firing rates for the population along the rows (top) and columns (below).

Experimental setup. The temporal transformer and the spatial transformer both have a depth of 2 and 6 heads. We set the dimension of single neuron representation to be 16 through the transformer training, and bottleneck it to be 1d when evaluating the activity representation, which makes the activity representation equal to the total number of neurons. We train EIT end-to-end with an Adam optimizer of learning rate 0.0001 for 400 epochs. We benchmark our models' performance with a MLP, bi-directional GRU [55], NDT [35], and a supervised variant of the NDT that we implemented for sequential data decoding. For domain generalization tasks, we re-trained the first-layer of NDT, and compared our model against numbers obtained when training on the same individuals. We also tested EIT 's performance when considering only non-sequential data and compared it with the recent self-supervised methods [34, 39] in Appendix 3.2.

Decoding performance. As shown in Table 2, our model provides strong decoding performance on various benchmarks in terms of both its classification of reach (Mihi-Chewie) and continuous decoding of position and velocity in the regression task studied (Jenkins). For the classification task on Mihi-Chewie, we followed [34, 39] and tested the model's robustness under both shorter sequence setting (T=2) and longer sequence setting (T=6), while for J's Maze we evaluated our model on the full sequence with a regression loss. Both our model and NDT [35] provide good performance, and in some cases, the EIT (T) temporal transformer does quite well on these tasks. Our results suggests that our model is competitive on these diverse decoding tasks from different neural populations in multiple animals.

Generalization task #1: across recording session and animal. After validating the model's decoding performance, we investigate its generalization across different sessions and animals. As the

number of neurons in different recordings varies, we cannot use the model in an end-to-end manner. Instead we train our model on one recording, freeze the weights, and apply the channel-invariant portion of our model to the new data as described in Section 3.3 (Figure 3(A)). In our experiments, we re-train only a final linear classification layer to predict the labels on the new dataset.

Even when tested on new populations of neurons, our model provides impressive overall performance in decoding across sessions and animals (Table 2, bottom). Here, we compare with a re-trained NDT (with a new input layer) and with models trained and tested on the same animal (within domain, top). In these cases, we find that through a linear readout from `EIT` , we can actually outperform many models that are trained from scratch within domain. These results suggests that our temporal transformer has learned information about the firing patterns of neurons that can be transferred across populations in the same and different animals.

To further investigate the underlying factors that contribute to our ability to generalize across populations, we visualized the latent space of individual neurons that have either a closer distribution (in terms of their OT distance) or distant distribution in Figure 3(B). In multiple cases, we found that the embedding of neurons in the training set (M1) had close neighbors in our test set (M2) that had same overall functional properties and tuning (bottom right of each latent embedding). At the same time, neurons that were more distant had orthogonal (or distinct) functional tuning.

To characterize this functional alignment at a population level, we measured the divergence between all pairs of neurons in the train and test set (Figure 3(C)). When sorting neurons in both conditions by their firing rate (bottom), we found that neurons already have a good overall global correspondence in terms of their latent embeddings. This provides some evidence that the learnt latent representation of individuals can accurately reflect how active the neurons are in terms of their overall firing rate and that neurons in both conditions can potentially be aligned. These experiments open up a lot of possibilities for finding functional groups in the individual embeddings and that `EIT` could be used to find aligned neurons across datasets.

Generalization task #2: across behaviours. To test the models ability to transfer when the total class distribution shifts between the train and the test, we tested the model when we fix the neurons but the targets are different between the train (2 classes) and the test (all 8 classes). Again, we can test generalization in this condition by training another linear decoder to decode a different set of labels. Our results in Table 3 suggest that `EIT` can work well when tested in new behavioral conditions and provides significant gaps over other approaches, even when the training data is highly limited.

Table 3: *Performance on Mihi-Chewie reach decoding task when trained on two targets and tested on all eight targets.*

	C-1	C-2	M-1	M-2
MLP	74.28	74.00	83.33	75.81
GRU	74.64	70.67	78.49	79.30
NDT-Sup	75.72	71.00	84.41	82.80
`EIT`	**82.25**	**83.00**	**91.94**	**85.22**

5 Conclusion

In this work, we proposed `EIT` as a model to learn representations of individual and collective dynamics when modeling multi-variate time-series. In our experiments on both synthetic many-body systems and real-world neural systems recordings, we demonstrated that our model not only provides state-of-the-art decoding performance, but its permutation invariance (enabled by the separation of individuals and populations) led to excellent domain generalization performance. These results suggest that our model could open up a new way to decode complex time-series datasets, especially for neural activities.

While our model provides a novel point of view for building representations of collective and individual dynamics, we note that there are also limitations: 1) treating individual systems separately, limits the capacity of individuals (also discussed in [56]) as each individual has limited dimensionality as in the final representation space. 2) our model incorporates individual dynamics in a deterministic way, while many individuals can actually be expressed with a probabilistic model (such as in [57]).

Aside from addressing the above limitations, we also point out that our methods could be extended in a number of directions, including: 1) training the whole network in a self-supervised way, perhaps using recent advances in transformers for vision [58, 59] and contrastive approaches for neural activity [34, 39]; 2) improving domain transfer through structured transport methods [60].

References

[1] A. R. Ives, B. Dennis, K. L. Cottingham, and S. R. Carpenter, "Estimating community stability and ecological interactions from time-series data," *Ecological Monographs*, vol. 73, no. 2, pp. 301–330, 2003.

[2] D. M. Gordon, "The ecology of collective behavior," *PLOS Biology*, vol. 12, no. 3, p. e1001805, 2014.

[3] C. Cattuto, A. Barrat, A. Baldassarri, G. Schehr, and V. Loreto, "Collective dynamics of social annotation," *Proceedings of the National Academy of Sciences*, vol. 106, no. 26, pp. 10511–10515, 2009.

[4] M. Moussaïd, J. E. Kämmer, P. P. Analytis, and H. Neth, "Social influence and the collective dynamics of opinion formation," *PloS one*, vol. 8, no. 11, p. e78433, 2013.

[5] F. L. Pinheiro, F. C. Santos, and J. M. Pacheco, "Linking individual and collective behavior in adaptive social networks," *Physical Review Letters*, vol. 116, no. 12, p. 128702, 2016.

[6] S. van Vliet, C. Hauert, K. Fridberg, M. Ackermann, and A. Dal Co, "Global dynamics of microbial communities emerge from local interaction rules," *PLOS Computational Biology*, vol. 18, no. 3, p. e1009877, 2022.

[7] S. Vyas, M. D. Golub, D. Sussillo, and K. V. Shenoy, "Computation through neural population dynamics," *Annual Review of Neuroscience*, vol. 43, pp. 249–275, 2020.

[8] N. Li, T.-W. Chen, Z. V. Guo, C. R. Gerfen, and K. Svoboda, "A motor cortex circuit for motor planning and movement," *Nature*, vol. 519, no. 7541, pp. 51–56, 2015.

[9] Y. Li and M. Meister, "Functional cell types in the mouse superior colliculus," *bioRxiv*, 2022.

[10] V. S. Polikov, P. A. Tresco, and W. M. Reichert, "Response of brain tissue to chronically implanted neural electrodes," *Journal of Neuroscience Methods*, vol. 148, no. 1, pp. 1–18, 2005.

[11] W. M. Grill, S. E. Norman, and R. V. Bellamkonda, "Implanted neural interfaces: biochallenges and engineered solutions," *Annual Review of Biomedical Engineering*, vol. 11, pp. 1–24, 2009.

[12] D. McCreery, V. Pikov, and P. R. Troyk, "Neuronal loss due to prolonged controlled-current stimulation with chronically implanted microelectrodes in the cat cerebral cortex," *Journal of Neural Engineering*, vol. 7, no. 3, p. 036005, 2010.

[13] 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.

[14] J. A. Gallego, M. G. Perich, S. N. Naufel, C. Ethier, S. A. Solla, and L. E. Miller, "Cortical population activity within a preserved neural manifold underlies multiple motor behaviors," *Nature Communications*, vol. 9, no. 1, pp. 1–13, 2018.

[15] 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.

[16] N. Geneva and N. Zabaras, "Transformers for modeling physical systems," *Neural Networks*, vol. 146, pp. 272–289, 2022.

[17] T. Bai and P. Tahmasebi, "Characterization of groundwater contamination: A transformer-based deep learning model," *Advances in Water Resources*, p. 104217, 2022.

[18] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial temporal transformer network for skeleton-based action recognition," in *International Conference on Pattern Recognition*, pp. 694–701, Springer, 2021.

[19] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2019.

[20] K. Chen, G. Chen, D. Xu, L. Zhang, Y. Huang, and A. Knoll, "Nast: non-autoregressive spatial-temporal transformer for time series forecasting," *arXiv preprint arXiv:2102.05624*, 2021.

[21] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi, and H. Xiong, "Spatial-temporal transformer networks for traffic flow forecasting," *arXiv preprint arXiv:2001.02908*, 2020.

[22] S. Li, Q. Cao, L. Liu, K. Yang, S. Liu, J. Hou, and S. Yi, "Groupformer: Group activity recognition with clustered spatial-temporal transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13668–13677, 2021.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[24] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6816–6826, 2021.

[25] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[26] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in *International Conference on Machine Learning*, PMLR, 2021.

[27] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, p. 500, 1952.

[28] W. M. Kistler, W. Gerstner, and J. L. v. Hemmen, "Reduction of the hodgkin-huxley equations to a single-variable threshold model," *Neural Computation*, vol. 9, no. 5, pp. 1015–1045, 1997.

[29] R. D. Traub, R. K. Wong, R. Miles, and H. Michelson, "A model of a ca3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances," *Journal of Neurophysiology*, vol. 66, no. 2, pp. 635–650, 1991.

[30] M. R. Panahi, G. Abrevaya, J.-C. Gagnon-Audet, V. Voleti, I. Rish, and G. Dumas, "Generative models of brain dynamics–a review," *arXiv preprint arXiv:2112.12147*, 2021.

[31] J. D. Semedo, E. Gokcen, C. K. Machens, A. Kohn, and M. Y. Byron, "Statistical methods for dissecting interactions between brain areas," *Current Opinion in Neurobiology*, vol. 65, pp. 59–69, 2020.

[32] M. G. Perich and K. Rajan, "Rethinking brain-wide interactions through multi-region 'network of networks' models," *Current Opinion in Neurobiology*, vol. 65, pp. 146–151, 2020.

[33] S. L. Keeley, D. M. Zoltowski, M. C. Aoi, and J. W. Pillow, "Modeling statistical dependencies in multi-region spike train data," *Current Opinion in Neurobiology*, vol. 65, pp. 194–202, 2020.

[34] 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.

[35] J. Ye and C. Pandarinath, "Representation learning for neural population activity with Neural Data Transformers," *Neurons, Behavior, Data analysis, and Theory*, Aug. 2021.

[36] F. Pei, J. Ye, D. M. Zoltowski, A. Wu, R. H. Chowdhury, H. Sohn, J. E. O'Doherty, K. V. Shenoy, M. T. Kaufman, M. Churchland, M. Jazayeri, L. E. Miller, J. Pillow, I. M. Park, E. L. Dyer, and C. Pandarinath, "Neural latents benchmark 21: Evaluating latent variable models of neural population activity," *Advances in Neural Information Processing Systems (NeurIPS), Track on Datasets and Benchmarks*, 2021.

[37] J. Nassar, S. W. Linderman, M. Bugallo, and I. M. Park, "Tree-structured recurrent switching linear dynamical systems for multi-scale modeling," in *International Conference on Learning Representations*, 2019.

[38] 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.

[39] R. Liu, M. Azabou, M. Dabagia, C.-H. Lin, M. Gheshlaghi Azar, K. Hengen, M. Valko, and E. Dyer, "Drop, swap, and generate: A self-supervised approach for generating neural activity," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[40] A. E. Urai, B. Doiron, A. M. Leifer, and A. K. Churchland, "Large-scale neural recordings call for new insights to link brain and behavior," *Nature Neuroscience*, pp. 1–9, 2022.

[41] M. Dabagia, K. P. Kording, and E. L. Dyer, "Comparing high-dimensional neural recordings by aligning their low-dimensional latent representations," *Nature Biomedical Engineering*, 2020.

[42] A. D. Degenhart, W. E. Bishop, E. R. Oby, E. C. Tyler-Kabara, S. M. Chase, A. P. Batista, and B. M. Yu, "Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity," *Nature Biomedical Engineering*, vol. 4, no. 7, pp. 672–685, 2020.

[43] A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, L. E. Miller, and S. A. Solla, "Adversarial domain adaptation for stable brain-machine interfaces," 2018.

[44] J. Jude, M. G. Perich, L. E. Miller, and M. H. Hennig, "Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation," *arXiv preprint arXiv:2202.06159*, 2022.

[45] L. Duncker, G. Bohner, J. Boussard, and M. Sahani, "Learning interpretable continuous-time models of latent stochastic dynamical systems," in *International Conference on Machine Learning*, pp. 1726–1734, PMLR, 2019.

[46] T. D. Kim, T. Z. Luo, J. W. Pillow, and C. Brody, "Inferring latent dynamics underlying neural population activity via neural differential equations," in *International Conference on Machine Learning*, pp. 5551–5561, PMLR, 2021.

[47] 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.

[48] 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.

[49] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), vol. 26, Curran Associates, Inc., 2013.

[50] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science," *Found. Trends Mach. Learn.*, vol. 11, no. 5-6, pp. 355–607, 2019.

[51] R. Jastrow, "Many-body problem with strong forces," *Physical Review*, vol. 98, no. 5, p. 1479, 1955.

[52] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[53] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[54] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, S. I. Ryu, and K. V. Shenoy, "Cortical preparatory activity: representation of movement or first cog in a dynamical machine?," *Neuron*, vol. 68, no. 3, pp. 387–400, 2010.

[55] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International conference on machine learning*, pp. 2397–2406, PMLR, 2016.

[56] S. Dutta, T. Gautam, S. Chakrabarti, and T. Chakraborty, "Redesigning the transformer architecture with insights from multi-particle dynamical systems," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[57] A. Shalova and I. Oseledets, "Tensorized transformer for dynamical systems modeling," *arXiv preprint arXiv:2006.03445*, 2020.

[58] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.

[59] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[60] Y. Ge, F. Zhu, R. Zhao, and H. Li, "Structured domain adaptation with online relation regularization for unsupervised person re-id," *arXiv preprint arXiv:2003.06650*, 2020.

[61] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[62] 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," *International Conference on Learning Representations*, 2016.

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]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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] The code and instructions will be included as a URL after the double-blinded review.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the error bars in Appendix.3.2
   (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]
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? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All datasets we used are publicly available
   (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]