

## A Appendix

### A.1 Within-session performance comparison

We include the within-session performance comparison between SPINT and baselines in Table S1. This table is similar to Table 1 in the main paper, but with metrics obtained on EvalAI’s private splits within the held-in sessions. As observed from the table, SPINT also consistently outperforms ZS and FSU baselines on the held-in splits.

	Class	M1	M2	H1
Wiener Filter (WF)	OR	$0.54 \pm 0.01$	$0.27 \pm 0.02$	$0.24 \pm 0.02$
RNN	OR	$0.75 \pm 0.03$	$0.59 \pm 0.07$	$0.51 \pm 0.09$
NDT2 Multi [1]	OR	$0.77 \pm 0.03$	$0.62 \pm 0.03$	$0.68 \pm 0.05$
NDT2 Multi [1]	FSS	$0.77 \pm 0.03$	$0.63 \pm 0.03$	$0.62 \pm 0.04$
WF	ZS	$0.46 \pm 0.06$	$0.15 \pm 0.07$	$0.20 \pm 0.04$
RNN	ZS	$0.52 \pm 0.15$	$0.20 \pm 0.29$	$0.31 \pm 0.13$
CycleGAN + WF [2]	FSU	$0.61 \pm 0.02$	$0.32 \pm 0.03$	$0.15 \pm 0.04$
NoMAD + WF [3]	FSU	$0.64 \pm 0.01$	$0.35 \pm 0.05$	$0.21 \pm 0.06$
<b>SPINT (Ours)</b>	GF-FSU	<b><math>0.77 \pm 0.02</math></b>	<b><math>0.59 \pm 0.01</math></b>	<b><math>0.47 \pm 0.06</math></b>

Table S1: Within-session performance comparison against oracles (OR), few-shot supervised (FSS), few-shot unsupervised (FSU), and zero-shot (ZS) methods. Our SPINT approach belongs to a special class which we termed Gradient-Free Few-Shot Unsupervised (GF-FSU), where models perform adaptation based on few-shot unlabeled data but *without* any parameter updates at test time. Results are reported as mean  $\pm$  standard deviation  $R^2$  across held-in sessions, achieved on EvalAI private held-in splits.

### A.2 Proof of SPINT’s permutation-invariance

Let  $P_R, P_C$  be the row and column permutation matrices of the same permutation  $\pi$  ( $P_C = P_R^\top = P_R^{-1}$  and  $P_C P_R = I$ ). Also let  $X' = P_R X$  and  $(X^C)' = P_R X^C$  be the row-permuted neural windows and row-permuted calibration trials.

Since the ID embedding of each neural unit  $i$  is computed individually from the set of calibration trials for that unit:

$$E_i = \text{IDEncoder}(X_i^C) = \psi(\text{pool}(\phi(X_i^C))), \quad (1)$$

permuting the neural units in the original population (neural windows  $X$  or calibration trials  $X^C$ ) will permute the embedding matrix  $E$  in the exact same order, i.e.,  $E' = P_R E$ .

It follows that:

$$Z' = X' + E' = P_R X + P_R E = P_R (X + E) = P_R Z \quad (2)$$

In other words,  $Z$  is equivariant to the permutation of neural units.

Cross-attention performed on  $Z'$  then becomes:

$$\begin{aligned}
\text{CrossAttn}(Q, Z', Z') &= \text{CrossAttn}(Q, P_R Z, P_R Z) \\
&= \text{softmax} \left( \frac{Q W_K^\top Z^\top P_R^\top}{\sqrt{d_k}} \right) P_R Z W_V \\
&= \text{softmax} \left( \frac{Q W_K^\top Z^\top P_C}{\sqrt{d_k}} \right) P_R Z W_V \\
&= \text{softmax} \left( \frac{Q W_K^\top Z^\top}{\sqrt{d_k}} \right) P_C P_R Z W_V \\
&= \text{softmax} \left( \frac{Q W_K^\top Z^\top}{\sqrt{d_k}} \right) Z W_V \\
&= \text{CrossAttn}(Q, Z, Z)
\end{aligned} \quad (3)$$

where  $\text{softmax}\left(\frac{QW_K^T Z^T P_C}{\sqrt{d_k}}\right) = \text{softmax}\left(\frac{QW_K^T Z^T}{\sqrt{d_k}}\right) P_C$  because an element is always normalized with the same group of elements in the same row regardless of whether column permutation is performed before or after softmax.

Equation 3 concludes Proposition 1 in the main paper.

We note that multi-layer perceptron (MLP), layer normalization, and residual connection are applied row-wise and hence do not affect the overall permutation-invariance property of our SPINT framework.

### A.3 Correlation of attention scores and firing statistics

We ask whether the attention scores SPINT assigns for each neural unit are correlated with its firing statistics. To answer this question, in each held-out calibration window, we measure the average attention scores over  $B$  behavior covariates, and its firing statistics (mean/standard deviation) over the held-out calibration trials, then calculate the Pearson’s correlation between these two quantities using all held-out calibration windows. We show the results in Table S2.

We observe that the attention scores correlate moderately with the mean and the standard deviation of the neural unit’s firing rates, with higher correlation for the standard deviation than the mean, suggesting that SPINT might be extracting neural units that are active (having high mean firing rates) and behaviorally relevant (having high variance throughout the calibration periods where behavior is varied) to pay attention to in behavioral decoding.

	<b>M1</b>	<b>M2</b>	<b>H1</b>
$\rho(\text{attention scores, mean firing rates})$	$0.33 \pm 0.16$	$0.76 \pm 0.03$	$0.51 \pm 0.04$
$\rho(\text{attention scores, standard deviation of firing rates})$	$0.45 \pm 0.16$	$0.87 \pm 0.02$	$0.57 \pm 0.03$

Table S2: Pearson’s correlation between attention scores for each neural unit and that unit mean/standard deviation of firing rates during the held-out calibration periods. Results are reported as the mean correlation  $\pm$  standard deviation across held-out sessions. All  $p$ -values are less than 0.05.

## A.4 Implementation details

### A.4.1 Data preprocessing

For neural activity, we use the binned spike count obtained by unit threshold crossing with the standard bin size of 20ms as set forth by the FALCON Benchmark. We follow FALCON’s continuous decoding setup for all three M1, M2, and H1 datasets, where rather than decoding trialized behavior from the trialized neural activity (often performed in a non-causal manner), we decode behavior at the last step of a neural activity window, mimicking the online, causal iBCI decoding. To construct the length- $W$  neural window at the beginning of each session, we pre-pad the session neural time series with  $(W - 1)$  zeros. We discard the windows whose last time step belongs to a non-evaluated period as defined by FALCON, e.g., inter-trial periods where there is no registered kinematics.

Our IDEncoder infers neural unit identity from trialized calibration trials. As calibration trials vary in length, we interpolate all calibration trials to the same length  $T$ , where  $T = 100$  for M2 and  $T = 1024$  for M1 and H1. We use the Python library `scipy.interpolate.interp1d` with a cubic spline for interpolation. Note that we only perform interpolation for neural calibration trials to synchronize their trial lengths. We still use the raw spike counts for the neural windows, conforming with the continuous decoding setup.

### A.4.2 Behavior output scaling

For M2 and H1, since values of behavior covariates are relatively small, during training we scale the network behavior predictions by a factor of 0.2 and 0.05 for M2 and H1, respectively, effectively asking the model to predict  $5\times$  and  $20\times$  the original behavior values. The MSE loss and  $R^2$  metrics are computed between the scaled predicted outputs and the original ground truth values.

<b>Dropout</b>	<b>0</b>	<b>0.2</b>	<b>0.4</b>	<b>0.6</b>	<b>0.8</b>	<b>DD [0,1]</b>
$R^2$	$0.51 \pm 0.13$	$0.62 \pm 0.10$	$0.63 \pm 0.10$	$0.63 \pm 0.10$	$0.60 \pm 0.09$	$0.64 \pm 0.10$

Table S3: SPINT’s cross-session performance against dynamic dropout and different choices of fixed dropout rates. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions. DD [0,1] stands for dynamic dropout with variable dropout rates between 0 and 1.

<b>DD range</b>	<b>[0, 0.1]</b>	<b>[0, 0.2]</b>	<b>[0, 0.3]</b>	<b>[0, 0.4]</b>	<b>[0, 0.5]</b>	<b>[0,1]</b>
$R^2$	$0.59 \pm 0.07$	$0.59 \pm 0.07$	$0.61 \pm 0.10$	$0.62 \pm 0.07$	$0.63 \pm 0.07$	$0.64 \pm 0.10$

Table S4: SPINT’s cross-session performance across different ranges of dynamic dropout. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions.

<b># heads</b>	<b>4</b>	<b>8</b>	<b>16</b>	<b>32</b>	<b>64</b>
$R^2$	$0.62 \pm 0.08$	$0.63 \pm 0.09$	$0.64 \pm 0.10$	$0.65 \pm 0.11$	$0.64 \pm 0.10$

Table S5: SPINT’s cross-session performance for different cross-attention head counts. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions.

<b># self-attention layers</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
$R^2$	$0.64 \pm 0.10$	$0.63 \pm 0.13$	$0.57 \pm 0.13$	$0.61 \pm 0.10$	$0.60 \pm 0.15$

Table S6: SPINT’s cross-session performance for different number of self-attention layers. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions.

<b># cross-attention layers</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
$R^2$	$0.64 \pm 0.10$	$0.65 \pm 0.10$	$0.65 \pm 0.10$	$0.64 \pm 0.11$	$0.62 \pm 0.13$

Table S7: SPINT’s cross-session performance for different number of cross-attention layers. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions.

<b>Window size</b>	<b>50</b>	<b>100</b>	<b>200</b>	<b>400</b>	<b>600</b>
$R^2$	$0.65 \pm 0.10$	$0.64 \pm 0.10$	$0.64 \pm 0.10$	$0.60 \pm 0.10$	$0.61 \pm 0.09$

Table S8: SPINT’s cross-session performance for different context window sizes. Results are reported as mean  $\pm$  standard deviation across held-out calibration sessions.

#### A.4.3 Inferring neural unit identity

We follow the permutation-invariant framework in [4] for inferring identity  $E_i$  of neural unit  $i$ :

$$E_i = \text{IDEncoder}(X_i^C) = \text{MLP}_2\left(\frac{1}{M} \sum_{j=1}^M (\text{MLP}_1(X_i^{C_j}))\right) \quad (4)$$

where  $M$  is the number of calibration trials,  $X_i^{C_j}$  is the neural activity of the  $j^{\text{th}}$  calibration trial of neural unit  $i$ ,  $\text{MLP}_1$  and  $\text{MLP}_2$  are two 3-layer fully connected networks.  $\text{MLP}_1$  projects the length- $T$  trials to a hidden dimension  $H$ , and  $\text{MLP}_2$  projects the length- $H$  hidden features to length- $W$  neural unit identity output.

#### A.4.4 Behavioral decoding by cross-attention

After neural identity for all units  $E$  is inferred, we add it to the neural window input  $X$  to form the identity-aware neural activity  $Z$ , i.e.,  $Z = X + E$ . We then use the cross-attention mechanism in the latent space to decode last step behavior covariates. Specifically:

$$Z_{in} = \text{MLP}_{in}(Z) \quad (5)$$

$$\tilde{Z} = Q + \text{CrossAttn}(Q, \text{LayerNorm}(Z_{in}), \text{LayerNorm}(Z_{in})) \quad (6)$$

$$Z_{out} = \tilde{Z} + \text{MLP}_{attn}(\text{LayerNorm}(\tilde{Z})) \quad (7)$$

$$Y = \text{MLP}_{out}(Z_{out}) \quad (8)$$

#### A.4.5 Hyperparameters

We include the notable hyperparameters used to optimize SPINT in Table S9. We train and evaluate models for each M1, M2, and H1 dataset separately. We train the models using all available held-in sessions and evaluate on all available held-out sessions. We use Adam optimizer [5] for all training.

	<b>M1</b>	<b>M2</b>	<b>H1</b>
Batch size	32	32	32
Window size	100	50	700
Max trial length	1024	100	1024
Number of IDEncoder layers	3, 3	3, 3	3, 3
Number of cross attention layers	1	1	1
Hidden dimension	1024	512	1024
Behavior scaling factor	1	0.2	0.05
Learning rate	1e−5	5e−5	1e−5

Table S9: Hyperparameters used to train SPINT on the M1, M2, and H1 datasets.

We include representative hyperparameter sweeps demonstrating SPINT’s robustness to hyperparameter choices in Tables S3, S4, S5, S6, S7, S8. This robustness allows SPINT to effectively capture long-range context while maintaining a minimalist architecture without compromising generalizability. All sweep results were obtained on 20% of calibration trials held out from each session of the M1 dataset rather than on the EvalAI test split.

#### A.4.6 Computational resources

SPINT was trained using a single A40 GPU, consuming less than 2GB of GPU memory with batch size of 32 and taking around 12 hours, 5 hours, and 8 hours to finish 50 training epochs for M1, M2, and H1, respectively. We select checkpoints for evaluation at epoch 50 in all M1, M2, and H1 datasets.

## References

- [1] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, 36:80352–80374, 2023.
- [2] Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*, 12:e84296, 2023.
- [3] Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *BioRxiv*, pages 2022–04, 2022.
- [4] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.
- [5] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.