

1 A Appendix

2 A.1 Within-session performance comparison

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

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

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.

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

8 Let P_R, P_C be the row and column permutation matrices of the same permutation π ($P_C = P_R^\top =$
 9 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
 10 windows and row-permuted calibration trials.

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

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

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

15 It follows that:

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

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

17 Cross-attention performed on Z' then becomes:

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

18 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
 19 with the same group of elements in the same row regardless of whether column permutation is
 20 performed before or after softmax.

21 Equation 3 concludes Proposition 1 in the main paper.

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

25 A.3 Correlation of attention scores and firing statistics

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

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

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

35

36 A.4 Implementation details

37 A.4.1 Data preprocessing

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

46 Our IDEncoder infers neural unit identity from trialized calibration trials. As calibration trials vary
 47 in length, we interpolate all calibration trials to the same length T , where $T = 100$ for M2 and
 48 $T = 1024$ for M1 and H1. We use the Python library `scipy.interpolate.interp1d` with a
 49 cubic spline for interpolation. Note that we only perform interpolation for neural calibration trials to
 50 synchronize their trial lengths. We still use the raw spike counts for the neural windows, conforming
 51 with the continuous decoding setup.

52 A.4.2 Behavior output scaling

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

57 A.4.3 Inferring neural unit identity

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

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

59 where M is the number of calibration trials, $X_i^{C_j}$ is the neural activity of the j^{th} calibration trial of
60 neural unit i , MLP_1 and MLP_2 are two 3-layer fully connected networks. MLP_1 projects the length-
61 T trials to a hidden dimension H , and MLP_2 projects the length- H hidden features to length- W
62 neural unit identity output.

63 A.4.4 Behavioral decoding by cross-attention

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

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

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

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

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

70 A.4.5 Hyperparameters

71 We include the notable hyperparameters used to optimize SPINT in Table S3. We train and evaluate
72 models for each M1, M2, and H1 dataset separately. We train the models using all available held-in
73 sessions and evaluate on all available held-out sessions. We use Adam optimizer [5] for all training.

	M1	M2	H1
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 S3: Hyperparameters used to train SPINT on the M1, M2, and H1 datasets.

74 A.4.6 Computational resources

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

79 References

- 80 [1] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-
81 context pretraining for neural spiking activity. *Advances in Neural Information Processing*
82 *Systems*, 36:80352–80374, 2023.
- 83 [2] Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy.
84 Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*,
85 12:e84296, 2023.
- 86 [3] Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza
87 Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E Miller, and Chethan Pandarinath. Stabilizing
88 brain-computer interfaces through alignment of latent dynamics. *BioRxiv*, pages 2022–04, 2022.
- 89 [4] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov,
90 and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.
- 91 [5] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint*
92 *arXiv:1412.6980*, 2014.