PRISM: POPULATION RELATIONAL INFERENCE FOR STREAMING MOTOR DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Intracortical brain–computer interfaces must remain robust to session-to-session neural drift while supporting causal, low-latency streaming inference. We introduce PRISM, Population Relational Inference for Streaming Motor Decoding, which separates session interpretation from per-bin prediction. PRISM uses an unlabeled target-session support recording once to construct and cache a Session interface; each subsequent query bin is decoded causally through two Channel SSM blocks, a Population head, HoloSSM, and a motor readout. The decoder maintains fixed-size recurrent state and requires neither target behavior labels nor test-time gradient updates. On the official Few-shot Algorithms for Consistent Neural Decoding (FALCON) private held-out evaluation, PRISM ranks second among few-shot-unsupervised (FSU) entries on all three tasks in the 20 September 2026 leaderboard snapshot, with held-out R^2 values of 0.424, 0.574, and 0.272 at official dimensionless normalized latencies of 0.0273–0.0413. On 16 held-out dates of the public Indy reaching dataset, the full model achieves date-averaged $R^2 = 0.551$. After one-time support calibration, the 262,782-parameter decoder processes each 20-ms query bin in 0.484 ms on one CPU thread. PRISM therefore converts session-level adaptation into a reusable interface while retaining compact, fixed-state, low-latency causal inference.

1 INTRODUCTION

Intracortical brain–computer interfaces must decode movement while the recorded neural population and its statistics change across sessions (Sussillo et al., 2016; Gallego et al., 2020; Degenhart et al., 2020). A useful decoder therefore needs both a way to interpret a changed recording session and a way to track movement within that session. These computations operate at different time scales: session calibration can use an initial support recording, whereas a streaming predictor must respond to every new neural bin with bounded latency and memory.

The Few-shot Algorithms for Consistent Neural Decoding (FALCON) benchmark evaluates human reach-to-grasp kinematics (H1), monkey reach-and-grasp muscle activity (M1), and monkey finger kinematics (M2) under explicit zero-shot, supervised, and unsupervised information budgets (Karpowicz et al., 2024). For a separate cross-day analysis, we use the Indy recordings from the publicly released *Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology* dataset (O'Doherty et al., 2017); these recordings provide 96-channel primary motor cortex activity across dates. Figure 1 makes the challenge concrete: similar behavior can arise from different neural activity. A decoder must accommodate this change while preserving the temporal information needed for movement prediction.

Prior work provides complementary approaches to this problem. NoMAD aligns latent neural dynamics for reuse across recordings (Karpowicz et al., 2025); SPINT uses unlabeled context and gradient-free adaptation under changes in recorded-unit sets (Le et al., 2025); and POSSM studies efficient causal state-space decoding (Ryoo et al., 2025). Earlier self-recalibration and unsupervised stabilization methods include parameter updates, latent-space alignment, adversarial adaptation, and long-term probabilistic recalibration (Bishop et al., 2014; Li et al., 2011; Farshchian et al., 2019; Ma et al., 2023; Wilson et al., 2026). Building on these directions, we ask how session conditioning and temporal decoding should be organized together: what can be computed once from support, what must be updated per bin, and where should channel histories enter the model?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

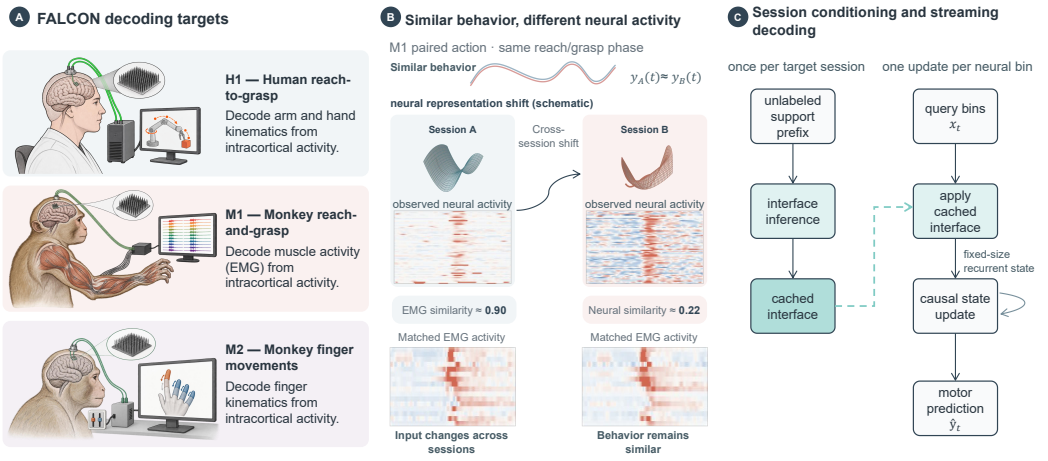

Figure 1: Problem setting. FALCON spans three decoding targets (a); similar behavior can arise from different session activity (b); PRISM infers a session interface from unlabeled support once and then updates a causal decoder once per neural bin (c).

PRISM turns session conditioning into a reusable input interface. It describes each channel using its activity and relationships to the population, compares these descriptors with a fixed source reference, and infers a residual soft channel map with bounded gain and bias corrections. The interface is cached for query decoding. Two communicating channel-wise state-space blocks then maintain local histories before a Population head aggregates their features and a multiscale memory produces motor predictions. The distinction is an explicit separation of session-dependent input interpretation from per-bin state updates, together with channel-level recurrence before population aggregation. Deployment uses neural support without target behavioral labels or gradient updates.

We evaluate the complete system first, then test the mechanisms that explain its operating point. On FALCON, the official held-out R^2 values are 0.424, 0.574, and 0.272 on H1, M1, and M2, placing PRISM second among few-shot-unsupervised (FSU) entries on each task in the 20 September 2026 leaderboard snapshot, with the official dimensionless normalized latency 0.0273–0.0413. On the Indy dataset, the $h = 16$ model reaches $R^2 = 0.551$ across 16 held-out dates at 0.484 ms per 20-ms bin on one CPU thread. Paired controls connect this operating point to the design: target calibration adds $+0.298$ All R^2 and $+0.547$ on Far dates, while removing channel recurrence costs 0.0520 All R^2. Calibration, compact parameters, and streaming latency are evaluated together, including the one-time software startup cost after support acquisition. Our contributions are: (i) a computation schedule that separates reusable session-interface inference from causal recurrence; (ii) a relational, reference-conditioned interface followed by channel history before population aggregation; and (iii) an empirical characterization of the resulting accuracy–resource trade-off through official FALCON evaluation, controlled Indy ablations, and startup, latency, and memory measurements.

2 Problem Setup

For session s, let $x_{s,t} \in \mathbb{R}^C_{\geq 0}$ be the neural count in bin t and $y_{s,t} \in \mathbb{R}^O$ its motor target. Source sessions provide supervision. At deployment, a task-permitted unlabeled support set U_s and the causal query stream are available; target behavior and test-time gradient updates are not.

PRISM first constructs an interface from support and a fixed source reference, then updates a recurrent state:

$$\mathcal{I}_s = \mathcal{C}_\phi(U_s; D_{\text{ref}}), \qquad \xi_{s,t} = \mathcal{F}_\theta(\xi_{s,t-1}, \mathcal{T}_{\mathcal{I}_s}(x_{s,t})), \qquad \widehat{y}_{s,t} = \mathcal{R}_\theta(\xi_{s,t}). \tag{1}$$

In Eq. 1, U_s is the unlabeled support prefix for session s, and D_{ref} is the fixed source-session descriptor used as the matching reference. The operator $\mathcal{C}_\phi$ is the learned session-interface constructor with parameters ϕ; it maps support activity to $\mathcal{I}_s = (\mu_s, \sigma_s, A_s, g_s, b_s)$, containing normalization statistics, the soft channel map, and per-channel gain and bias. The operator $\mathcal{T}_{\mathcal{I}_s}$ applies this cached

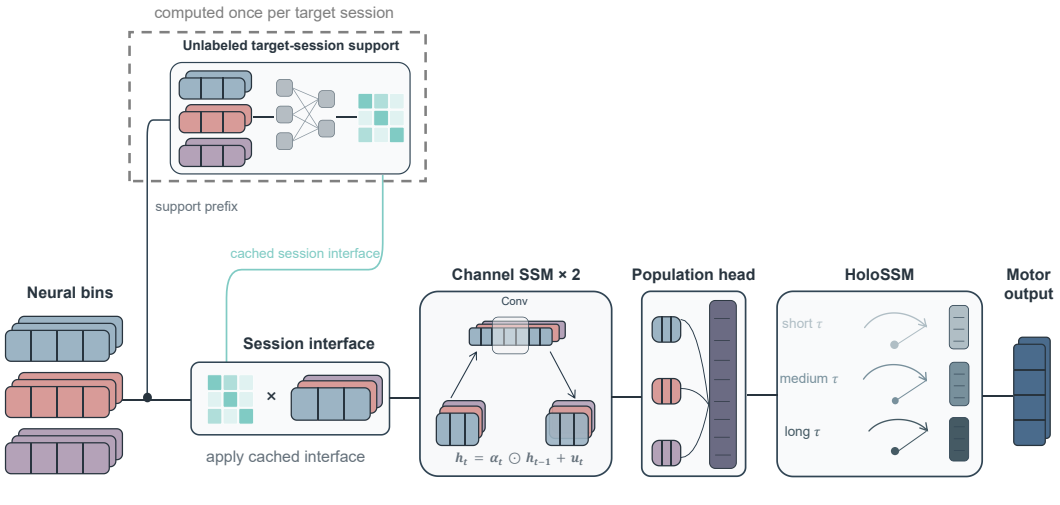

Figure 2: PRISM computation schedule. Unlabeled target-session support is processed once on the cached-support deployment path to infer and cache the Session interface. Each subsequent neural bin follows the causal path through Channel SSM $\times 2$, the Population head, HoloSSM, and Motor output. No target behavior labels or future query observations are used.

interface to a new neural count $x_{s,t}$, producing the calibrated input. The recurrent operator $\mathcal{F}_\theta$ updates all channel and population states, collected in $\xi_{s,t}$, using only the current calibrated bin and the previous state; $\mathcal{R}_\theta$ reads that state to produce the motor estimate $\widehat{y}_{s,t}$. Here θ denotes the shared decoder parameters, C is the fixed number of neural channels, O is the number of motor outputs, and t indexes the causal query bins. Thus $\mathcal{I}_s$ is computed once on the cached-support deployment path, whereas $\xi_{s,t}$ changes once per observed bin. Each task has its own trained model and fixed channel layout; the separate FALCON fallback refresh path and the exact support and state protocols are in Appendix A.5.

3 PRISM

PRISM is organized around a division of labor between session interpretation and streaming prediction. Figure 2 shows the complete causal path. An unlabeled target-session support prefix is first converted into a cached *Session interface*. Once this interface is available, every query bin follows the same path, *Session interface* $\rightarrow$ *Channel SSM* $\times 2$ $\rightarrow$ *Population head* $\rightarrow$ *HoloSSM* $\rightarrow$ *Motor output*. On the cached-support path, the interface is computed once, whereas the recurrent states are updated once per neural bin. Let $x_{s,t} \in \mathbb{R}^C_{\geq 0}$ denote the C-channel spike-count vector in bin t of session s, and let Δt denote the bin duration.

3.1 Session interface: interpreting a target session

The target session can have different channel statistics and a different relationship between channels than the source sessions (Sussillo et al., 2016; Gallego et al., 2020; Degenhart et al., 2020). PRISM handles this change before the recurrent decoder sees the query stream. The support recording contains neural activity but no motor labels. We use it to estimate the target normalization, describe each channel in population context, and match the resulting description to a fixed source reference. The reference is selected once from source-session descriptors and remains fixed for every target session; target observations never determine which source reference is used.

Support normalization and descriptors. We compute support-only statistics after a log transform and apply them to both support and query bins:

$$\bar{x}_t = \big(\log(1 + x_t) - \mu_s\big) \oslash \max(\sigma_s, 10^{-3}), \tag{2}$$

where $\mu_s, \sigma_s \in \mathbb{R}^C$ are the support mean and standard deviation and $\oslash$ denotes elementwise division. This operation removes session-specific scale before any learned matching is performed; it does not use target motor values.

From eligible contiguous support segments, we select $K_{\text{win}} = 48$ short windows. For channel i and window k, the descriptor $d_{i,k}$ contains four types of information: the channel's standardized temporal trace, the standardized population-mean trace, four local activity statistics, and quantiles of lagged associations with the other channels at lags $\{0, 1, 2\}$. The first two terms describe local and population activity, while the lagged terms describe how a channel relates to its neighbors. This gives the interface a population-level view without introducing a separate learned parameter for every channel pair. This shared encoding follows the set-function and permutation-aware modeling literature (Vinyals et al., 2016; Zaheer et al., 2017; Lee et al., 2019). The exact window construction and the resulting 51-dimensional descriptor are specified in Appendix A.1.

Shared support encoding and reference matching. A shared encoder maps every descriptor to a token. Four learned queries then pool the K_{win} tokens belonging to a channel, so the representation is shared across channels but can emphasize different support windows:

$$\begin{aligned} v_{i,k} &= f_\phi(d_{i,k}), \qquad a_{i,k,m} = \mathrm{softmax}_k\left(q_m^\mathsf{T} v_{i,k}/\sqrt{d_e}\right), \\ e_i &= h_\phi\left(\left[\sum_k a_{i,k,1} v_{i,k}; \ldots; \sum_k a_{i,k,4} v_{i,k}\right]\right). \end{aligned} \tag{3}$$

Here f_ϕ and h_ϕ are MLPs, q_m is the m-th learned query, $a_{i,k,m}$ is its softmax weight over support windows, d_e is the token-embedding dimension, and e_i is the resulting embedding for channel i. The source reference is encoded by the same functions. Thus the decoder learns how to compare support descriptions, rather than memorizing a target-session channel ID. In our implementation the fixed reference is the medoid of the source-session descriptor set, the four queries produce a 12-dimensional channel embedding, and the residual coefficient α is learned and bounded in $[0, 1]$. The map, gain, and bias are stored together with (μ_s, σ_s) after the $K_{\text{win}} = 48$ support windows have been processed. This makes the support pass a single explicit computation: no descriptor, assignment, or statistic is recomputed when the query stream advances.

We compare target and reference embeddings with cosine similarity and convert the similarity matrix into a soft channel assignment:

$$S_{ij} = \frac{e_i^\mathsf{T} e_j^{\text{ref}}}{\|e_i\|_2 \|e_j^{\text{ref}}\|_2}, \qquad P = \mathrm{Sinkhorn}(S/\tau_{\text{match}}), \qquad A = (1-\alpha)I + \alpha P. \tag{4}$$

The similarity matrix S measures compatibility between target channel i and reference channel j. The temperature τ_{match} controls the softness of the matching. Sinkhorn normalization produces the approximately doubly stochastic soft assignment P, and A mixes it with the identity map I. The normalization is the standard entropic optimal-transport scaling operation used for soft correspondence rather than a claim of neuron identity recovery (Cuturi, 2013; Peyré & Cuturi, 2019; Courty et al., 2017). The residual form is important: a session with little evidence for reassignment remains close to the nominal channel layout, while a session with a changed population can redistribute information across channels.

The same channel embedding predicts two bounded corrections,

$$g_i = \exp\big(\gamma \tanh(w_g^\mathsf{T} e_i + c_g)\big), \qquad b_i = \beta \tanh(w_b^\mathsf{T} e_i + c_b), \tag{5}$$

where g_i is a positive gain and b_i is an additive bias; w_g, c_g, w_b, c_b are learned parameters, while γ and β set their bounds. The map-mixing coefficient α here is distinct from the per-bin retention gate $\alpha_{i,t}^{(\ell)}$ below. For readability, the session subscript is suppressed on A, g, and b in the detailed equations below. The cached interface and its application to a query bin are therefore

$$\mathcal{I}_s = (\mu_s, \sigma_s, A, g, b), \qquad \widetilde{x}_t = A^\mathsf{T}(g \odot \bar{x}_t + b). \tag{6}$$

On the cached-support path, all quantities in $\mathcal{I}_s$ are computed from the support once; query decoding reuses them; it does not re-estimate the map or update parameters at every bin.

3.2 CHANNEL SSM ×2: LOCAL HISTORY BEFORE MIXING

The calibrated signal next enters two channel-wise selective state-space model (SSM) blocks, drawing on the efficient long-sequence recurrence family developed in modern state-space models (Gu et al., 2022; Smith et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024). Their placement is a central design choice. A conventional decoder can mix all channels first and model only the resulting population vector. PRISM instead gives every channel a private temporal state before the population representation is formed. The model can therefore preserve channel-specific histories while still allowing information to pass between channels through a controlled low-rank message.

For block $\ell \in \{1, 2\}$, $u_{i,t}^{(\ell)}$ is the input feature of channel i, and $h_{i,t}^{(\ell)}$ is its private state. The first block receives the calibrated scalar, $u_{i,t}^{(1)} = \widetilde{x}_{i,t}$; the second receives the output of the first, $u_{i,t}^{(2)} = v_{i,t}^{(1)}$. The final baseline does not concatenate a separate session embedding to these dynamics. Session information has already entered through the calibrated stream. We apply RMSNorm to $u_{i,t}^{(\ell)}$ and split a learned projection into a signal branch and a gate branch. A causal DWConv acts only on the signal branch. The resulting local feature, the input-dependent write size, and the read gate are

$$\begin{aligned}
\left(r_{i,t}^{(\ell)}; q_{i,t}^{(\ell)}\right) &= \mathrm{Linear}^{(\ell)}(\mathrm{RMSNorm}(u_{i,t}^{(\ell)})), \\
c_{i,t}^{(\ell)} &= \mathrm{SiLU}(\mathrm{DWConv}_{\mathrm{causal}}(r_{i,\le t}^{(\ell)})) \odot \mathrm{sigmoid}(q_{i,t}^{(\ell)}), \\
\delta_{i,t}^{(\ell)} &= \Delta t \left[0.5 + \mathrm{clip}(\mathrm{softplus}(W_{\delta}^{(\ell)} c_{i,t}^{(\ell)} + c_{\delta}^{(\ell)}), 0.005, 5.0)\right], \\
\alpha_{i,t}^{(\ell)} &= \exp(-\delta_{i,t}^{(\ell)} / \tau^{(\ell)}).
\end{aligned} \tag{7}$$

The state update and emitted local feature are

$$\begin{aligned}
h_{i,t}^{(\ell)} &= \alpha_{i,t}^{(\ell)} \odot h_{i,t-1}^{(\ell)} + 0.35 \tanh(W_{w}^{(\ell)} c_{i,t}^{(\ell)} + c_{w}^{(\ell)}), \\
o_{i,t}^{(\ell)} &= \mathrm{sigmoid}(W_{o}^{(\ell)} c_{i,t}^{(\ell)} + c_{o}^{(\ell)}), \\
\ell_{i,t}^{(\ell)} &= \mathrm{Out}^{(\ell)}(o_{i,t}^{(\ell)} \odot h_{i,t}^{(\ell)} + W_{\mathrm{skip}}^{(\ell)} u_{i,t}^{(\ell)}).
\end{aligned} \tag{8}$$

Here $\tau^{(\ell)}$ is a learned time-constant vector shared across channels within a block, and $\mathrm{Out}^{(\ell)}$ is a learned output projection. The matrices and biases $W_{\delta}^{(\ell)}, c_{\delta}^{(\ell)}, W_{w}^{(\ell)}, c_{w}^{(\ell)}, W_{o}^{(\ell)}, c_{o}^{(\ell)}, W_{\mathrm{skip}}^{(\ell)}$ are learned time-scale, write, read, and skip projections. The gate α controls state retention, and o controls how much of the retained state is exposed to the output. The bounded write keeps one neural bin from driving the state without bound. The convolution history and parameter bounds are given in Appendix A.2.

The blocks do not keep the channels completely isolated. At each bin they form a rank-R message from all local features and send a gated version back to each channel:

$$\begin{aligned}
m_t^{(\ell)} &= \sum_i \bar{U}_i^{(\ell)} \odot W_{\mathrm{send}}^{(\ell)} \ell_{i,t}^{(\ell)}, \\
v_{i,t}^{(\ell)} &= \ell_{i,t}^{(\ell)} + \eta^{(\ell)} \tanh\Big(W_{\mathrm{recv}}^{(\ell)}(V_i^{(\ell)} \odot m_t^{(\ell)}) + c_{\mathrm{recv}}^{(\ell)}\Big) + \zeta^{(\ell)} W_{\mathrm{global}}^{(\ell)} m_t^{(\ell)}.
\end{aligned} \tag{9}$$

Here $m_t^{(\ell)} \in \mathbb{R}^R$ is the shared message, $W_{\mathrm{send}}^{(\ell)}$, $W_{\mathrm{recv}}^{(\ell)}$, and $W_{\mathrm{global}}^{(\ell)}$ are learned projection matrices; $\bar{U}^{(\ell)}$ and $V^{(\ell)}$ are sender and receiver weights, and $\eta^{(\ell)}, \zeta^{(\ell)}$ are learned gates. The message is recomputed for the current bin, but the channel states and convolution histories remain fixed-size. This ordering is what the channel-history ablation tests: whether temporal information is more useful when modeled before, rather than only after, population aggregation. Each block uses hidden width $h = 16$, expansion factor two, a causal kernel of width three, and a rank-$R = 16$ communication message. The private state is maintained independently for all C channels; only the rank-R message is shared. The first block therefore turns one calibrated scalar per channel into a local history, while the second refines that history before the ordered channel features are handed to the Population head.

3.3 POPULATION HEAD AND HOLOSSM: INTEGRATING THE DECODED STATE

After the second communicating block, the Population head converts the channel features into one vector for motor decoding. It retains the learned channel order and also adds order-independent

summaries:

$$p_t = \mathrm{MLP}([v_{1,t}^{(2)}; \ldots; v_{C,t}^{(2)}]) + 0.2 W_{\mathrm{pool}}[\mathrm{mean}_i\, v_{i,t}^{(2)}; \mathrm{std}_i\, v_{i,t}^{(2)}], \qquad p_t \in \mathbb{R}^{96}. \tag{10}$$

The ordered path preserves task-specific channel structure; the pooled path makes the representation less dependent on a single channel coordinate. No support re-encoding is introduced here because session conditioning has already been completed by $\mathcal{I}_s$.

The Population head feeds HoloSSM, a multiscale complex-valued memory. Its three banks have different decay and oscillation time scales, allowing the decoder to retain both short motor transients and slower context, following the broader recurrent-memory view of sequence modeling (Elman, 1990; Hochreiter & Schmidhuber, 1997; Voelker et al., 2019; Orvieto et al., 2023). With $\nu_t = \mathrm{LN}(p_t)$, and with $D = 96$ denoting the Population-head width, the normalized population vector controls both a bounded amplitude gate a_t and a feature gate g_t: $a_t = \exp(0.35 \tanh(w_a \log r_t + c_a))$ and $g_t = 1 + 0.5 \tanh(W_g \nu_t + c_g)$, where r_t is the RMS magnitude of p_t. This HoloSSM feature gate g_t is distinct from the cached per-channel gain g in Eq. 6. Bank b stores a complex state $H_{t,b}$ and receives the same bounded write z_t:

$$\begin{aligned} z_t &= a_t g_t \odot (W_{\mathrm{re}} \nu_t + \mathrm{i} W_{\mathrm{im}} \nu_t), \qquad H_{t,b} = \rho_b H_{t-1,b} + K z_t, \\ c_t &= [K^H H_{t,1}; \ldots; K^H H_{t,B}], \qquad \widehat{y}_t^{\mathrm{norm}} = W_{\mathrm{out,re}} \Re(c_t) - W_{\mathrm{out,im}} \Im(c_t) + W_{\mathrm{out,mag}} |c_t| + b_{\mathrm{out}}. \end{aligned} \tag{11}$$

Here $\rho_b = \exp(-\Delta t/\tau_b + \mathrm{i}\omega_b \Delta t)$ determines the decay time τ_b and angular frequency ω_b of bank b; i is the imaginary unit, and $W_{\mathrm{re}}, W_{\mathrm{im}}$ are learned real and imaginary write projections while $W_{\mathrm{out,re}}, W_{\mathrm{out,im}}, W_{\mathrm{out,mag}}$ and b_{out} form the readout; K is a fixed complex read/write projection, K^H projects the bank state back to canonical coordinates, and B is the number of memory banks; $\widehat{y}_t^{\mathrm{norm}}$ is the normalized motor prediction. Source-target statistics convert this prediction to task units. The bounded write gates and the initialization ranges are implementation details in Appendix A.3. During training, dropout is applied to the cleaned real and imaginary coordinates; it is disabled at deployment. The Population head has width 96 and feeds three HoloSSM banks. The banks share the write projection but use distinct learned decay and angular frequency parameters, so the same population feature is represented at short, medium, and long time scales. Amplitude and feature gates bound each write; the readout concatenates the three bank states and restores the task-specific motor scale with source target statistics. Consequently, the HoloSSM state is fixed in size and does not grow with the duration of the query stream.

3.4 Training and causal deployment

All interface and decoder parameters are trained jointly on labeled source sessions. For an update batch $\mathcal{B}$, we minimize

$$\mathcal{L}_{\mathcal{B}} = \sum_{(s,t)\in\mathcal{B}} w_{s,t} \|\widehat{y}_{s,t} - y_{s,t}\|_2^2 + \lambda_{\mathrm{comm}} \mathcal{R}_{\mathrm{comm}}, \qquad \lambda_{\mathrm{comm}} = 10^{-5}. \tag{12}$$

The weight $w_{s,t}$ implements the session-balanced, target-energy-scaled loss used in the experiments. The regularizer $\mathcal{R}_{\mathrm{comm}}$ is the mean absolute mixing-weight penalty, and λ_{comm} controls its contribution. AdamW, cosine learning-rate annealing, gradient clipping, and truncated backpropagation through time are used for source training. Training evaluates the diagonal recurrences with an associative affine scan; deployment uses the equivalent serial update. Exact trainer balancing and checkpoint selection are reported in Appendix A.4.

Deployment has two phases. During the support phase, the model computes $\mathcal{I}_s$ and initializes the recurrent state using the protocol-permitted neural observations. During the query phase, each observed bin is normalized and mapped by $\mathcal{I}_s$, passed through both Channel SSM blocks, aggregated by the Population head, written into HoloSSM, and converted to motor units. The state carries information from past observed bins, but the current prediction never uses future observations or target behavior labels. For N query bins, the total cost is

$$C_{\mathrm{stream}}(N) = C_{\mathrm{interface}} + C_{\mathrm{init}} + N C_{\mathrm{step}}, \tag{13}$$

where interface construction and state initialization occur once on the cached-support path per target session and C_{step} updates the fixed-size channel, convolution, and HoloSSM states for one bin. The computation schedule therefore explains both the accuracy controls in Section 4.3 and the measured efficiency trade-offs: session adaptation is amortized, while causal temporal decoding remains bounded per bin.

Method	Access	H1 R^2	H1 Lat.↓	M1 R^2	M1 Lat.↓	M2 R^2	M2 Lat.↓
PRISM	GF-FSU	0.4238 ± 0.0973	0.027	0.5736 ± 0.0647	0.041	0.2725 ± 0.1334	0.034
SPINT	GF-FSU	0.29 ± 0.15	0.140	0.66 ± 0.07	0.130	0.26 ± 0.13	0.130
NoMAD + WF	FSU	0.13 ± 0.10	1.030	0.49 ± 0.03	0.990	0.20 ± 0.10	0.910
CycleGAN + WF	FSU	0.12 ± 0.06	0.160	0.43 ± 0.04	0.070	0.22 ± 0.06	0.090
WF	ZS	0.16 ± 0.03	0.150	0.34 ± 0.06	0.060	0.06 ± 0.04	0.080
RNN	ZS	0.09 ± 0.18	0.020	-0.60 ± 0.45	0.030	-0.07 ± 0.23	0.010

Figure 3: Official FALCON private held-out R^2 comparison. PRISM scores are official evaluator returns; comparison scores are reported in SPINT (Le et al., 2025). GF-FSU denotes unlabeled target calibration without gradient updates; WF denotes a Wiener-filter decoder and RNN a recurrent neural network. The complete point-by-point rendering is in Supplementary Figure S4.

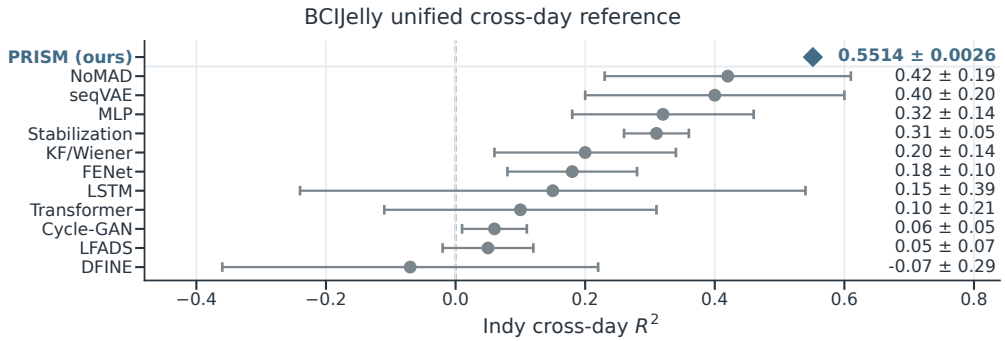

Figure 4: Indy cross-day reference comparison. Gray rows reproduce the unified BCIJelly values (Han et al., 2026); the highlighted PRISM point is our 16-date evaluation.

4 Experiments

We organize the evidence in the same order as the computation. The official FALCON results establish external performance and FSU rank; the Indy result then provides a controlled operating point. We next test the two mechanisms that define PRISM's value: session-level target calibration and channel history before population mixing. Secondary robustness and resource curves are reported in Appendix B.

4.1 External benchmarks and operating point

Figure 3 is the official FALCON private-test comparison (Karpowicz et al., 2024) on H1, M1, and M2. PRISM uses unlabeled target calibration without target gradients and is therefore labelled gradient-free FSU (GF-FSU); here FSU denotes the benchmark's few-shot-unsupervised track and ZS denotes its zero-shot track. The comparison rows are the ZS/FSU/GF-FSU rows reported by SPINT (Le et al., 2025), with PRISM added from its official evaluator return.

As a rank snapshot, the official EvalAI Test pages read on 20 September 2026 place PRISM second among FSU entries on all three tasks: H1 rank 2/3 (13th overall), M1 rank 2/3 (12th overall), and M2 rank 2/3 (10th overall). We report the access-matched FSU rank because the task leaderboards contain different entries and task sizes. The public pages are H1 leaderboard, M1 leaderboard, and M2 leaderboard. ABACUS is above PRISM on H1 and M2, and SPINT is above PRISM on M1.

Table 1: Indy mechanism controls. All entries are date-macro means ± sample standard deviation (SD) across three training seeds under the same split and prefix.

Condition	Near	Far	All
Full PRISM	0.5843 ± 0.0193	0.5186 ± 0.0141	0.5514 ± 0.0026
No target calibration	0.5357 ± 0.0103	-0.0289 ± 0.2991	0.2534 ± 0.1522
Without channel recurrence	0.5430 ± 0.0106	0.4558 ± 0.0061	0.4994 ± 0.0082
Without population recurrence	0.5820 ± 0.0077	0.5114 ± 0.0150	0.5467 ± 0.0112

On the Indy dataset, the complete $h = 16$ system reaches $R^2 = 0.5514 \pm 0.0026$ (All; three seeds), with 0.5843 ± 0.0193 Near and 0.5186 ± 0.0141 Far. Figure 4 locates this operating point, while all controlled claims below use the date-macro split described next.

Computation organization and efficiency. On the cached-support path, PRISM performs session interpretation once, caches the normalization, relational descriptors, assignment map, gain, and bias, and then updates only fixed-size states per neural bin. This gives three measurable advantages at deployment: a fixed parameter/state budget, low steady-state latency, and GF-FSU adaptation without target motor labels or target-time gradient updates.

For the $h = 16$ operating point, the measured deployment costs are 262,782 allocated parameters (1.00 MiB), a one-time 2.9026 s startup after support availability, 0.4843 ms per 20-ms query bin, 63 KiB of recurrent state, and an approximately 3.0 MiB persistent tensor working set. The first 1,000 query bins therefore require 3.3869 s including startup. Detailed timing and memory accounting is in Appendix B. The separate width study reports its own 0.514 ms/bin measurement at $h = 16$.

4.2 Protocols

FALCON uses separately trained H1, M1, and M2 models and official held-out, held-in, and normalized-latency outputs under the benchmark protocol (Karpowicz et al., 2024). Its source-data expansion is a local re-evaluation of released target-session recordings, labelled the *public proxy*. For Indy, we use 96-channel primary motor cortex activity from the released reaching recordings (O'Doherty et al., 2017), binned at 20 ms. The protocol uses a 108-s neural-only prefix for calibration and recurrent-state warm-up, and eight Near plus eight Far test dates; Near and Far are the pre-specified test-date groups, and All denotes all 16 dates. Target-calibration, temporal-memory, width, and source-data studies use seeds 17–19 and 8,190 updates; context uses one frozen checkpoint and masking uses five fixed masks per nonzero rate. For Indy, the model is trained separately for each seed on the six source dates, selected on two validation dates, and evaluated on eight Near and eight Far dates. We use the coefficient of determination $R^2 = 1 - \mathrm{SSE}/\mathrm{SST}$, computed per date and averaged equally (date-macro), so a single long recording cannot dominate the summary. The 108-s prefix is used only for the permitted support and recurrent warm-up; all query predictions then follow the cached interface and causal state update in Figure 2.

4.3 Mechanism evidence

Table 1 connects the full operating point to the two architectural claims. Full target calibration estimates normalization, support descriptors, and the learned interface from the target prefix. NoTargetCalibration keeps the learned interface and the same 108-s state replay but replaces target statistics and support quantities with source values. The channel control removes recurrent state from both Channel SSM blocks while retaining causal convolutions, the population backend, and the complete interface; the population control removes only population recurrence.

Target calibration raises All by $+0.2980$ and Far by $+0.5474$ relative to NoTargetCalibration; every paired seed improves. This is the largest measured mechanism effect and directly supports the staged computation: expensive cross-session interpretation is separated from streaming prediction. Removing channel recurrence lowers All by 0.0520 (Near by 0.0413 and Far by 0.0627) despite retaining the same interface, causal convolutions, and population memory. Removing population recurrence costs only 0.0047 All (Near by 0.0022 and Far by 0.0071). The evidence therefore links

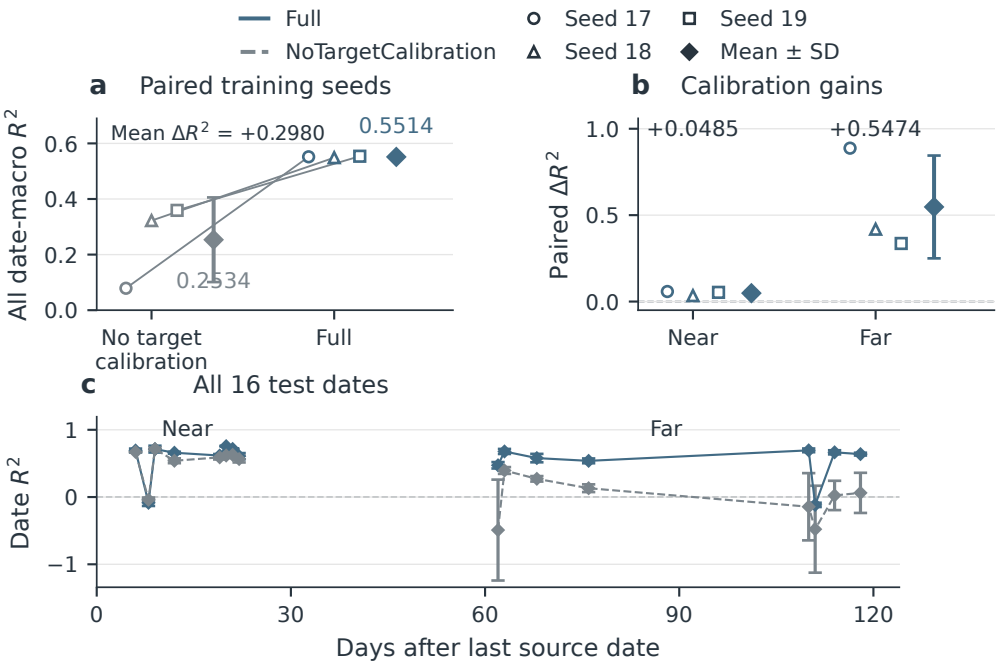

Figure 5: Target-session calibration gains on Indy. Full uses target-prefix calibration and NoTarget-Calibration uses source-fixed quantities; both retain 108-s state replay. Markers show three paired seeds and date-level Near/Far structure.

the two main advantages to their corresponding operations: target calibration sets the input interface, and channel history before population mixing supplies the larger temporal gain. Every ablation uses the same three training seeds and the same date split as the full model; the table reports the seed mean and sample standard deviation of the date-macro score. Figure 5 shows the paired calibration gains. Figure 6 then tests persistence under shared channel masks, while Figure 7 summarizes context length, width, and local source-coverage curves. Supplementary Figures Figures S1–S4 provide the velocity, date-wise channel-loss, context diagnostic, and FALCON benchmark renderings that underlie these summaries.

5 Discussion

PRISM separates cross-session decoding into two computations: inferring an input interface from unlabeled support and tracking the calibrated stream with compact recurrent states. Official FALCON results establish cross-task performance, while controlled Indy experiments attribute the main gains to Target Calibration and channel-level recurrence. Together with measured startup, latency, and memory costs, these results support separating session interpretation from per-bin prediction. Combining official independent evaluation with controlled local testing strengthens the external validity of these conclusions. Fine-grained ablations can clarify individual parameters, but real-world BCI deployment requires treating Target Calibration as a unified system-level interface. Stage-level controls show that this separation yields the largest observed gain, while channel control supports placing local history before population aggregation. The evidence remains open-loop and centered on cross-date transfer within one recording collection, while dense interface matching incurs $O(C^2)$ cost per bin. Refreshing the interface while coordinating it with existing recurrent states remains a natural next step. More broadly, PRISM shows that HoloSSM's fixed-size recurrent state, combined with explicit relational inference, provides an effective route to ultra-low-latency edge deployment and a new architectural paradigm for end-to-end adaptive neuromorphic chips and more general streaming decoders.

AI USE STATEMENT

Generative AI tools were used solely to aid and polish the English writing of this manuscript (e.g., grammar correction, phrasing, and stylistic refinement). They were not used to generate scientific ideas, conduct experimental analyses, write code, or alter data. The authors reviewed all AI-polished text and take full responsibility for the final manuscript, claims, and artifacts.

ETHICS STATEMENT

This study analyzes the publicly released FALCON benchmark and Indy nonhuman-primate reaching recordings; it does not collect new human or animal data or introduce an intervention. We follow the access and use conditions associated with the released datasets. The work concerns offline algorithm evaluation and does not claim clinical deployment or autonomous control of a person or animal.

REPRODUCIBILITY STATEMENT

The main paper defines the causal information boundary and evaluation metrics, while Appendix A specifies the model and streaming state and Appendix B gives the Indy splits, preprocessing, controls, capacity settings, context budgets, and timing protocol. The release bundle should include the training and evaluation source snapshots, configuration files, checkpoints, and saved reports referenced by the appendix. FALCON scores are tied to the official evaluator outputs and task-specific submission protocol.

REFERENCES

William E. Bishop, Cynthia A. Chestek, Vikash Gilja, Paul Nuyujukian, Justin D. Foster, Stephen I. Ryu, Krishna V. Shenoy, and Byron M. Yu. Self-recalibrating classifiers for intracortical brain–computer interfaces. *Journal of Neural Engineering*, 11(2):026001, 2014. doi: 10.1088/1741-2560/11/2/026001. URL https://doi.org/10.1088/1741-2560/11/2/026001.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017. doi: 10.1109/TPAMI.2016.2615921. URL https://doi.org/10.1109/TPAMI.2016.2615921.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Alan D. Degenhart, William E. Bishop, Emily R. Oby, Elizabeth C. Tyler-Kabara, Steven M. Chase, Aaron P. Batista, and Byron M. Yu. Stabilization of a brain-computer interface via the alignment of low-dimensional spaces of neural activity. *Nature Biomedical Engineering*, 4(7):672–685, 2020. doi: 10.1038/s41551-020-0542-9. URL https://doi.org/10.1038/s41551-020-0542-9.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1.

Ali Farshchian, Juan A. Gallego, Joseph P. Cohen, Yoshua Bengio, Lee E. Miller, and Sara A. Solla. Adversarial domain adaptation for stable brain–machine interfaces. In *International*

Conference on Learning Representations, 2019. URL https://openreview.net/pdf/42b1af648df0cfebe9b454256f895b390736f949.pdf.

Juan A. Gallego, Matthew G. Perich, Raeed H. Chowdhury, Sara A. Solla, and Lee E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270, 2020. doi: 10.1038/s41593-019-0555-4. URL https://doi.org/10.1038/s41593-019-0555-4.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024. URL https://openreview.net/pdf?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/pdf?id=uYLFoz1vlAC.

Liyuan Han, Xinrui Yang, Tianyu Zheng, Qizhi Yang, Yitao Qin, Liang Chen, Qinglai Wei, Binjie Hong, Xinhe Zhang, Rui Xiong, Yong Gu, Mu-ming Poo, Bo Xu, Chengyu Li, and Tielin Zhang. Bcijelly: An integrated ecosystem for brain–computer interface research. *bioRxiv*, 2026. doi: 10.64898/2026.08.13.744531. URL https://www.biorxiv.org/content/early/2026/08/18/2026.08.13.744531.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Brianna M. Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, Fabio Rizzoglio, Clay Washington, Thiago Scodeler, Diogo de Lucena, Samuel R. Nason-Tomaszewski, Matthew Mender, Xuan Ma, Ezequiel Matias Arneodo, Leigh R. Hochberg, Cynthia Chestek, Jaimie M. Henderson, Timothy Q. Gentner, Vikash Gilja, Lee Miller, Adam G. Rouse, Robert Gaunt, Jennifer L. Collinger, and Chethan Pandarinath. Few-shot algorithms for consistent neural decoding (falcon) benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76578–76615. Curran Associates, Inc., 2024. doi: 10.52202/079017-2439. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8c2e6bb15be1894b8fb4e0f9bcad1739-Paper-Datasets_and_Benchmarks_Track.pdf.

Brianna M. Karpowicz, Yahia H. Ali, Lahiru N. Wimalasena, Andrew R. Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Daniel B. Rubin, Ziv M. Williams, Sydney S. Cash, Leigh R. Hochberg, Lee E. Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *Nature Communications*, 16(1):4662, 2025. doi: 10.1038/s41467-025-59652-y. URL https://doi.org/10.1038/s41467-025-59652-y.

Trung Le, Hao Fang, Jingyuan Li, Tung Nguyen, Lu Mi, Amy L Orsborn, Uygar Sümbül, and Eli Shlizerman. Spint: Spatial permutation-invariant neural transformer for consistent intracortical motor decoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 25815–25835. Curran Associates, Inc., 2025. doi: 10.52202/085713-0870. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/24c8d2fe8520746f0084174b75b93ebe-Paper-Conference.pdf.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Zheng Li, Joseph E. O'Doherty, Mikhail A. Lebedev, and Miguel A. L. Nicolelis. Adaptive decoding for brain-machine interfaces through bayesian parameter updates. *Neural Computation*, 23(12):

3162–3204, 12 2011. ISSN 0899-7667. doi: 10.1162/NECO_a_00207. URL https://doi.org/10.1162/NECO_a_00207.

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, aug 2023. ISSN 2050-084X. doi: 10.7554/eLife.84296. URL https://doi.org/10.7554/eLife.84296.

Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology, May 2017. URL https://doi.org/10.5281/zenodo.583331.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26670–26698. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/orvieto23a.html.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 02 2019. ISSN 1935-8237. doi: 10.1561/2200000073. URL https://doi.org/10.1561/2200000073.

Avery Hee-Woon Ryoo, Nanda H Krishna, Ximeng Mao, Mehdi Azabou, Eva L Dyer, Matthew G Perich, and Guillaume Lajoie. Generalizable, real-time neural decoding with hybrid state-space models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 51764–51791. Curran Associates, Inc., 2025. doi: 10.52202/085713-1725. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4a40f444c3b0e6486586dabd86fbede4-Paper-Conference.pdf.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf?id=Ai8Hw3AXqks.

David Sussillo, Sergey D. Stavisky, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 7(1):13749, 2016. doi: 10.1038/ncomms13749. URL https://doi.org/10.1038/ncomms13749.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations*, 2016. URL https://iclr.cc/archive/www/2016.html.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 15544–15553. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf.

Guy H. Wilson, Elias A. Stein, Foram Kamdar, Donald T. Avansino, Tsam Kiu Pun, Ronnie Gross, Tommy Hosman, Tyler Singer-Clark, Anastasia Kapitonava, Leigh R. Hochberg, John D. Simeral, Krishna V. Shenoy, Shaul Druckmann, Jaimie M. Henderson, and Francis R. Willett. Long-term unsupervised recalibration of cursor-based intracortical brain–computer interfaces using a hidden markov model. *Nature Biomedical Engineering*, 10(7):1466–1484, 2026. doi: 10.1038/s41551-025-01536-z. URL https://doi.org/10.1038/s41551-025-01536-z.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401. Curran Associates, Inc.,

2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

A Implementation of the Session Interface and Recurrent Decoder

A.1 Support Descriptors and Reference Selection

For each channel, compute the mean μ_s and standard deviation σ_s of $\log(1+x)$ over the support bins permitted by the protocol. Inputs are standardized as

$$\overline{x}_t = (\log(1+x_t) - \mu_s) \oslash \max(\sigma_s, 10^{-3}). \tag{14}$$

The maximum is elementwise. Descriptor windows are drawn from contiguous eligible support segments and do not cross supplied trial boundaries. Candidate windows have length $L = 16$ bins and stride $L/2$; the final complete window of each segment is included if the stride misses it. We select 48 candidates at evenly spaced integer indices in the ordered candidate list. Indices can repeat when fewer than 48 candidates exist. If no segment contains a complete window, the first nonempty segment is zero-padded to length L.

For each selected window, standardized activity is clipped to $[-6, 6]$. Its descriptor for channel i concatenates: (i) its temporal trace, centered and standardized within the window; (ii) the similarly standardized temporal trace of the mean across channels; (iii) the channel's window mean, standard deviation, maximum, and minimum; and (iv) five quantiles of lagged channel associations at each lag in $\{0, 1, 2\}$. The first two groups each have L entries, the third has four, and the last has 15, giving 51 features per channel-window pair.

For lag ℓ, form the two overlapping traces of length $n = L - \ell$, standardize each channel separately over its respective overlap, and compute

$$Q_{ij}^{(\ell)} = \frac{1}{n-1} \sum_{t=1}^{n} Z_{t,i}^{\mathrm{left},\ell} Z_{t,j}^{\mathrm{right},\ell}. \tag{15}$$

Temporal standard deviations use the population convention and a 10^{-3} floor; therefore this finite-sample cross-product is not identified with an exactly bounded Pearson correlation. For each row, we take quantiles at $0.1, 0.25, 0.5, 0.75, 0.9$. The self-entry is excluded at lag zero and retained at nonzero lags. Nonfinite cross-products are replaced by zero before excluding the lag-zero self-entry; a row without eligible entries receives zeros. Finally, each descriptor coordinate is centered and scaled across all channels and selected windows, again with a 10^{-3} floor, and any remaining nonfinite entries are replaced by zero.

The token encoder is Linear(51,48)–SiLU–Linear(48,12)–LayerNorm. Four learned queries aggregate tokens independently for each channel. The concatenated query outputs pass through Linear(48,48)–SiLU–Linear(48,12)–LayerNorm to produce e_i. Cosine normalization uses a 10^{-6} floor. Matching uses temperature $\tau_{\mathrm{match}} = 0.2$ and 16 alternating row/column normalization iterations in log space. The residual coefficient is initialized at $\alpha = 0.1$; log-gain and bias bounds are $\gamma = 0.2$ and $\beta = 0.1$. Both correction predictors are initialized to zero, giving unit gains and zero biases initially.

Reference selection uses the flattened, preprocessed descriptor tensors from source recordings. If these are $d^{(1)}, \ldots, d^{(S)}$, select

$$s^* = \arg\min_s \left\| d^{(s)} - \frac{1}{S} \sum_{r=1}^{S} d^{(r)} \right\|_2^2. \tag{16}$$

The selected descriptor is fixed. Its encoding is recomputed with the shared encoder during training and cached for deployment. No target recording determines the reference.

A.2 Selective Channel Blocks

Each block uses hidden dimension h and expanded dimension $h_{\mathrm{in}} = 2h$. The first block receives a scalar per calibrated channel and the second receives an h-dimensional feature. Transformations

within a block are shared across channels. Let

$$\begin{aligned}[a_{i,t}^{\mathrm{sig}}; a_{i,t}^{\mathrm{gate}}] &= \mathrm{Linear}(\mathrm{RMSNorm}(u_{i,t})),\\ \chi_{i,t} &= \mathrm{SiLU}\Big(\mathrm{DWConv}_{\mathrm{causal}}(a_{i,\le t}^{\mathrm{sig}})\Big) \odot \mathrm{sigmoid}(a_{i,t}^{\mathrm{gate}}).\end{aligned} \tag{17}$$

The depthwise convolution has kernel size $k = 3$ and stores the previous $k - 1$ projected signal vectors. Its filters are shared across channels, with separate filters for expanded feature coordinates. The selective quantities in equation 8 are

$$\delta_{i,t} = \Delta t\,[0.5 + \mathrm{clip}(\mathrm{softplus}(W_\delta \chi_{i,t} + c_\delta), 0.005, 5.0)]\,, \tag{18}$$

$$o_{i,t} = \mathrm{sigmoid}(W_r \chi_{i,t} + c_r), \qquad \tau_{\mathrm{ch}} = \exp(\mathrm{clip}(\vartheta_{\mathrm{ch}}, \log 0.01, \log 5))\,. \tag{19}$$

The time-constant parameters are shared across channels within a block. The session interface conditions the block through the calibrated input; the baseline channel dynamics do not concatenate those embeddings to their inputs.

Population communication uses rank $R = 16$. For raw learned matrices $U, V^{\mathrm{raw}} \in \mathbb{R}^{C\times R}$, the matrices in equation 9 are

$$\overline{U}_{ir} = \frac{U_{ir}}{\sqrt{\sum_j U_{jr}^2 + 10^{-6}}}, \qquad V_{ir} = \tanh(V_{ir}^{\mathrm{raw}}). \tag{20}$$

The sending and global-receive projections have no biases; the private receive projection has a bias. The coefficients η and ζ are sigmoids of learned scalars, initialized with logits -1.5 and -2.5. The shared communication bottleneck has R coordinates; this does not assert that the full nonlinear channel transformation has matrix rank R.

A.3 Population Projection, Write, and Readout

Let $u_{i,t}^{(2)}$ be the output of the second communicating block. The population vector combines channel order and pooled population statistics:

$$p_t = \mathrm{MLP}\Big([u_{1,t}^{(2)}; \ldots; u_{C,t}^{(2)}]\Big) + 0.2 W_{\mathrm{pool}}[\mathrm{mean}_i\, u_{i,t}^{(2)}; \mathrm{std}_i\, u_{i,t}^{(2)}]. \tag{21}$$

The MLP is Linear(Ch,128)–SiLU–Linear(128,D), with $D = 96$; the pooling projection is bias-free, and the pooled standard deviation uses the population convention. This implementation retains a fixed channel layout.

Writing uses a LayerNorm representation $\nu_t = \mathrm{LN}(p_t)$, two bias-free real projections, and bounded gates:

$$r_t = \sqrt{D^{-1}\|p_t\|_2^2 + 10^{-6}}, \qquad a_t = \exp(0.35 \tanh(w_a \log r_t + c_a))\,, \tag{22}$$

$$g_t^{\mathrm{write}} = 1 + 0.5 \tanh(W_g \nu_t + c_g), \qquad z_t = a_t g_t^{\mathrm{write}} \odot (W_{\mathrm{re}} \nu_t + \mathrm{i} W_{\mathrm{im}} \nu_t). \tag{23}$$

The fixed matrix K is obtained by reduced QR factorization of a complex Gaussian matrix using a fixed seed. The baseline uses $E = 128$ and $B = 3$ memory banks. Decay times are initialized logarithmically between 0.02 and 1.5 seconds; frequencies are initialized logarithmically between 0.25 and 14 Hz. The learned decay times are parameterized as $\tau_b = \exp(\mathrm{clip}(\vartheta_b, \log 0.002, \log 20))$. The complex pole in equation 11 is scalar within each bank, and all banks receive the same Kz_t. Recurrent states start at zero before any protocol-defined initialization observations.

For the concatenated cleaned state $c_t \in \mathbb{C}^{BD}$, the normalized prediction is

$$\widehat{y}_t^{\mathrm{norm}} = W_{\mathrm{out,re}} \Re(c_t) - W_{\mathrm{out,im}} \Im(c_t) + W_{\mathrm{out,mag}} \sqrt{\Re(c_t)^2 + \Im(c_t)^2 + 10^{-7}} + b_{\mathrm{out}}. \tag{24}$$

Squares and square roots are elementwise. During training, inverted dropout with probability 0.05 is applied to the cleaned real and imaginary coordinates with the same mask before computing the readout features. Predictions are restored to behavioral units using a mean and standard deviation fitted only to eligible source targets; the standard deviation has a 10^{-4} floor.

A.4 Supervised Objective and Optimization

The communication regularizer averages the two block penalties:

$$\mathcal{R}_{\mathrm{comm}} = \frac{1}{2}\sum_{\ell=1}^{2}\left[\frac{\|\overline{U}^{(\ell)}\|_1}{CR} + \frac{\|V^{(\ell)}\|_1}{CR}\right], \qquad \lambda_{\mathrm{comm}} = 10^{-5}. \tag{25}$$

For a source evaluation group s, define its target energy

$$E_s = \max\left(\frac{1}{N_s}\sum_{t\in\mathcal{Y}_s}\|y_{s,t} - \mu_{y,s}\|_2^2, 10^{-8}\right), \tag{26}$$

where $\mathcal{Y}_s$ contains its eligible training targets and $N_s = |\mathcal{Y}_s|$. This is the sum of output variances in behavioral units. Both trainers use this energy to scale squared prediction errors, but implement group balancing differently.

FALCON trainer. Within a chunk, aggregate errors from streams belonging to the same official evaluation group. If $n_{s,\mathcal{B}}$ eligible bins from group s occur in the chunk, its behavior term is

$$\mathcal{L}^{\mathrm{F}}_{\mathrm{beh},\mathcal{B}} = \frac{1}{|\mathcal{G}_{\mathcal{B}}|}\sum_{s\in\mathcal{G}_{\mathcal{B}}}\frac{\sum_{t\in\mathcal{B}_s}\|\widehat{y}_{s,t} - y_{s,t}\|_2^2}{\max(n_{s,\mathcal{B}}E_s, 10^{-8})}. \tag{27}$$

Groups without eligible bins contribute no behavior term. Source streams are scheduled by recording, with each real stream visited once per epoch in the baseline trainer. The loss groups follow the official session-key reduction. Finite neural bins update state even when the evaluation mask excludes them from supervised loss.

Indy trainer. Source recordings are grouped by date. Each file batch includes one recording per date, cycling shorter file lists when dates have different numbers of recordings. Let n_s^{epoch} denote the number of query bins scheduled for date s in that epoch, including this repetition. The chunk behavior term is

$$\mathcal{L}^{\mathrm{I}}_{\mathrm{beh},\mathcal{B}} = \frac{\max_s n_s^{\mathrm{epoch}}}{64\,|\mathcal{S}_{\mathrm{train}}|}\sum_{r\in\mathcal{B}}\frac{\sum_{t\in\mathcal{B}_r}\|\widehat{y}_{r,t} - y_{r,t}\|_2^2}{E_{s(r)}n_{s(r)}^{\mathrm{epoch}}}. \tag{28}$$

Here r indexes a recording and $s(r)$ its date. The weighting compensates for scheduled duration and file-count differences at the date level. The 108-second neural prefix contributes to conditioning and state initialization, but its behavioral labels do not enter the training loss.

The baseline optimization uses AdamW with learning rate 3×10^{-4}, weight decay 10^{-5}, and a cosine schedule ending at five percent of the initial learning rate. Training uses 64-bin chunks and a 128-bin horizon for TBPTT. Chunk losses, including regularization, are averaged over each update group; a final partial group uses its actual chunk count. Gradients are clipped to norm 1.0. Numerical states are carried forward and detached after each update. The interface and recurrent decoder are trained through the supervised objective, while support statistics and descriptors are computed outside the gradient graph.

Checkpoint selection uses session-macro R^2 on the prescribed FALCON held-in minival or date-macro R^2 on separate Indy validation dates. The FALCON minival is an overlapping debug subset and is not treated as independent generalization evidence. The two trainers' balancing and selection protocols are reported separately rather than described as an identical training procedure.

A.5 Calibration Access and State Execution

Indy. The first 5,400 bins (108 seconds at $\Delta t = 20$ ms) form the neural-only support. Mean, scale, and descriptors are computed from this prefix. Before query processing, the model starts from zero state and replays the support with the inferred interface. Training uses evaluation-mode, gradient-free prefix replay, then detaches the resulting state and resumes training mode for the query. Evaluation and CPU streaming replay use the same prefix information. Query begins after the prefix and state is reset between recordings.

FALCON. For recognized recording tags, the wrapper loads the stored neural mean, scale, and descriptors computed from the supplied calibration recording. Precomputed calibration uses the preprocessing evaluation and validity masks (including target-finiteness eligibility in the loader), falling back to finite neural bins when no selected bins remain; the feature calculations use neural values. Thus calibration is not uniformly a four-second prefix, and its available duration must be reported with each task's evaluation protocol.

The wrapper resets recurrent state at a file reset and carries state across trial boundaries; its trial-completion callback does not clear state. Supplied observation calls advance the same recurrence as prediction calls. Stored calibration statistics alone do not cause a replay of the separate calibration recording.

For an unrecognized tag, initial predictions use source fallback statistics and the reference support. The wrapper accumulates the first 200 observed neural bins, constructs new statistics and descriptors, and refreshes the interface when that count is reached. It retains the existing recurrent state at this refresh and does not replay those bins under the new map. Subsequent predictions reuse the refreshed interface. This fallback is a causal online calibration path, distinct from the precomputed-record path and the Indy prefix-replay protocol.

State and computational cost. For batch size one, two channel blocks with hidden size h, expanded size h_{in}, convolution width k, and B complex banks of width E store

$$N_{\mathrm{state}} = 2Ch + 2C(k-1)h_{\mathrm{in}} + 2BE \tag{29}$$

real-valued elements. This counts channel states, convolution histories, and real/imaginary population states. It excludes parameters, the cached session map, descriptors, and raw calibration storage. For the Indy configuration ($C = 96$, $h = 16$, $h_{\mathrm{in}} = 32$, $k = 3$, $B = 3$, $E = 128$), this is 16,128 FP32 elements, or 64,512 bytes.

The dense session map occupies C^2 entries and its online application has $O(C^2)$ arithmetic cost. The state and per-step work are constant with respect to elapsed history for fixed dimensions, not constant with respect to channels or width. Equation 13 accounts for one interface construction and initialization event; additional calibration events incur their corresponding construction cost.

B Additional Experimental Details

B.1 Indy Data, Targets, and Scoring

We aggregate all supplied disjoint sorted and unsorted spike-unit rows within each of the 96 named M1 electrodes. Counts use nonoverlapping 20-ms bins. Cursor velocity is formed from backward position differences over the corresponding bin, in mm/s, without future interpolation or temporal smoothing. The first 5,400 bins supply neural conditioning and state initialization; their behavioral targets are excluded from fitting and scoring. The date assignments are listed in Table 2; the source date 30 September contributes two recordings and each other listed date contributes one.

Table 2: Indy date split. Dates use YYYY-MM-DD notation.

Split	Recording dates
Source	2016-09-15, 2016-09-16, 2016-09-21, 2016-09-27, 2016-09-30, 2016-10-05
Validation	2016-10-06, 2016-10-07
Near	2016-10-11, 2016-10-13, 2016-10-14, 2016-10-17, 2016-10-24, 2016-10-25, 2016-10-26, 2016-10-27
Far	2016-12-06, 2016-12-07, 2016-12-12, 2016-12-20, 2017-01-23, 2017-01-24, 2017-01-27, 2017-01-31

For a test date d, the reported metric is

$$R_d^2 = 1 - \frac{\sum_{t,o}(y_{d,t,o} - \widehat{y}_{d,t,o})^2}{\max\left(\sum_{t,o}(y_{d,t,o} - \overline{y}_{d,o})^2, 10^{-12}\right)}. \tag{30}$$

We average R_d^2 equally across dates within each split. All test dates, including dates with negative R^2, remain in each aggregate. The source-training fit and later resource analyses reuse the established recordings after the initial baseline evaluation.

B.2 Baselines and Replicated Studies

Ridge uses the current and previous nine normalized neural bins, giving 960 input coordinates and an intercept. Source fitting balances dates and selects the regularization coefficient from $\{1, 10, 100\}$ using validation dates; the selected coefficient is 100. Its 1,922 parameters include the two output intercepts. Its streaming buffer stores ten 96-channel input vectors.

The replicated target-calibration, temporal-memory, and width studies use seeds 17, 18, and 19 with the same source dates, validation dates, 30-epoch budget, and 8,190 optimizer updates. For target calibration, the Full condition fits target-prefix normalization and interface quantities from the 108-second prefix. NoTargetCalibration uses source-fixed statistics and support quantities, retains the learned interface and target-prefix replay, and performs no target calibration update. Paired differences are computed within seed before averaging across the three seeds; the reported standard deviations are sample standard deviations across seeds.

B.3 Timing and Parameter Accounting

Local streaming measurements use an AMD Ryzen 7 9700X, PyTorch 2.11, FP32, one CPU thread, and batch size one. Each timed repeat processes the first 1,000 query bins of `indy_20161011_03.mat` after its 108-second prefix. Model order alternates across three repeats. The reported latency is the median total prediction time divided by 1,000, including neural preprocessing and output rescaling. This is a mean per-bin throughput measurement within each repeat, not a tail-latency or closed-loop performance measurement. Streaming outputs are compared with chunked inference from the corresponding saved model.

For the Full $h = 16$ checkpoint, the measured target-session path is 0.2969 s for support-feature construction, 0.0049 s for interface compilation, and 2.6007 s for state replay, for a one-time startup/calibration cost of 2.9026 s. The first 1,000 query bins add 0.4843 s, giving 3.3869 s of compute from available support samples through that query segment. These values exclude physical acquisition of the 108-s support prefix and model-weight loading; they are local CPU measurements and are separate from the official FALCON normalized-latency metric.

Persistent deployment memory is accounted separately from process RSS. In FP32, the allocated weights occupy approximately 1.00 MiB, the two 96-by-48-by-51 support descriptor buffers occupy approximately 1.79 MiB, the cached session map and constants occupy approximately 0.2 MiB, and the $h = 16$ recurrent state occupies 63 KiB. The resulting persistent tensor working set is approximately 3.0 MiB when raw support samples are released after descriptor construction. The measured run-level peaks (approximately 2.98 GB process RSS and 455 MB NVIDIA CUDA allocator memory) include framework, dataset, and allocator overhead and are not deployment-memory claims.

Reported parameter counts include all allocated parameters, including registered compatibility parameters unused by the forward path. Full has 262,782 allocated and 261,234 active parameters; Small has 158,190 and 156,642; Large has 479,262 and 477,714. State counts include convolution histories and real/imaginary memory states, and exclude weights, session maps, support descriptors, and raw calibration observations.

B.4 Capacity and Context Evaluation

The capacity study preserves the baseline training implementation, changing only the channel hidden-width argument and its dependent layer dimensions. Each width and seed completes 8,190 updates over 30 epochs; the checkpoint is selected independently from the validation dates. Recorded training plus per-epoch validation times are reported only as provenance, because the runs occurred separately and do not establish a training-speed advantage for any width.

The context study uses the original $h = 16$ checkpoint without parameter updates or checkpoint reselection. Only observations within the chosen prefix determine the normalization and descriptor tensor. The resulting interface remains fixed throughout the full 108-second replay and subsequent

query. The 108-second condition reproduces the original date scores within 10^{-5} and preserves every date's scored sample count. This 108-second point is a one-frozen-checkpoint context study; the three-seed Full PRISM value in the main text is the replicated operating point. Prefix-isolation and streaming/chunked consistency checks were performed on two real recordings before full evaluation. Because normalization and learned interface construction change together, the curve in Table 7 describes their combined sensitivity to context.

B.5 Width-Dependent Streaming Resources

Table 3: Resources for the Indy width study. CPU timing uses the same single-thread, batch-one, FP32 measurement batch across widths and reports the median of three alternating repeats.

Width	Parameters	Recurrent state (KiB)	CPU ms/bin
$h = 8$	158,190	33	0.478
$h = 16$	262,782	63	0.514
$h = 32$	479,262	123	0.625

Compared with width 16, width 8 reduces allocated parameters by 39.8% and recurrent state by 30 KiB. Width 32 increases allocated parameters by 82.4% and measured per-bin latency by 21.6%. The width-study timings in Table 3 come from a separate measurement batch.

B.6 Persistent Channel-Masking Protocol

The masking study uses one Full checkpoint and one separately trained StandardizationOnly checkpoint. Nonzero removal rates are 25%, 50%, and 75%, with mask seeds 101, 202, 303, 404, and 505. Full and StandardizationOnly receive identical channel masks. The mask remains fixed throughout a recording, including the 108-second prefix and subsequent query. Input tensors retain their 96-channel shape. Only retained channels contribute to prefix normalization statistics and population descriptor construction. Missing normalized inputs and their descriptor rows are filled with zeros. Weights remain frozen, and the session interface is rebuilt for each masked recording. All 16 test dates and their original query bins remain in the aggregates.

For each mask, scores are averaged over dates. The reported mean and sample SD summarize the five resulting date-macro scores. Paired model differences are formed within each mask before summarization. The clean condition occurs once per model. The saved clean-condition checks report a maximum absolute prediction difference of 1.8311×10^{-4} over the 16 dates relative to the reference path. This is a numerical consistency check, not an additional accuracy sample.

B.7 Source-Session Expansion Protocol

The source-data study evaluates three nested subsets obtained from sorted source-session identifiers: 5/9/13 sessions for H1, 2/3/4 for M1, and 3/5/7 for M2. Every task–scale combination uses seeds 17, 18, and 19, giving 27 runs. Each run completes 30 training epochs and 8,190 optimizer updates. The checkpoint maximizes held-in minival session-macro R^2; ties use the earliest epoch. Selected checkpoints can consequently have different update counts. The source subsets use one fixed ordering, rather than repeated random draws of source sessions.

Evaluation uses the public target-session calibration proxy, aggregated equally over sessions. Table 4 gives each training seed. Table 5 records overlapping minival scores for provenance; these are checkpoint-selection diagnostics.

B.8 Secondary controls and resource curves

The following analyses support the main mechanism chain without changing the headline operating point. They use the local Indy or FALCON public-proxy protocols described in the corresponding subsections.

Table 4: Public-proxy R^2 for each source-data run.

Task	Scale	Seed 17	Seed 18	Seed 19
H1	Small	0.2162	0.2027	0.2338
H1	Medium	0.3617	0.3525	0.3488
H1	Full	0.3780	0.4024	0.4020
M1	Small	0.4944	0.4920	0.4892
M1	Medium	0.4493	0.4741	0.5185
M1	Full	0.4853	0.4859	0.5083
M2	Small	0.0886	0.0855	0.1511
M2	Medium	0.2434	0.1704	0.2211
M2	Full	0.2557	0.2144	0.2737

Table 5: Overlapping held-in minival means across three seeds for the source-data study. These values are not independent test performance.

Task	Small	Medium	Full
H1	0.469	0.802	0.986
M1	0.654	0.767	0.804
M2	0.302	0.706	0.848

Persistent channel loss. We remove 25%, 50%, or 75% of the 96 Indy channels with five shared masks per rate, applying each mask to both support and query streams. The interface is rebuilt from surviving channels and weights remain frozen. Full obtains All R^2 of 0.3874 ± 0.0325, 0.3062 ± 0.0676, and 0.1086 ± 0.0385 at the three rates, compared with 0.3697 ± 0.0497, 0.2830 ± 0.0686, and 0.0928 ± 0.0336 for StandardizationOnly (Table 6). Figure 6 shows the shared-mask outcomes.

Table 6: Persistent channel loss on Indy. Values are All R^2 means $\pm$ sample SD across five masks; the clean condition is evaluated once.

Removed	Full	StandardizationOnly	Paired difference
0%	0.5518	0.5513	0.0005
25%	0.3874 ± 0.0325	0.3697 ± 0.0497	0.0177 ± 0.0211
50%	0.3062 ± 0.0676	0.2830 ± 0.0686	0.0232 ± 0.0103
75%	0.1086 ± 0.0385	0.0928 ± 0.0336	0.0157 ± 0.0113

Context and capacity. A frozen $h = 16$ checkpoint reaches All $R^2 = 0.4521$, 0.5114, 0.5472, and 0.5518 when interface context is 4, 16, 64, and 108 s; all conditions retain the 108-s recurrent warmup. The width study gives All scores 0.5526 ± 0.0057, 0.5514 ± 0.0026, and 0.5699 ± 0.0015 for $h = 8, 16, 32$, respectively, with 158,190, 262,782, and 479,262 allocated parameters; the corresponding three-seed values are listed in Table 8.

Source-session expansion. On the FALCON public proxy, nested source subsets improve H1 and M2 as coverage grows while M1 remains near 0.49:

	Small	Medium	Full
H1	0.218 ± 0.016	0.354 ± 0.007	0.394 ± 0.014
M1	0.492 ± 0.003	0.481 ± 0.035	0.493 ± 0.013
M2	0.108 ± 0.037	0.212 ± 0.037	0.248 ± 0.030

The source subsets and checkpoint-selection details are given in Appendix B.7; Figure 7 places these proxy curves beside the Indy width and context curves.

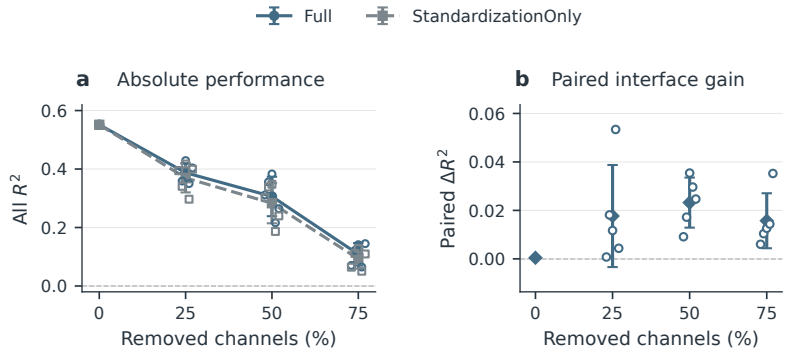

Figure 6: Learned session conditioning under persistent channel loss. Each mask persists through calibration and query decoding.

Table 7: Indy interface-context curve for one frozen checkpoint.

Context (s)	Near	Far	All
4	0.4743	0.4299	0.4521
16	0.5405	0.4823	0.5114
64	0.5834	0.5109	0.5472
108	0.5888	0.5148	0.5518

Table 8: Three-seed width study on Indy.

Width	Parameters	Near	Far	All
$h = 8$	158,190	0.5912 ± 0.0088	0.5141 ± 0.0064	0.5526 ± 0.0057
$h = 16$	262,782	0.5843 ± 0.0193	0.5186 ± 0.0141	0.5514 ± 0.0026
$h = 32$	479,262	0.6123 ± 0.0040	0.5275 ± 0.0021	0.5699 ± 0.0015

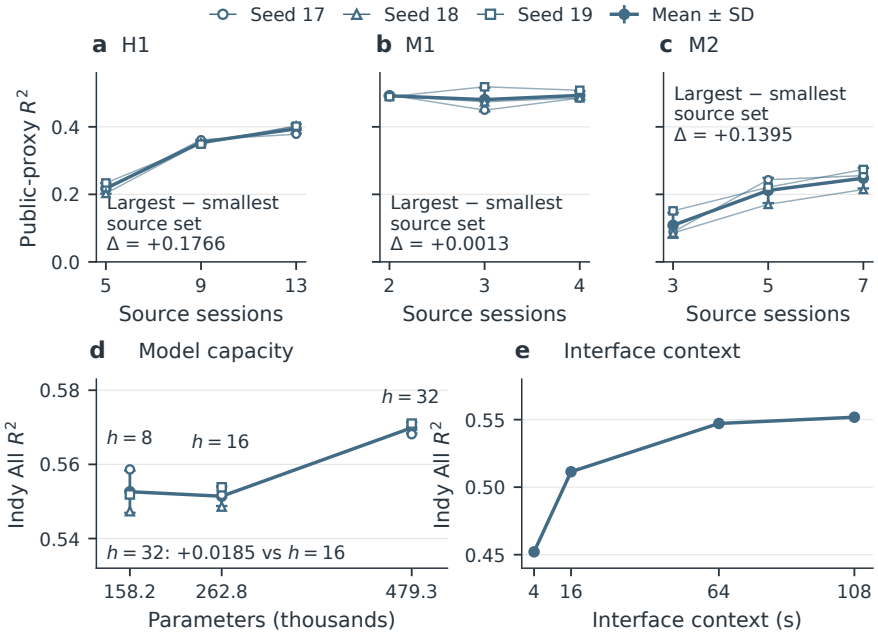

Figure 7: Secondary resource controls. Panels (a–c) are local FALCON public-proxy source-coverage curves; panels (d–e) are local Indy width and context curves.

C SUPPLEMENTARY FIGURES

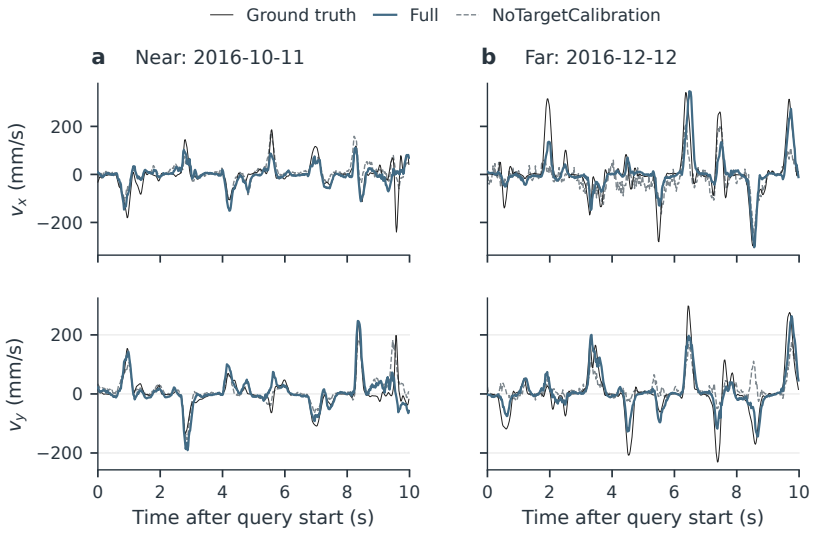

Figure S1: Velocity rendering for the Indy evaluation. The caption should identify the displayed recording or date, the query interval, the velocity components, and the meanings of all lines, markers, and shaded regions.

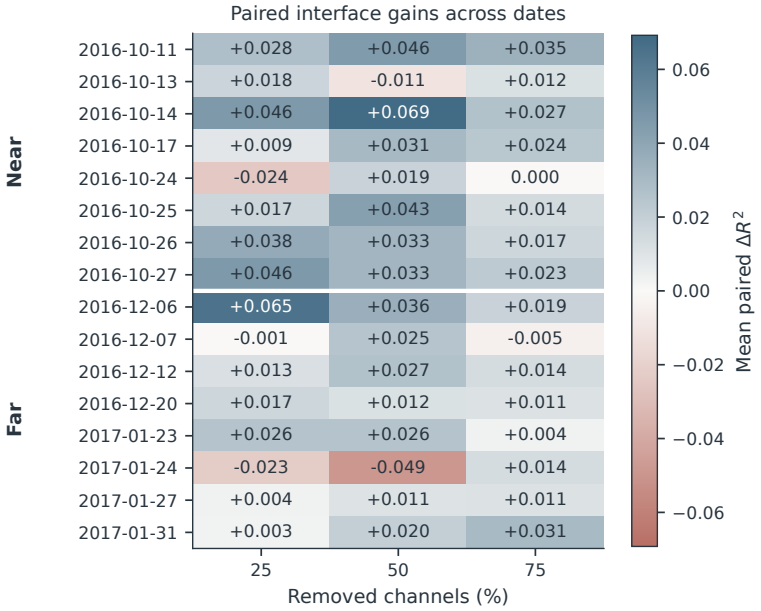

Figure S2: Date-wise persistent-channel-loss results on Indy. Each channel mask is held fixed throughout the support prefix and subsequent query stream. Full and StandardizationOnly use identical masks; model weights remain frozen and the session interface is rebuilt from the retained channels. The caption should additionally define every line, marker, removal rate, and error summary shown in the panel.

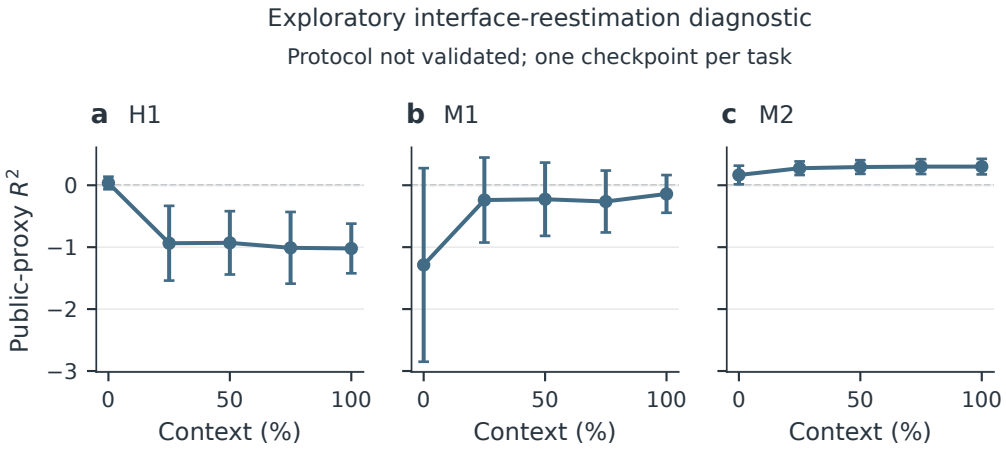

Figure S3: Interface-context diagnostic for one frozen $h = 16$ Indy checkpoint. Only the support used to estimate normalization and interface descriptors is varied among 4, 16, 64, and 108 seconds; every condition retains the full 108-second recurrent-state warm-up. Scores are date-macro R^2 values on the 16 Indy test dates.

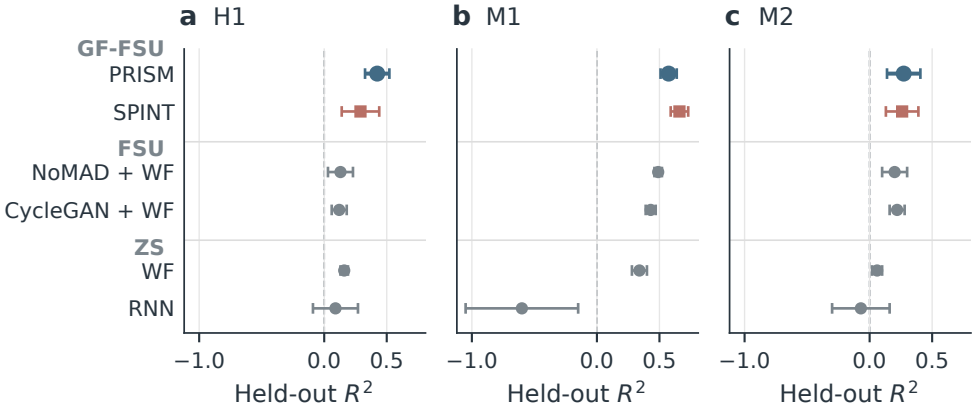

Figure S4: Complete point-by-point rendering of the official FALCON private held-out R^2 results summarized in Figure 3. PRISM scores are official evaluator returns; comparison scores are reported in SPINT (Le et al., 2025). GF-FSU denotes unlabeled target calibration without gradient updates; WF denotes a Wiener-filter decoder and RNN denotes a recurrent neural network. The caption should also define the task panels, individual points, summary markers, and error bars used in this rendering.