UPCSD: UNSUPERVISED PREFIX CALIBRATION WITH STATE–DYNAMIC MODELING FOR LONG-TERM INTRACORTICAL NEURAL DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-day neural drift degrades decoding performance in intracortical brain-computer interfaces (iBCIs), challenging robust long-term decoding under practical deployment constraints. We propose unsupervised prefix calibration with state–dynamic modeling (UPCSD), which uses the first 30 s of unlabeled test-day neural activity to correct prefix-estimable first-order cross-day variation and training-time augmentation to improve robustness to residual recording variation. A state–dynamic representation separately models slowly varying neural states and fast multiscale fluctuations, which are fused and decoded by a lightweight LSTM. Chronological offline evaluation on four iBCI datasets covering center-out (CO) and random target tracking (RTT) tasks showed that UPCSD achieved the numerically highest mean daily R^2 on each dataset. Mean daily R^2 ranged from 0.62 to 0.85 within 100 days and remained 0.57 and 0.46 over the full 700- and 757-day spans of Link-CO and Link-RTT, respectively. Under strictly causal inference using only past and current neural activity, UPCSD achieved the numerically highest mean daily R^2 on these four datasets and 0.672 ± 0.091 across 12 online test days on an additional NW-CO dataset. On the Lynxi HE200, on-chip inference closely matched GPU decoding performance while reducing active device power by approximately 92%. These results support UPCSD as a label-free approach for practical, long-term iBCI decoding.

1 INTRODUCTION

Reliable neural-to-behavior decoding over months or years is essential for the everyday autonomous use of intracortical brain-computer interfaces (iBCIs). Yet chronic recordings undergo cross-day neural drift, in which changes in recorded population statistics create a mismatch between historical training data and future-day activity, degrading fixed decoders (Chestek et al., 2011; Downey et al., 2018; Rokni et al., 2007). Supervised recalibration and closed-loop adaptation can restore decoding performance but require test-day behavioral labels, user participation or continuous feedback (Dangi et al., 2013; Orsborn et al., 2014). Unsupervised neural-manifold and distribution-alignment methods reduce this dependence, but often rely on assumptions about cross-day correspondence or distributional structure (Degenhart et al., 2020; Li et al., 2020). Deep temporal models can learn expressive neural representations, but sequence modeling alone does not address unlabeled cross-day mismatch and may increase deployment costs (Pandarinath et al., 2018; Ye & Pandarinath, 2021).

Cross-day mismatch reflects both recording-related and behavior-dependent variation (Jensen et al., 2022). We therefore focus on a prefix-estimable first-order component characterized by day-specific shifts in neural population center and channel-wise gain. As illustrated in Figure 1A, correcting these statistics substantially reduces day-dependent neural displacement while leaving residual cross-day variation. PFDC estimates this channel-wise affine correction from a short unlabeled prefix and applies it to subsequent activity. RDA complements PFDC by exposing the decoder during training to channel loss, residual gain and baseline changes, and observation noise that may remain after correction. Together, PFDC and RDA address prefix-estimable first-order mismatch and residual recording variation. A second-order covariance-alignment control under the same 30-s prefix constraint is provided in Appendix A.

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.

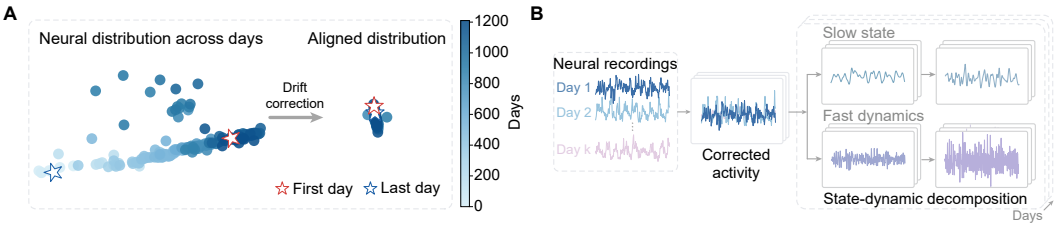

Figure 1: Prefix-based first-order drift correction and state–dynamic representation. (A) PCA projections of daily neural population means from Link-RTT before and after prefix-based first-order drift correction (PFDC). (B) First-order-corrected neural activity is decomposed into a slowly varying state flow and fast dynamic fluctuations estimated over local temporal windows.

Correcting cross-day mismatch does not determine how neural activity should be represented for decoding. Behavior-related neural activity exhibits structured population dynamics over time (Churchland et al., 2012; Oby et al., 2025), motivating representations that distinguish slowly varying context from faster local fluctuations. Existing representation-learning methods, including LFADS (Pandarinath et al., 2018), NDT2 (Ye et al., 2023), CEBRA (Schneider et al., 2023) and MARBLE (Gosztolai et al., 2025), model neural dynamics through latent-variable modeling, contextual sequence learning, contrastive embeddings or local flow-field representations. As illustrated in Figure 1B, we represent corrected activity as a slowly varying state flow together with fast dynamic fluctuations around that state. Separately encoding these components preserves slow neural context while capturing faster local changes for decoding.

Practical iBCI systems must operate within tight resource budgets, making lightweight models important for deployment. Compact architectures and hardware-aware optimization emphasize efficient deployment (Boi et al., 2016; Shaikh et al., 2019; An et al., 2022), whereas cross-day alignment and adaptation emphasize long-term robustness (Degenhart et al., 2020; Ma et al., 2023a; Karpowicz et al., 2025). This motivates a lightweight framework that combines label-free cross-day correction with structured temporal modeling and supports resource-constrained deployment.

Building on these observations, we introduce unsupervised prefix calibration with state–dynamic modeling (UPCSD), a lightweight framework for long-term intracortical neural decoding. UPCSD uses the initial 30 s of unlabeled neural activity from each recording day to correct channel-wise translation and gain, while training-time residual drift augmentation improves robustness to remaining recording variation. A mask-aware state–dynamic tokenizer separately encodes slow state flow and fast multiscale fluctuations, which are fused and mapped to behavior by an LSTM decoder. UPCSD requires neither test-day behavioral labels nor supervised recalibration. We further evaluate a strictly causal configuration and validate real-time online decoding on NW-CO, together with on-chip inference on resource-constrained hardware. The main contributions of this work are as follows:

- We propose a label-free cross-day calibration strategy using only the first 30 s of neural activity from each test day. PFDC corrects prefix-estimable channel-wise translation and gain, while training-time RDA improves robustness to residual recording variation. We validate this strategy chronologically on four long-term intracortical datasets spanning up to 757 days.
- We develop a mask-aware state–dynamic representation that separately models slowly varying neural states and fast multiscale fluctuations. Ablations quantify the complementary contributions of the state and dynamic branches to decoding.
- We demonstrate causal decoding, real-time online operation, and hardware deployment of UPCSD. A strictly causal configuration using only past and current neural activity is evaluated on four longitudinal datasets and further validated through real-time decoding on 12 NW-CO test days. UPCSD is also deployed on the Lynxi HE200, closely matching GPU decoding performance with substantially lower active-inference device power.

2 Related Work

Cross-Day Calibration and Neural Alignment Cross-day changes in recorded neural activity progressively degrade fixed decoders despite stable behavioral intent. Previous studies have shown that neural population structure or dynamics can remain stable despite changes in individual-neuron activity (Gallego et al., 2020; Natraj et al., 2025; Driscoll et al., 2017; Qin et al., 2023). Existing approaches address this challenge through supervised recalibration, closed-loop adaptation, self-calibrating classifiers, manifold alignment, domain adaptation and meta-learning across recordings (Bishop et al., 2014; Dangi et al., 2013; Orsborn et al., 2014; Degenhart et al., 2020; Karpowicz et al., 2025; Li et al., 2020; Ma et al., 2023a; Vermani et al., 2025). UPCSD instead performs label-free test-day calibration using only a short unlabeled neural prefix.

Temporal Representation Learning for Neural Decoding Classical neural decoders model temporal dependencies using state-space formulations such as Kalman filters (Wu et al., 2006). More recent methods learn neural representations through latent dynamical modeling (LFADS), Transformer-based contextual modeling (NDT and NDT2), contrastive learning (CEBRA), and geometric representation learning (MARBLE) (Pandarinath et al., 2018; Ye & Pandarinath, 2021; Ye et al., 2023; Schneider et al., 2023; Gosztolai et al., 2025). These approaches have substantially advanced temporal representation learning for neural decoding. UPCSD extends this direction by coupling explicit cross-day calibration with structured state–dynamic representation learning in a lightweight framework.

Lightweight Neural Decoding and Hardware Deployment Prior work has demonstrated low-power intracortical neural decoding using neuromorphic processors and application-specific hardware (Boi et al., 2016; Shaikh et al., 2019; An et al., 2022). These studies primarily emphasize efficient feature extraction, decoder implementation or hardware optimization rather than long-term cross-day robustness. Our work focuses on long-term label-free neural decoding and further evaluates UPCSD through on-chip inference on the resource-constrained Lynxi HE200 edge-computing chip.

3 Methods

Given neural activity X and behavioral targets Y from training days, our goal is to learn a neural-to-behavior mapping that generalizes to test days without behavioral labels. As illustrated in Figure 2, UPCSD comprises four stages. First, PFDC estimates channel-wise translation and gain from a short unlabeled prefix for first-order drift correction. Second, training-only RDA simulates residual recording variation. Third, mask-aware state–dynamic tokenization separately encodes slow state flow and fast dynamic fluctuations. Finally, a lightweight recurrent decoder maps the fused representation to behavior. UPCSD supports both noncausal and causal temporal configurations within the same framework.

3.1 Problem Formulation

On recording day n, the neural sequence is $X_n = \{x_{n,t}\}_{t=1}^{T}$, with $x_{n,t} \in \mathbb{R}^C$, and the corresponding behavioral sequence is $Y_n = \{y_{n,t}\}_{t=1}^{T}$, with $y_{n,t} \in \mathbb{R}^D$, where C and D denote the number of neural channels and behavioral dimensions, respectively. UPCSD is trained on historical days with paired neural and behavioral sequences and evaluated on subsequent days for which only neural activity is available. For each test day, UPCSD uses only a short unlabeled prefix

$$\mathcal{P}_n = \{x_{n,t}\}_{t=1}^{T_p}, \tag{1}$$

where T_p corresponds to approximately 30 s of neural activity. The prefix is then used to calibrate the full neural sequence X_n before predicting the corresponding behavioral sequence Y_n without behavioral labels or supervised recalibration.

We use the following decomposition to distinguish the drift components that can be estimated from a short prefix from those that cannot:

$$x_{n,t} = \boldsymbol{\mu}_n + \mathrm{Diag}(\boldsymbol{\sigma}_n)(s_{n,t} + d_{n,t}) + r_{n,t} + \epsilon_{n,t}. \tag{2}$$

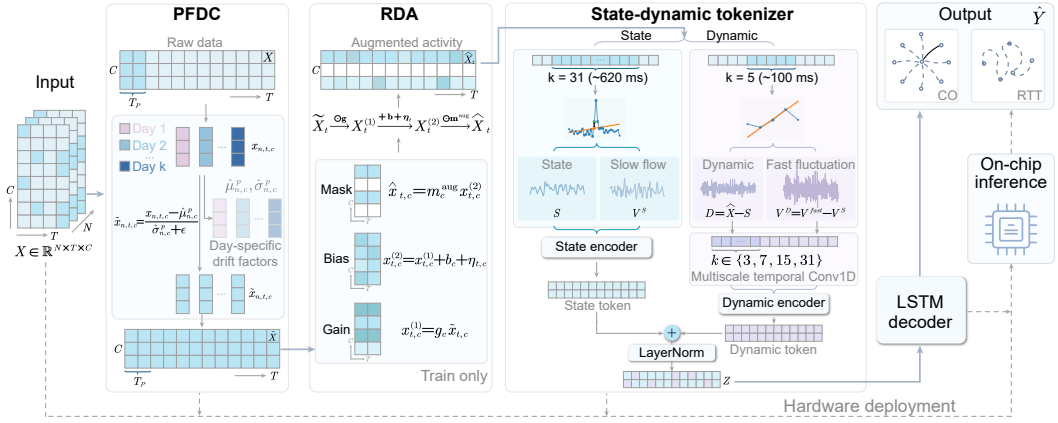

Figure 2: Overview of UPCSD. Prefix-based first-order drift correction (PFDC) corrects channel-wise translation and gain using a 30-s unlabeled prefix, while training-only residual drift augmentation (RDA) simulates residual recording variation. A mask-aware state–dynamic tokenizer encodes slow state flow and multiscale fast dynamic fluctuations. The fused tokens are decoded by an LSTM decoder, with optional on-chip inference.

Here, $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$ denote the day-specific channel-wise translation and gain. The variables $s_{n,t}$ and $d_{n,t}$ represent the slow state flow and fast dynamic fluctuations, respectively. The residual term $r_{n,t}$ captures channel loss and residual gain or baseline changes that cannot be reliably identified from the short prefix, while $\epsilon_{n,t}$ denotes observation noise. UPCSD estimates only $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$ at test time and does not assume that prefix correction removes all cross-day variation.

3.2 Prefix-based First-Order Drift Correction

Under the decomposition in Eq. (2), PFDC models the prefix-estimable first-order drift as a day-specific diagonal affine transformation. For each recording day n, the translation and gain parameters are estimated exclusively from the prefix:

$$\hat{\boldsymbol{\mu}}_n = \frac{1}{T_p}\sum_{t=1}^{T_p} x_{n,t}, \qquad \hat{\boldsymbol{\sigma}}_n = \mathrm{Std}_{t\le T_p}(x_{n,t}). \tag{3}$$

For numerical stability, the effective scale is $\bar{\sigma}_{n,c} = 1$ if $\hat{\sigma}_{n,c} < \tau$, and $\bar{\sigma}_{n,c} = \hat{\sigma}_{n,c}$ otherwise, with $\tau = 10^{-6}$. The same affine correction is applied to every time bin recorded on that day,

$$\tilde{x}_{n,t} = \mathrm{Diag}(\bar{\boldsymbol{\sigma}}_n)^{-1}(x_{n,t} - \hat{\boldsymbol{\mu}}_n), \tag{4}$$

The corrected activity $\tilde{X}_n$ is used as a first-order normalized representation rather than treated as drift-free. The same prefix-based correction is applied to both training and test days, using no behavioral labels or post-prefix statistics. We restrict test-time calibration to channel-wise translation and gain, since higher-order covariance structure estimated from a short unlabeled prefix may also reflect behavioral variability and observation noise. Full covariance alignment is evaluated separately in Appendix A.

3.3 Residual Drift Augmentation

PFDC corrects only the prefix-estimable channel-wise translation and gain. It does not explicitly account for channel loss, residual gain or baseline changes, or observation noise. We therefore apply residual drift augmentation (RDA) during training to simulate these remaining variations:

$$\mathcal{A}(\tilde{x}_t) = m^{\mathrm{aug}} \odot (g \odot \tilde{x}_t + b + \eta_t)\,, \tag{5}$$

where the channel-wise perturbations are sampled as

$$\begin{cases} m_c^{\mathrm{aug}} \sim \mathrm{Bernoulli}(1-p), & g_c \sim \mathrm{LogNormal}(0, \sigma_g^2), \\ b_c \sim \mathcal{N}(0, \sigma_b^2), & \eta_{t,c} \sim \mathcal{N}(0, \sigma_\eta^2). \end{cases} \tag{6}$$

The mask m^{aug} simulates unstable or dropped channels, whereas g, b, and η_t model residual gain changes, baseline shifts, and observation noise, respectively. Masked channels are set to zero after the gain, bias, and noise perturbations are applied. Accordingly, the tokenizer receives

$$\hat{X} = \begin{cases} \mathcal{A}(\tilde{X}), & \text{during training,} \\ \tilde{X}, & \text{during testing.} \end{cases} \tag{7}$$

The augmentation mask m^{aug} is distinct from the validity mask introduced below to exclude padded time bins.

3.4 State–Dynamic Tokenization

Mask-aware local fitting UPCSD first characterizes the local level and temporal trend of $\hat{X}$ using a shared linear fitting operator. For a window of length k, let $\mathcal{I}_k$ denote the temporal offsets relative to time t. The noncausal configuration uses a centered window, whereas the causal configuration uses a window ending at the current time:

$$\mathcal{I}_k = \begin{cases} \{-h, \dots, h\}, \quad k = 2h+1, & \text{noncausal,} \\ \{-(k-1), \dots, 0\}, & \text{causal.} \end{cases} \tag{8}$$

The local intercept $S_t^{(k)}$ and slope $V_t^{(k)}$ are obtained by

$$(S_t^{(k)}, V_t^{(k)}) = \arg\min_{a,b} \sum_{\ell \in \mathcal{I}_k} v_{t+\ell} w_\ell \left\| \hat{x}_{t+\ell} - a - \ell b \right\|_2^2, \tag{9}$$

where w_ℓ is a temporal window weight (Hann for the four longitudinal datasets and Hamming for NW-CO state fitting) (Harris, 1978), and $v_{t+\ell} \in \{0, 1\}$ indicates whether the corresponding time bin is valid. The validity mask excludes padded bins from the local fit. The fitted intercept $S_t^{(k)}$ and slope $V_t^{(k)}$ summarize the local level and temporal trend within the window, respectively. We denote this operator by

$$F_k(\hat{X})_t = (S_t^{(k)}, V_t^{(k)}).$$

Slow state flow For the four longitudinal datasets, the state branch applies the local fitting operator with a 31-bin window, corresponding to approximately 620 ms at a bin width of 20 ms:

$$(S_t, V_t^S) = F_{31}(\hat{X})_t, \qquad Q_t^S = [S_t, \alpha_S V_t^S]. \tag{10}$$

Using a relatively long window suppresses rapid fluctuations, allowing S_t to capture the slowly varying neural state and V_t^S to characterize its gradual evolution. Their concatenation forms the state representation Q_t^S, where α_S scales the slope component. NW-CO uses F_{26} with the preceding 25 bins and the current bin.

Fast dynamic fluctuations The dynamic branch captures rapid neural fluctuations relative to the slow state. Specifically, the amplitude fluctuation and slope fluctuation are defined as

$$D_t^{\mathrm{amp}} = \hat{x}_t - S_t, \qquad V_t^D = V_t^{\mathrm{fast}} - V_t^S, \tag{11}$$

where $(\cdot, V_t^{\mathrm{fast}}) = F_5(\hat{X})_t$. The dynamic representation is then constructed as

$$Q_t^D = [D_t^{\mathrm{amp}}, \alpha_D V_t^D]. \tag{12}$$

To capture fast fluctuations across multiple temporal scales, the dynamic branch applies depthwise temporal convolutions to the sequence $Q^D = \{Q_t^D\}_{t=1}^T$, with $\mathcal{K} = \{3, 7, 15, 31\}$ corresponding to approximately 60, 140, 300, and 620 ms. The representation at time t is

$$H_t^D = \mathrm{Concat}_{k \in \mathcal{K}} \left[\mathrm{DWConv}_k(Q^D) \right]_t. \tag{13}$$

The temporal convolutions use centered padding in the noncausal configuration and left-only padding in the causal configuration.

Token projection and fusion The state and dynamic representations are encoded separately by a state encoder $f_S(\cdot)$ and a dynamic encoder $f_D(\cdot)$:

$$h_t^S = f_S(Q_t^S), \qquad h_t^D = f_D(H_t^D). \tag{14}$$

Their outputs are fused by elementwise summation followed by layer normalization:

$$z_t = \mathrm{LN}\big(h_t^S + h_t^D\big). \tag{15}$$

The fused representation combines slowly varying neural state information with fast dynamic fluctuations and serves as the input to the sequence decoder.

3.5 Lightweight Sequence Decoder

The fused token sequence $Z = \{z_t\}_{t=1}^T$ is mapped to behavior by an LSTM–MLP decoder, $\hat{Y}_{1:T} = g_{\mathrm{dec}}(Z_{1:T})$. The noncausal and causal configurations use two-layer bidirectional and one-layer unidirectional LSTMs, respectively, with 128 hidden units.

3.6 Training Loss

The model is trained on labeled historical days using masked mean-squared error over valid time bins, $\mathcal{L} = (\sum_t v_t D)^{-1} \sum_t v_t \|\hat{y}_t - y_t\|_2^2$, where $v_t \in \{0, 1\}$ indicates valid time bins. Behavioral targets are normalized using training-label statistics and restored to their original scale before evaluation. All trainable components are optimized jointly using AdamW.

4 Experiments

4.1 Experimental Setup

Datasets We evaluated UPCSD on four longitudinal intracortical datasets and one online validation dataset covering center-out (CO) and random target tracking (RTT) tasks. **Link-CO** and **Link-RTT**, derived from the Link dataset (Temmar et al., 2025), use 96-channel spiking-band-power features to decode two-dimensional finger position. **Indy-RTT** uses M1 recordings from subject Indy (O'Doherty et al., 2020) to decode two-dimensional movement velocity, while **Jango-CO** uses 96-channel M1 multiunit activity from an eight-direction isometric wrist task (Ma et al., 2023a;b) to decode cursor position. **NW-CO** is an in-house 1024-channel center-out dataset used for online cursor decoding. Across datasets, the elapsed calendar time between the first and last test recordings ranged from 31 to 757 days.

Chronological Evaluation Protocol For the four longitudinal datasets, recording days were ordered chronologically. The earliest 20% formed the development pool, with 10% of this pool reserved for validation, while the remaining 80% of recording days were used for testing. NW-CO used four training recording days followed by 12 online test recording days spanning 31 calendar days. Exact split boundaries are provided in Appendix B. For each test recording, PFDC estimated channel-wise translation and gain from the first 30 s of unlabeled neural activity and fixed the resulting correction for all subsequent samples. The prefix was excluded from evaluation. For UPCSD, calibration used no test-day behavioral labels or post-prefix neural statistics, and model parameters remained fixed during testing. Post-prefix R^2 was computed per test recording day and summarized as mean $\pm$ sample standard deviation.

Implementation Details. Behavioral targets were normalized using statistics computed from the training labels only and restored to their original scale before evaluation. The four longitudinal datasets were evaluated offline, including a strictly causal zero-look-ahead setting, whereas NW-CO was evaluated online during task execution using zero-look-ahead neural inputs. Exact evaluation settings and hyperparameters are provided in Appendix B.

4.2 Results

Cross-Day Mean Drift and State–Dynamic Structure We first quantified cross-day variation in the neural population mean on Link-RTT and assessed how much could be corrected using a short

unlabeled prefix. For each recording day, PFDC parameters were estimated from the initial prefix and applied to the remaining post-prefix activity. Before correction, the day-specific population means varied substantially across recording days, whereas PFDC sharply reduced this variation (Figure 3A). Increasing the prefix duration from 1 to 30 s rapidly improved drift reduction, reaching approximately 90% at 30 s, while extending the prefix to 120 s yielded only modest additional improvement. We therefore used a 30 s prefix in subsequent experiments as a practical trade-off between estimation quality and calibration duration.

Figure 3B visualizes the corrected Link-RTT activity and its state and dynamic representations as local vector fields. The state representation formed smooth local trajectories describing slowly varying activity levels and trends, whereas the dynamic representation isolated faster amplitude and slope fluctuations around the local state. These qualitative differences motivate separate encoding of the two representations, whose contributions to decoding performance are quantified in the subsequent ablation study.

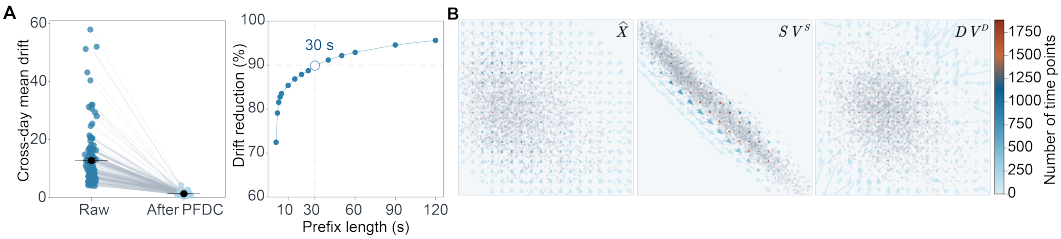

Figure 3: PFDC and state–dynamic factorization. (A) Cross-day mean drift before and after PFDC and drift reduction versus prefix length. (B) Vector fields of corrected activity $\hat{X}$, slow state flow (S, V^S), and fast dynamic flow (D, V^D).

Table 1: Main cross-day decoding performance. Entries are mean ± standard deviation of daily R^2. Methods marked with "+ PFDC" use first-order correction estimated from the first 30 s of unlabeled test-day neural activity. UPCSD includes PFDC.

Method	Link-CO (700 days)	Link-RTT (757 days)	Indy-RTT (123 days)	Jango-CO (88 days)
CEBRA (Schneider et al., 2023)	-0.061 ± 0.070	-0.028 ± 0.042	0.283 ± 0.191	0.511 ± 0.309
CEBRA + PFDC	0.387 ± 0.179	0.395 ± 0.134	0.338 ± 0.263	0.770 ± 0.078
Transformer (Vaswani et al., 2017)	0.323 ± 0.197	0.185 ± 0.208	0.567 ± 0.299	0.602 ± 0.254
Transformer + PFDC	0.520 ± 0.193	0.416 ± 0.156	0.551 ± 0.242	0.787 ± 0.066
BiLSTM (Graves, 2012)	0.210 ± 0.103	0.007 ± 0.042	0.566 ± 0.269	0.668 ± 0.197
BiLSTM + PFDC	0.487 ± 0.231	0.455 ± 0.161	0.523 ± 0.259	0.827 ± 0.061
UPCSD (Ours)	$\mathbf{0.568 \pm 0.179}$	$\mathbf{0.462 \pm 0.178}$	$\mathbf{0.588 \pm 0.246}$	$\mathbf{0.849 \pm 0.044}$

Main Cross-Day Decoding Performance Table 1 compares UPCSD with three neural decoders evaluated with and without PFDC in the offline full-sequence setting, while strictly causal decoding is evaluated separately in Table 2. Detailed reproduction settings and fairness controls are provided in Appendix C. PFDC had the strongest effect on the two long-term Link benchmarks: across CEBRA, Transformer, and BiLSTM, it improved mean daily R^2 by 0.197–0.448 on Link-CO and 0.231–0.448 on Link-RTT. PFDC also improved all three baselines on Jango-CO, with gains of 0.159–0.259, indicating that a substantial component of cross-day mismatch can be reduced using only the first 30 s of unlabeled test-day activity.

On Indy-RTT, the effect of PFDC was more model dependent: it improved CEBRA, whereas the uncorrected Transformer and BiLSTM already achieved relatively strong performance and showed no additional benefit from PFDC. Despite these dataset-dependent differences, UPCSD achieved the highest mean daily R^2 on all four benchmarks. Its margins over the strongest PFDC-enabled baseline were largest on Link-CO and Indy-RTT, while the gains were smaller on Link-RTT and Jango-CO, where the strongest calibrated baselines already approached UPCSD performance.

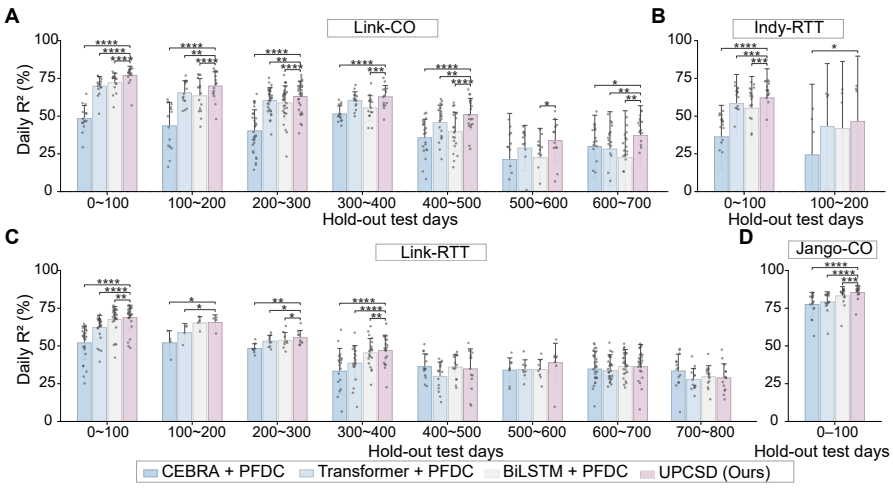

Figure 4: Daily R^2 across held-out intervals for Link-CO (A), Indy-RTT (B), Link-RTT (C), and Jango-CO (D). Points are test days, bars summarize intervals, and brackets show significance; all decoders use PFDC.

Figure 4 further resolves the PFDC-enabled results by elapsed calendar time. On Link-CO and Link-RTT, decoding performance generally decreased and day-to-day variability increased as the test interval moved farther from the training period, highlighting the increasing difficulty of long-horizon generalization. Within the first 100 calendar days, UPCSD achieved mean daily R^2 values of approximately 0.76 and 0.68 on Link-CO and Link-RTT, respectively. Across the full test spans, these values decreased to 0.568 over 700 days and 0.462 over 757 days, respectively, while UPCSD remained competitive with the strongest baselines across subsequent intervals. Detailed interval-wise statistical comparisons are reported in Appendix D.

Table 2: Causal decoding performance. The four longitudinal datasets use offline zero-look-ahead inference; Causal UPCSD on NW-CO was evaluated online over 12 test days. "+ PFDC" denotes 30-s unlabeled prefix correction. Entries are mean ± SD of daily R^2.

Method	NW-CO	Link-CO	Link-RTT	Indy-RTT	Jango-CO
NoMAD	-0.023 ± 0.020	0.012 ± 0.020	0.028 ± 0.198	0.102 ± 0.606	0.756 ± 0.121
NoMAD + PFDC	0.276 ± 0.233	0.223 ± 0.297	0.294 ± 0.159	0.253 ± 0.247	0.846 ± 0.057
LSTM	0.179 ± 0.197	-0.015 ± 0.041	-0.829 ± 0.638	0.339 ± 0.276	0.590 ± 0.168
LSTM + PFDC	0.295 ± 0.226	0.353 ± 0.174	0.258 ± 0.215	0.418 ± 0.244	0.829 ± 0.081
Causal UPCSD	$\mathbf{0.672 \pm 0.091}$	$\mathbf{0.397 \pm 0.221}$	$\mathbf{0.331 \pm 0.177}$	$\mathbf{0.589 \pm 0.247}$	$\mathbf{0.850 \pm 0.059}$

Causal and Online Evaluation Table 2 evaluates causal UPCSD using only past and current neural samples. Link-CO, Link-RTT, Indy-RTT, and Jango-CO were evaluated offline, whereas Causal UPCSD was deployed online on NW-CO for 12 test days spanning 31 calendar days while the subject performed the center-out task; the NW-CO baselines were evaluated offline on the corresponding recordings. Causal UPCSD achieved mean daily R^2 values of 0.397, 0.331, 0.589, and 0.850 on the four longitudinal datasets, respectively, and 0.672 ± 0.091 across the 12 online NW-CO test days. It achieved the numerically highest mean performance among the compared methods on all five datasets, demonstrating that UPCSD remains effective under strictly causal inference and can support real-time online decoding without future neural samples.

Ablation Study Table 3 examines component ablations and interaction controls. Removing PFDC reduced mean daily R^2 by 0.119–0.313, while removing RDA reduced performance by 0.047–0.121, indicating that PFDC provides the primary contribution to cross-day robustness and RDA offers complementary robustness to residual recording variation. Removing the state–dynamic decomposition reduced performance across all four datasets by 0.016–0.051. State-only consistently

outperformed dynamic-only, while adding the dynamic branch improved performance on all four datasets. Jointly removing PFDC and RDA caused the largest degradation on the two Link benchmarks. On Indy-RTT, however, removing either the state or dynamic branch in the absence of PFDC improved performance relative to removing PFDC alone, suggesting that the benefit of the full state–dynamic representation partly depends on first-order calibration. Overall, RDA and state–dynamic modeling provide additional, partly dataset-dependent gains. A second-order covariance control provided no additional benefit (Appendix A).

Table 3: Ablation and interaction-control results. Entries are mean ± standard deviation of daily R^2 across test days.

Configuration	Link-CO	Link-RTT	Indy-RTT	Jango-CO
w/o PFDC	0.255 ± 0.296	0.177 ± 0.315	0.286 ± 0.321	0.730 ± 0.177
w/o RDA	0.447 ± 0.207	0.386 ± 0.195	0.482 ± 0.212	0.802 ± 0.063
w/o Dynamic	0.513 ± 0.204	0.446 ± 0.161	0.522 ± 0.275	0.845 ± 0.050
w/o State	0.272 ± 0.139	0.145 ± 0.156	0.418 ± 0.237	0.798 ± 0.050
w/o State & Dynamic	0.517 ± 0.192	0.446 ± 0.171	0.543 ± 0.279	0.819 ± 0.055
w/o Dynamic & PFDC	0.249 ± 0.302	0.193 ± 0.328	0.570 ± 0.272	0.739 ± 0.175
w/o State & PFDC	0.174 ± 0.221	0.139 ± 0.205	0.539 ± 0.213	0.801 ± 0.064
w/o PFDC & RDA	-0.187 ± 0.582	-0.620 ± 0.604	0.479 ± 0.222	0.693 ± 0.170
UPCSD (Ours)	$\mathbf{0.568 \pm 0.179}$	$\mathbf{0.462 \pm 0.178}$	$\mathbf{0.588 \pm 0.246}$	$\mathbf{0.849 \pm 0.044}$

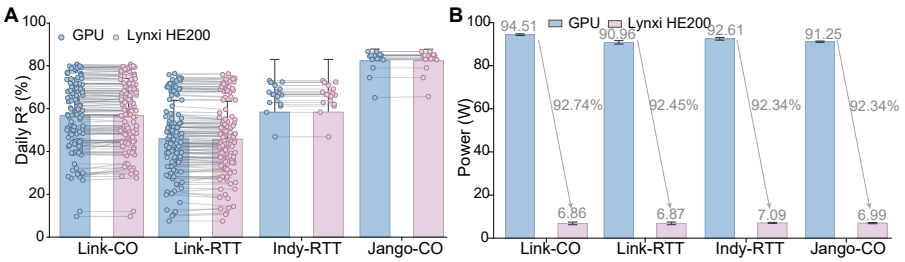

Figure 5: GPU–HE200 comparison. (A) Daily R^2. (B) Mean decoder-device power during active inference; percentages denote reductions relative to GPU.

On-Chip Deployment Figure 5 compares device-level UPCSD inference on a GPU and the Lynxi HE200 edge-computing chip. Across all four benchmarks, on-chip inference produced negligible differences in daily R^2 relative to GPU inference, with paired results closely overlapping across test days and establishing numerical agreement between the two platforms. The HE200 also substantially reduced mean decoder-device power during active inference, averaging 6.86–7.09 W across datasets compared with 90.96–94.51 W for GPU inference, corresponding to reductions of 92.34%–92.74%. Thus, the tested HE200 configuration closely matched GPU decoding performance while operating at a small fraction of the measured GPU power budget. Detailed deployment configurations, matched accuracy results, latency measurements, and power-measurement settings are provided in Appendix E.

5 Conclusion

We introduced UPCSD, a label-free framework for long-term intracortical neural decoding using 30-s prefix calibration, residual drift augmentation, and state–dynamic modeling. UPCSD achieved the numerically highest mean daily R^2 across four longitudinal datasets spanning up to 757 days. Under strictly causal inference, it retained the numerically highest mean daily R^2 on these datasets and achieved $R^2 = 0.672 \pm 0.091$ across 12 online test days on NW-CO. On the Lynxi HE200, UPCSD closely matched GPU decoding performance with approximately 92% lower active device power, supporting practical long-term iBCI decoding.

AI USE STATEMENT

Generative AI tools were employed to improve the language and organization of the manuscript. All technical content, derivations, experimental results, and citations were independently checked by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

Detailed experimental configurations, including implementation and training settings, are provided in the Appendix. The supplementary material includes the UPCSD implementations for offline evaluation and causal inference, together with the NW online deployment code.

REFERENCES

Hyochan An, Samuel R Nason-Tomaszewski, Jongyup Lim, Kyumin Kwon, Matthew S Willsey, Parag G Patil, Hun-Seok Kim, Dennis Sylvester, Cynthia A Chestek, and David Blaauw. A power-efficient brain-machine interface system with a sub-mw feature extraction and decoding asic demonstrated in nonhuman primates. *IEEE transactions on biomedical circuits and systems*, 16(3):395–408, 2022.

William Bishop, Cynthia C 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.

Fabio Boi, Timoleon Moraitis, Vito De Feo, Francesco Diotalevi, Chiara Bartolozzi, Giacomo Indiveri, and Alessandro Vato. A bidirectional brain-machine interface featuring a neuromorphic hardware decoder. *Frontiers in neuroscience*, 10:563, 2016.

Cynthia A Chestek, Vikash Gilja, Paul Nuyujukian, Justin D Foster, Joline M Fan, Matthew T Kaufman, Mark M Churchland, Zuley Rivera-Alvidrez, John P Cunningham, Stephen I Ryu, et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of neural engineering*, 8(4):045005, 2011.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487 (7405):51–56, 2012.

Siddharth Dangi, Amy L Orsborn, Helene G Moorman, and Jose M Carmena. Design and analysis of closed-loop decoder adaptation algorithms for brain-machine interfaces. *Neural computation*, 25(7):1693–1731, 2013.

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.

John E Downey, Nathaniel Schwed, Steven M Chase, Andrew B Schwartz, and Jennifer L Collinger. Intracortical recording stability in human brain–computer interface users. *Journal of neural engineering*, 15(4):046016, 2018.

Laura N Driscoll, Noah L Pettit, Matthias Minderer, Selmaan N Chettih, and Christopher D Harvey. Dynamic reorganization of neuronal activity patterns in parietal cortex. *Cell*, 170(5):986–999, 2017.

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.

Adam Gosztolai, Robert L Peach, Alexis Arnaudon, Mauricio Barahona, and Pierre Vandergheynst. Marble: interpretable representations of neural population dynamics using geometric deep learning. *Nature Methods*, 22(3):612–620, 2025.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Fredric J Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

Kristopher T Jensen, Naama Kadmon Harpaz, Ashesh K Dhawale, Steffen BE Wolff, and Bence P Ölveczky. Long-term stability of single neuron activity in the motor system. *Nature neuroscience*, 25(12):1664–1674, 2022.

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, et al. Stabilizing brain-computer interfaces through alignment of latent dynamics. *Nature Communications*, 16(1):4662, 2025.

Wei Li, Shaohua Ji, Xi Chen, Bo Kuai, Jiping He, Peng Zhang, and Qiang Li. Multi-source domain adaptation for decoder calibration of intracortical brain-machine interface. *Journal of neural engineering*, 17(6):066009, 2020.

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, 2023a.

Xuan Ma, Fabio Rizzoglio, Stephanie N. Thacker, Christian Ethier, Matthew G. Perich, and Lee E. Miller. Data from: Using adversarial networks to extend brain computer interface decoding accuracy over time. Dryad Digital Repository, 2023b. URL `https://doi.org/10.5061/dryad.cvdncjt7n`. Dataset.

Nikhilesh Natraj, Sarah Seko, Reza Abiri, Runfeng Miao, Hongyi Yan, Yasmin Graham, Adelyn Tu-Chan, Edward F Chang, and Karunesh Ganguly. Sampling representational plasticity of simple imagined movements across days enables long-term neuroprosthetic control. *Cell*, 188(5):1208–1225, 2025.

Emily R. Oby, Alan D. Degenhart, Erinn M. Grigsby, Asma Motiwala, Nicole T. McClain, Patrick J. Marino, Byron M. Yu, and Aaron P. Batista. Dynamical constraints on neural population activity. *Nature Neuroscience*, 28:383–393, 2025. doi: 10.1038/s41593-024-01845-7.

Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology. Zenodo, 2020. URL `https://doi.org/10.5281/zenodo.3854034`. Dataset, version 2.

Amy L Orsborn, Helene G Moorman, Simon A Overduin, Maryam M Shanechi, Dragan F Dimitrov, and Jose M Carmena. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron*, 82(6):1380–1393, 2014.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Shanshan Qin, Shiva Farashahi, David Lipshutz, Anirvan M Sengupta, Dmitri B Chklovskii, and Cengiz Pehlevan. Coordinated drift of receptive fields in hebbian/anti-hebbian network models during noisy representation learning. *Nature Neuroscience*, 26(2):339–349, 2023.

Uri Rokni, Andrew G Richardson, Emilio Bizzi, and H Sebastian Seung. Motor learning with unstable neural representations. *Neuron*, 54(4):653–666, 2007.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

Shoeb Shaikh, Rosa So, Tafadzwa Sibindi, Camilo Libedinsky, and Arindam Basu. Towards intelligent intracortical bmi (i^2bmi): Low-power neuromorphic decoders that outperform kalman filters. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1615–1624, 2019.

Hisham Temmar, Yixuan Wang, Nina Gill, Nicholas B. Mellon, Chang Liu, Luis Hernan Cubillos, Rio I. Parsons, Joseph T. Costello, Matteo Ceradini, Madison M. Kelberman, Matthew J. Mender, Aren I. Hite, Dylan M. Wallace, Samuel R. Nason-Tomaszewski, Matthew S. Willsey, Parag G. Patil, Anne W. Draelos, and Cynthia A. Chestek. Long-term intracortical neural activity and kinematics (LINK): An intracortical neural dataset for chronic brain–machine interfaces, neuroscience, and machine learning. In *Proceedings of the 39th Conference on Neural Information Processing Systems: Datasets and Benchmarks Track*, 2025. URL https://dandiarchive.org/dandiset/001201. DANDI Archive, Dandiset 001201.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ayesha Vermani, Josue Nassar, Hyungju Jeon, Matthew Dowling, and Il Memming Park. Meta-dynamical state space models for integrative neural data analysis. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SRpq5OBpED.

Wei Wu, Yun Gao, Elie Bienenstock, John P Donoghue, and Michael J Black. Bayesian population decoding of motor cortical activity using a kalman filter. *Neural computation*, 18(1):80–118, 2006.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

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.

A Second-Order Covariance Alignment under an Unlabeled Prefix

One might attempt to extend PFDC's diagonal first-order test-day calibration with a full second-order covariance transform estimated from the same unlabeled prefix. Writing $A_n = \mathrm{Diag}(\boldsymbol{\sigma}_n)$, $z_n = s_n + d_n$, and $u_n = r_n + \epsilon_n$, the decomposition in the main text gives

$$\begin{aligned}\mathrm{Cov}(x_n) = A_n \,\mathrm{Cov}(z_n) A_n^\top + \mathrm{Cov}(u_n) \\ + A_n \,\mathrm{Cov}(z_n, u_n) + \mathrm{Cov}(u_n, z_n) A_n^\top .\end{aligned} \tag{16}$$

For continuous behavior, $\mathrm{Cov}(s_n + d_n)$ varies with behavioral occupancy, trajectory distribution, and movement speed. The observed covariance therefore mixes recording drift, behavior distribution, and noise. These components are not uniquely identifiable from a short unlabeled prefix. UPCSD consequently restricts test-day calibration to channel-wise translation and gain, while full covariance alignment is evaluated only as a second-order control.

Table 4 compares Full UPCSD with a second-order CORAL transform estimated from the same 30-s unlabeled prefix. Across all four datasets, the CORAL control reduced mean daily joint R^2 relative to Full UPCSD. The reductions were 0.101, 0.095, 0.056, and 0.047 for Link-CO, Link-RTT, Indy-RTT, and Jango-CO, respectively.

Table 4: Daily R^2 after extending PFDC with second-order covariance alignment using 30-s CORAL. Full UPCSD is included as the reference. Entries are mean $\pm$ sample standard deviation across test days, and Δ is CORAL minus Full UPCSD.

Dataset	Full UPCSD joint R^2	30-s CORAL joint R^2	Δ
Link-CO	$\mathbf{0.568 \pm 0.179}$	0.467 ± 0.202	-0.101
Link-RTT	$\mathbf{0.462 \pm 0.178}$	0.367 ± 0.191	-0.095
Indy-RTT	$\mathbf{0.588 \pm 0.246}$	0.532 ± 0.232	-0.056
Jango-CO	$\mathbf{0.849 \pm 0.044}$	0.802 ± 0.052	-0.047

B Dataset and Implementation Details

Table 5 records the exact chronological split boundaries for the four longitudinal datasets. Recording days were never randomly mixed across training, validation, and test sets. Dataset-specific sequence construction and evaluation settings are summarized below the table.

Table 5: Chronological training, validation, and test splits for the four longitudinal datasets. Date ranges include recording days only; counts give the corresponding numbers of recording days.

Dataset	Training	Validation	Testing
Indy-RTT	20160915 20160916 20160921 20160926	20160927	20160930∼20170131 21 recording days
Jango-CO	20150730 20150731 20150801	20150805	20150806∼20151102 16 recording days
Link-CO	20200127∼20201211 28 recording days	20210104 20210105 20210106	20210108∼20221209 124 recording days
Link-RTT	20200228∼20210428 28 recording days	20210505 20210519 20210525	20210526∼20230622 125 recording days

Sequence construction and evaluation details. The four longitudinal datasets contain 96-channel neural inputs at a temporal resolution of 20 ms. Indy-RTT days were divided into overlapping 60-bin

(1.20-s) windows with a stride of 20 bins (0.40 s), producing 33,849 windows in total. Jango-CO trials contain 30 bins (0.60 s) and were used without truncation. Link-CO and Link-RTT contain variable-length trials. The implementation enforces a maximum length of 256 bins (5.12 s), but no trial in the evaluated Link datasets exceeded this limit, so this cap removed no bins. The noncausal configuration used a two-bin neural-to-behavior lag, corresponding to 40 ms, for evaluation, so a sequence of length L produced at most $L - 2$ valid prediction–target pairs. Under the PREFIX30S condition, the first 1,500 bins (30 s) of each recording day were used to estimate the PFDC calibration parameters, and performance was evaluated only on post-prefix samples.

Hyperparameter settings. Tables 6–8 report the dataset-specific and shared settings used for the noncausal configuration of Full UPCSD. Part I gives the dataset and split settings; Part II gives preprocessing and augmentation settings; Part III gives architecture and optimization settings. Split counts are exact, whereas allocation ratios are nominal.

Table 6: Hyperparameter settings of Full UPCSD, Part I: dataset and split settings. Day spans give the elapsed calendar time between the first and last held-out test days; split counts give the numbers of training, validation, and test recording days.

Parameter	Link-CO (700 days)	Link-RTT (757 days)	Indy-RTT (123 days)	Jango-CO (88 days)
Dataset and split settings				
Number of recorded days	155	156	26	20
Train/validation/test days	28/3/124	28/3/125	4/1/21	3/1/16
Nominal training-pool ratio	0.20 of all recording days			
Nominal validation ratio within training pool	0.10			
Prediction target	Position	Position	Velocity	Position
Input/output dimensions	96/2			
Temporal bin size	20 ms			
Noncausal evaluation lag	2 bins (40 ms)			
Sequence construction	Max. 256 bins	Max. 256 bins	60-bin windows	30-bin trials
Window stride	–	–	20 bins (400 ms)	–
Minimum valid sequence length	4 bins	4 bins	20 bins	4 bins

Table 7: Hyperparameter settings of Full UPCSD, Part II: PFDC calibration and data augmentation.

Parameter	Setting
PFDC calibration	
PFDC prefix duration	30 s
Number of prefix bins	1500
Standard-deviation threshold τ	10^{-6}
Calibration scope	Day-specific channel-wise translation and gain from the prefix
Training-reference matching	Not used
Target-label information	Not used
Second-order covariance alignment	Not used
Data augmentation	
Rotation magnitude	0 (disabled)
Gain perturbation σ_g	0.10
Bias perturbation σ_b	0.05
Gaussian noise σ_η	0.10
Channel-dropout probability	0.30 (Link-CO, Link-RTT, Indy-RTT); 0.35 (Jango-CO)
Augmentation application probability	1.0

Table 8: Hyperparameter settings of Full UPCSD, Part III: model architecture and optimization.

Parameter	Setting
Model architecture	
Ablation mode	Full UPCSD
Sequence backbone	LSTM
LSTM hidden units per direction	128
Number of LSTM layers	2
LSTM direction	Bidirectional
Token dimension	64
Model dropout	0.10
Temporal kernel sizes	$\{3, 7, 15, 31\}$
State kernel size	31
Fast-flow kernel size	5
State-flow scale α_S	10
Dynamic-flow scale α_D	10
Spatial rank	64
Multi-scale fusion	Concatenation
Learnable scale fusion	No
Optimization	
Optimizer	AdamW
Loss function	Masked mean-squared error
Initial learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Batch size	32
Maximum number of epochs	40
Gradient clipping norm	5.0
Learning-rate scheduler	None
Checkpoint selection	Minimum validation loss

Temporal configurations. Table 9 summarizes the two implementations for the four longitudinal datasets. The causal scripts retain the sequence construction, PFDC, RDA, and default optimization settings of the noncausal scripts. Both configurations use 64-dimensional tokens, rank-64 channel projections, 31-bin state fitting, 5-bin fast-flow fitting, and temporal kernels of sizes $\{3, 7, 15, 31\}$. For Indy-RTT, prefix statistics are computed before window construction, and overlapping predictions are averaged on the original time axis. In both configurations, calibration bins are excluded from metrics and prediction exports. PFDC uses the effective scale $\bar{\sigma}_{n,c}$ in Eq. (4): prefix standard deviations below $\tau = 10^{-6}$ are replaced by 1, and all other values are retained. Prefix standard deviations use the population convention ($\mathrm{ddof} = 0$).

Table 9: Default temporal configurations of the supplied UPCSD implementations for the four longitudinal datasets.

Setting	Noncausal	Causal
Local fitting offsets	$-h, \ldots, h$, $k = 2h + 1$	$-(k-1), \ldots, 0$
Temporal convolution padding	Centered	Left-only
LSTM layers	2	1
LSTM direction	Bidirectional	Unidirectional
Hidden units per direction	128	128
Prediction–target pairing	$\hat{y}_t$ with y_{t+2}	$\hat{y}_t$ with y_t
Calibration prefix	1,500 bins (30 s)	1,500 bins (30 s)

Daily metric and aggregation. For the four longitudinal datasets, daily joint R^2 is computed from the post-prefix prediction–target pairs after the configuration-specific temporal pairing. Let $\mathcal{E}_n$ denote the evaluated pairs for recording day n, and let $\bar{y}_{n,d}$ be the mean target value in dimension d over these pairs. Then

$$R_n^2 = 1 - \frac{\sum_{t \in \mathcal{E}_n} \sum_{d=1}^{D} (y_{n,t,d} - \hat{y}_{n,t,d})^2}{\sum_{t \in \mathcal{E}_n} \sum_{d=1}^{D} (y_{n,t,d} - \bar{y}_{n,d})^2}. \tag{17}$$

Here t indexes an aligned evaluation pair. For Indy-RTT, overlapping predictions are averaged at each original target time before computing this metric, so each evaluated time bin is counted once. The reported mean and sample standard deviation ($\mathrm{ddof} = 1$) are calculated across recording days, rather than across windows or independent training runs.

NW-CO training and evaluation days. The four training days were August 3, 4, 5, and 6. The August 6 training data were also monitored as a diagnostic validation set. The deployed checkpoint was the final model after 40 epochs, with no early stopping or validation-based checkpoint selection. NW-CO was evaluated on 12 test days between August 7 and September 7, with daily R^2 summarized across these days.

Table 10 summarizes the NW-CO model and training settings. Behavioral targets were normalized using training-label statistics. The reported label offset describes neural–behavioral temporal alignment; it is distinct from neural lookahead. Causality here refers to the use of only current and past neural inputs. The same-time causal pairing in Table 9 applies to the four longitudinal datasets.

Table 10: NW-CO model, calibration, and training settings.

Parameter	NW-CO configuration
Input and calibration	
Input/output dimensions	1024/2
Temporal bin size	20 ms
Label offset	2 bins (40 ms)
Neural lookahead	0 bins
Calibration prefix	1,500 bins (30 s)
Trial length	Untruncated; minimum 4 bins
Model architecture	
Sequence backbone	One-layer unidirectional LSTM
LSTM hidden dimension	128
Token dimension / spatial rank	64 / 64
State fitting	26-bin Hamming window: 25 past bins and current bin
Fast-flow fitting	5-bin Hann window, causal
Temporal kernel sizes	$\{3, 7, 15, 31\}$, causal convolutions
State-flow / dynamic-flow scale	10 / 10
Tokenizer and regression-head dropout	0.1
Training-only augmentation	
Gain / bias / noise scale	0.10 / 0.05 / 0.10
Channel-dropout probability	0.30
Optimization	
Optimizer	AdamW
Learning rate / weight decay	10^{-4} / 10^{-4}
Batch size	32
Gradient clipping norm	5.0
Loss	MSE on normalized targets
Training epochs	40
Checkpoint selection	Final epoch; no early stopping

NW-CO online inference. The NW decoder receives one 1,024-channel spike-count vector every 20 ms. The first 1,500 bins provide channel-wise calibration statistics, which remain fixed during subsequent decoding. Once calibration is complete, the prefix segment from the current trial is replayed to initialize the encoder history and LSTM state; behavioral outputs begin with the next bin. Each subsequent bin updates the causal encoder history and recurrent state and produces a prediction, which is restored to the behavioral scale using the target normalization parameters stored in the checkpoint. Trial boundaries reset the encoder history and recurrent state while retaining the calibration parameters; starting a new recording resets both.

Model parameters remain fixed during online inference.

C Baseline Reproduction and Fairness Controls

Tables 11–14 summarize the noncausal BiLSTM, Transformer, and CEBRA baselines and the NoMAD baseline reported in the causal comparison. In the noncausal benchmark comparison, all methods used the same chronological splits, 20-ms neural bins, a 40-ms prediction lag, and the same post-prefix evaluation samples.

Common split. For the four longitudinal datasets, recording days were sorted chronologically. The earliest 20% formed the development period; the final 10% of that period was used for validation, and all later recording days were held out for testing. The resulting train/validation/test recording-day lists were identical across methods within each dataset.

Common calibration budget. For the PFDC-enabled comparison, every method could access only the first 30 s (1,500 bins) of unlabeled neural activity from the current day. No behavioral label and no neural statistic from later than 30 s was available during calibration or alignment. All reported metrics were calculated only from samples after this prefix. The first-order calibration descriptions below apply to the "+ PFDC" variants; the corresponding variants without PFDC omit this normalization.

Common prediction protocol. The four longitudinal benchmarks contained 96 neural channels and two-dimensional behavioral targets. Neural activity was sampled in 20-ms bins. The noncausal comparison in Table 1 used a two-bin (40-ms) evaluation lag. Causal UPCSD on these four datasets used same-time prediction–target pairing, as specified in Table 9. NW-CO settings are reported separately in Table 10. Target normalization parameters were estimated exclusively from the training labels.

Interpretation of fairness. BiLSTM and Transformer received no train-time data augmentation and no test-time parameter adaptation. NoMAD received an additional unsupervised optimization step because target-day alignment is the defining operation of the original method. CEBRA was evaluated in its stricter source-only form; the 500-iteration supervised-adaptation option was disabled because it would require target-day behavioral labels.

Sequence construction. Link trial boundaries were preserved. The 256-bin implementation cap did not truncate any trial in the evaluated Link datasets. Jango trials were retained at their original 30-bin length. Indy continuous recordings were divided into overlapping 60-bin windows with a 20-bin stride; overlapping predictions were averaged back to the original time axis. NoMAD followed its method-specific LFADS sequence construction using 30-bin non-overlapping windows. Incomplete training/alignment tails were discarded, and the final inference window was padded and trimmed after prediction.

Table 11: Reproduction details for BiLSTM in the noncausal benchmark comparison.

Setting	**BiLSTM** (Graves, 2012)
Original formulation	A supervised sequence-to-sequence regressor without explicit shared-manifold estimation or cross-day parameter adaptation.
Architecture and optimization	A two-layer bidirectional LSTM encoder and one-layer bidirectional LSTM decoder with a LayerNorm–Linear$(256, 64)$–GELU–Dropout(0.1)–Linear$(64, 2)$ head (1,039,042 trainable parameters). Training used MSE and AdamW ($\mathrm{lr} = 10^{-4}$, weight decay 10^{-4}, batch size 32, gradient clipping 5, maximum 40 epochs); the checkpoint with minimum validation MSE was retained.
Calibration and alignment	Each day used the same first-order calibration, with channel-wise translation and gain estimated only from its first 1,500 unlabeled neural bins (30 s). The fitted network remained frozen, with no test-day update or additional alignment.
Labels and augmentation	**No data augmentation**. Behavioral targets were normalized using training labels only. The test prefix contained no behavioral labels, and the model operated on complete trial/window sequences.

Table 12: Reproduction details for Transformer in the noncausal benchmark comparison.

Setting	**Transformer** (Vaswani et al., 2017)
Original formulation	A supervised self-attention sequence regressor without an explicit cross-day alignment objective.
Architecture and optimization	Input projection $96 \to 64$, GELU, LayerNorm, sinusoidal positional encoding, and two pre-normalized encoder layers with four heads, feed-forward dimension 256, dropout 0.1, and non-causal self-attention. The LayerNorm–Linear$(64, 64)$–GELU–Dropout(0.1)–Linear$(64, 2)$ head gave 110,722 trainable parameters. Optimization and checkpoint selection matched the BiLSTM.
Calibration and alignment	The same per-day 30-s unlabeled first-order calibration was applied. The encoder and head remained frozen, with no target-day fine-tuning or latent-space alignment.
Labels and augmentation	**No data augmentation**. Training-label normalization, checkpoint selection, test information, prediction lag, and evaluation samples matched the BiLSTM. The model is non-causal within each trial/window.

Table 13: Reproduction details for NoMAD in the causal benchmark comparison.

Setting	**NoMAD** (Karpowicz et al., 2025)
Original formulation	Nonlinear Manifold Alignment with Dynamics. It freezes behavior-aware LFADS dynamics and the behavioral decoder, then learns a target-day input mapping without target-day behavioral labels by aligning Generator-state distributions while preserving neural reconstruction.
Architecture and optimization	Behavior-aware LFADS used a 100-dimensional Generator state, 30-dimensional factors, four-dimensional inferred controller input, and non-overlapping 30-bin sequences. It was trained for at most 300 epochs using Poisson reconstruction, LFADS KL/L2 regularization, and behavioral MSE; the checkpoint was selected by validation behavioral MSE. Chronological training days were pooled as the source domain.
Calibration and alignment	AlignLFADS received the first 1,500 unlabeled bins from each target day. Its three-layer FFN generator was initialized as an identity, zero-gated residual mapping. Only the read-in, generator, and rate/factor readouts were optimized; source dynamics and the behavioral decoder remained frozen. The loss combined empirical Gaussian KL divergence with target-day Poisson NLL. Alignment used $\mathrm{lr} = 4 \times 10^{-4}$, up to 10,000 epochs, patience 15, gradient clipping 200, covariance jitter 10^{-3}, a 100-epoch NLL ramp, and a deterministic 10% source-state holdout.
Labels and augmentation	No target-day behavioral label was used. For NoMAD + PFDC, the same 30-s per-day first-order calibration was applied before AlignLFADS; raw non-negative activity remained the Poisson target. Native LFADS regularization was retained, with no synthetic cross-day augmentation.

Table 14: Reproduction details for CEBRA in the noncausal benchmark comparison.

Setting	**CEBRA** (Schneider et al., 2023)
Original formulation	A contrastive method for consistent low-dimensional neural embeddings. The behavior-conditioned formulation learned the source embedding from training neural activity and behavior.
Architecture and optimization	The official offset1-model was used with conditional=time_delta, a 16-dimensional embedding, cosine distance, temperature 1, time offset 1, batch size 64, learning rate 3×10^{-4}, and 10,000 iterations. Source trials were registered once as a boundary-aware pooled stream because the sklearn interface did not support the required multi-day continuous-label input. A two-layer LSTM and MLP decoded the embeddings. The decoder used AdamW (lr $= 10^{-4}$, weight decay 10^{-4}) for 40 epochs and was selected by validation MSE.
Calibration and alignment	The reported configuration was **source-only CEBRA**. Each day received the same 30-s unlabeled first-order calibration, but neither encoder nor decoder was fine-tuned on the target day. No additional Procrustes, CORAL, or supervised test-prefix alignment was applied.
Labels and augmentation	Behavioral labels were used only for source training and decoder fitting; validation labels were used only for selection. No test-day label, post-prefix statistic, or data augmentation was used. Pooling introduced at most one artificial time-delta transition per trial boundary; decoder boundaries were preserved.

D Statistical Comparisons across Elapsed-Time Intervals

Within each elapsed-time interval, UPCSD, CEBRA, Transformer, and BiLSTM were compared using a one-way repeated-measures ANOVA, with recording day as the repeated-measures unit. The tables also report two-sided paired t-tests for all intervals. Pairwise significance is interpreted only following a significant omnibus test; values in intervals with a nonsignificant omnibus test are included for completeness. Pairwise p values were adjusted using Holm's procedure across all six comparisons within each dataset and interval. Tables 15–18 report the three comparisons between UPCSD and the PFDC-enabled baselines. Positive differences favor UPCSD.

Table 15: Interval-wise comparisons of post-prefix daily R^2 on Link-CO. Baseline columns report ΔR^2 (UPCSD minus baseline) and Holm-adjusted p values. The omnibus p value is unadjusted; n is the number of paired test days.

Day interval	n	RM-ANOVA $F(\mathrm{df});\ p$	CEBRA	Transformer	BiLSTM
0–100	15	$F(3, 42) = 129.27$ $p = 3.085 \times 10^{-21}$	$+0.282$ (1.261×10^{-8})	$+0.070$ (1.261×10^{-8})	$+0.047$ (2.115×10^{-5})
100–200	15	$F(3, 42) = 78.32$ $p = 3.055 \times 10^{-17}$	$+0.262$ (3.966×10^{-7})	$+0.045$ (1.129×10^{-3})	$+0.064$ (1.746×10^{-5})
200–300	31	$F(3, 90) = 108.94$ $p = 7.477 \times 10^{-30}$	$+0.226$ (6.376×10^{-13})	$+0.026$ (2.571×10^{-3})	$+0.042$ (9.802×10^{-6})
300–400	14	$F(3, 39) = 31.25$ $p = 1.827 \times 10^{-10}$	$+0.113$ (2.033×10^{-6})	$+0.025$ (6.409×10^{-2})	$+0.074$ (4.213×10^{-4})
400–500	20	$F(3, 57) = 30.58$ $p = 6.498 \times 10^{-12}$	$+0.152$ (1.031×10^{-6})	$+0.050$ (3.491×10^{-3})	$+0.112$ (2.910×10^{-7})
500–600	12	$F(3, 33) = 3.11$ $p = 3.970 \times 10^{-2}$	$+0.123$ (2.923×10^{-1})	$+0.049$ (7.600×10^{-2})	$+0.112$ (2.507×10^{-2})
600–700	17	$F(3, 48) = 10.39$ $p = 2.207 \times 10^{-5}$	$+0.074$ (1.145×10^{-2})	$+0.090$ (1.742×10^{-3})	$+0.145$ (1.009×10^{-3})

Table 16: Interval-wise comparisons of post-prefix daily R^2 on Link-RTT. Baseline columns report ΔR^2 (UPCSD minus baseline) and Holm-adjusted p values. The omnibus p value is unadjusted; n is the number of paired test days.

Day interval	n	RM-ANOVA $F(\mathrm{df});\ p$	CEBRA	Transformer	BiLSTM
0–100	27	$F(3,78) = 193.34$ $p = 5.078 \times 10^{-36}$	$+0.166$ (2.133×10^{-14})	$+0.064$ (4.040×10^{-13})	$+0.013$ (3.593×10^{-3})
100–200	4	$F(3,9) = 33.25$ $p = 3.371 \times 10^{-5}$	$+0.135$ (2.960×10^{-2})	$+0.067$ (2.224×10^{-2})	$+0.004$ (7.599×10^{-1})
200–300	9	$F(3,24) = 15.04$ $p = 1.022 \times 10^{-5}$	$+0.071$ (5.985×10^{-3})	$+0.023$ (3.266×10^{-2})	$+0.018$ (3.266×10^{-2})
300–400	20	$F(3,57) = 30.85$ $p = 5.572 \times 10^{-12}$	$+0.137$ (3.602×10^{-5})	$+0.084$ (1.395×10^{-5})	$+0.017$ (7.396×10^{-3})
400–500	14	$F(3,39) = 2.94$ $p = 4.494 \times 10^{-2}$	-0.015 (1.000)	$+0.049$ (1.112×10^{-1})	-0.012 (1.000)
500–600	8	$F(3,21) = 1.64$ $p = 2.096 \times 10^{-1}$	$+0.049$ (8.905×10^{-1})	$+0.044$ (8.905×10^{-1})	$+0.046$ (8.905×10^{-1})
600–700	29	$F(3,84) = 2.00$ $p = 1.208 \times 10^{-1}$	$+0.015$ (1.000)	$+0.026$ (3.548×10^{-1})	-0.003 (1.000)
700–800	14	$F(3,39) = 3.06$ $p = 3.956 \times 10^{-2}$	-0.046 (3.387×10^{-1})	$+0.010$ (9.518×10^{-1})	-0.007 (9.518×10^{-1})

Table 17: Interval-wise comparisons of post-prefix daily R^2 on Indy-RTT. Baseline columns report ΔR^2 (UPCSD minus baseline) and Holm-adjusted p values. The omnibus p value is unadjusted; n is the number of paired test days.

Day interval	n	RM-ANOVA $F(\mathrm{df});\ p$	CEBRA	Transformer	BiLSTM
0–100	17	$F(3,48) = 95.76$ $p = 2.868 \times 10^{-20}$	$+0.257$ (3.390×10^{-8})	$+0.038$ (1.390×10^{-4})	$+0.068$ (4.253×10^{-4})
100–200	4	$F(3,9) = 25.97$ $p = 9.158 \times 10^{-5}$	$+0.221$ (3.801×10^{-2})	$+0.034$ (1.056×10^{-1})	$+0.048$ (1.056×10^{-1})

Table 18: Interval-wise comparisons of post-prefix daily R^2 on Jango-CO. Baseline columns report ΔR^2 (UPCSD minus baseline) and Holm-adjusted p values. The omnibus p value is unadjusted; n is the number of paired test days.

Day interval	n	RM-ANOVA $F(\mathrm{df});\ p$	CEBRA	Transformer	BiLSTM
0–100	16	$F(3,45) = 43.56$ $p = 2.317 \times 10^{-13}$	$+0.080$ (8.127×10^{-6})	$+0.062$ (2.685×10^{-6})	$+0.022$ (9.964×10^{-4})

E Hardware Deployment Details

Table 19 reports the sequence-level deployment configuration for Link-CO, Link-RTT, Indy-RTT, and Jango-CO. These hardware measurements are separate from the NW-CO online experiment in Table 2. Dataset-specific checkpoints were selected by minimum validation loss and converted without additional retraining; prefix calibration was applied before decoder execution. Table 20 reports the matched GPU–Lynxi HE200 results together with the HE200 latency measurements.

The GPU and HE200 results use the same deployment input configuration for each dataset. They therefore quantify platform-level agreement within the deployment pipeline and are not intended to

replace the software-evaluation summary in Table 1. Power values represent device power during active inference, with idle power retained, rather than end-to-end system energy.

Across the four datasets, the absolute difference in mean daily R^2 between HE200 and GPU inference was at most 0.0019. Mean HE200 device power ranged from 6.86 to 7.09 W, corresponding to reductions of 92.34%–92.74% relative to the GPU measurements. These results establish numerical agreement and lower device power for the tested deployment configuration. Per-call latency measurements are reported in Table 20; each call processes one input sequence, not one 20-ms neural bin. These measurements do not by themselves establish a 50-Hz closed-loop deadline, and end-to-end system energy was not evaluated here.

Table 19: GPU and Lynxi HE200 deployment configuration. Dataset-specific checkpoints were selected by minimum validation loss and converted without additional retraining. Prefix calibration was applied before decoder execution.

Setting	Deployment configuration
GPU platform	NVIDIA GeForce RTX 4090
Edge platform	Lynxi HE200
Numerical precision	FP32
Model checkpoint	Dataset-specific checkpoint selected by minimum validation loss, without additional retraining after conversion
Deployed components	Fixed-shape decoder graph comprising the mask-aware state–dynamic tokenizer with low-rank projections, two-layer LSTM, and MLP regression head
Input configuration	One sequence per inference call ($B = 1$); input and output shapes were $[1, T, 96]$ and $[1, T, 2]$, respectively. Link-CO and Link-RTT used sequence-length buckets, whereas Indy-RTT and Jango-CO used $T = 60$ and $T = 30$, respectively
Prefix calibration	Applied before chip inference; the deployed decoder received prefix-calibrated inputs
Power measurement scope	Mean device power during active inference; HE200 power was sampled using `lynxi-smi` at approximately 0.2-s intervals, and idle power was not subtracted

Table 20: Matched GPU–Lynxi HE200 deployment results and HE200 inference latency across the four deployment datasets.

Setting	Link-CO	Link-RTT	Indy-RTT	Jango-CO
GPU mean daily R^2 (post-prefix)	0.5680	0.4621	0.5879	0.8494
HE200 mean daily R^2 (post-prefix)	0.5693	0.4602	0.5879	0.8498
ΔR^2 (HE200 − GPU)	$+0.0013$	-0.0019	$+0.0000$	$+0.0004$
GPU inference power (W)	94.51	90.96	92.61	91.25
HE200 inference power (W)	6.86	6.87	7.09	6.99
Power reduction	92.74%	92.45%	92.34%	92.34%
Completed inference calls	46,290	46,506	28,853	3,488
On-chip latency (ms/call)	25.84 ± 0.30	28.26 ± 0.29	25.44 ± 0.34	12.99 ± 0.35
End-to-end latency (ms/call)	25.96 ± 0.39	28.37 ± 0.37	25.52 ± 0.36	13.07 ± 0.38

Values are mean $\pm$ population standard deviation ($\mathrm{ddof} = 0$) over completed inference calls for the latency rows. Each call processes one sequence with the input shape specified in Table 19. Daily R^2 values are averaged across evaluation days under the deployment configuration in Table 19. Power is mean device power during active inference with idle power retained; reductions are relative to GPU inference. "On-chip" and "end-to-end" follow the timing definitions recorded by the Lynxi runtime. For Link-CO and Link-RTT, completed inference calls were aggregated across sequence-length buckets that were compiled separately for variable-length inputs.