BEHAVIORAL GEOMETRY GUIDES GENERALIZABLE NEURAL REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural recordings can change across sessions even when behavior remains similar. This makes behavior a natural reference for learning shared representations, but pointwise prediction alone does not specify how these representations should be organized. We introduce Behavioral Relational Guidance (BeRG), which uses behavioral geometry as a structural prior for neural pretraining. A frozen behavioral encoder maps behavioral histories and task context to state representations. For each state, a soft neighborhood assigns probability weights to states from other sessions of the same task, giving higher weights to more similar behavioral representations. BeRG matches these neighborhood distributions in a neural decoding subspace, jointly optimizing relational and decoding objectives. The framework supports fixed and learned encoders, while inference requires only neural inputs. We evaluate hand-velocity decoding from macaque motor-cortical recordings during center-out and random-target reaching. Across six neural architectures, BeRG reduces normalized mean squared decoding error by approximately 15–23% on validation data. Compared with fixed-feature guidance, learned guidance increases mean R^2 for 250-ms-ahead velocity readout from frozen representations by approximately 20%. It also improves neighborhood-based velocity readout and behavioral similarity among cross-session neighbors, with benefits in both the decoding subspace and final hidden representations. In longitudinal recordings during a finger-movement task, BeRG further improves decoding on previously unseen recording dates after few-shot calibration. These findings point to a broader role for behavioral data as structural supervision, offering a route toward neural models that remain useful across readouts and recordings.

1 INTRODUCTION

Brain–computer interfaces (BCIs) translate neural population activity into control signals, offering a route to restoring movement and interaction for people with motor impairments (Collinger et al., 2013). Accurate motor decoding is central to this capability, but maintaining performance across recordings remains challenging. Changes in recorded neural populations can alter the mapping from neural activity to behavior and require repeated decoder calibration (Karpowicz et al., 2025). These demands motivate learning neural representations that support accurate behavioral prediction and can be reused with limited calibration data.

Research on neural decoding has advanced through complementary efforts to model population activity, identify behaviorally relevant structure, and reuse models across recordings. Latent dynamical models and behavior-guided approaches learn neural states that support behavioral readout (Pandarinath et al., 2018; Sani et al., 2021). The NDT and POYO model families represent major advances in multisession neural pretraining, pooling heterogeneous recordings to improve neural modeling, behavioral decoding, and adaptation to new recording contexts (Ye et al., 2023; 2025; Azabou et al., 2023; 2025). In parallel, CEBRA uses behavioral information to organize neural embeddings, while BLEND uses behavior as privileged training information for neural-only models (Schneider et al., 2023; Guo et al., 2025). These developments provide complementary foundations for learning shared neural representations and incorporating behavioral supervision.

A central challenge is to learn reusable neural representations of behavior despite changes in the recorded neural populations. Similar movements can be reflected in different neural observations

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.

across sessions, making shared behavioral structure difficult to identify directly from the recordings. Current behavioral targets provide a common reference, but describe only part of the evolving behavioral state. For example, the same hand velocity can occur during acceleration and deceleration, with different histories and subsequent trajectories. Future-velocity readout therefore probes whether neural representations retain accessible information about how movement evolves, extending their evaluation beyond instantaneous decoding. Behavioral history and task context offer a richer basis for relating states across recordings. We investigate how this information can guide shared neural representations to improve current decoding while preserving structure useful for additional behavioral readouts and calibration in new recordings.

We introduce Behavioral Relational Guidance (BeRG), a framework that uses behavioral geometry as a structural prior for neural pretraining. A frozen encoder of behavioral histories and task context defines soft neighborhoods over states from other sessions of the same task. BeRG matches these neighborhood distributions in a subspace on the neural decoding path, jointly optimizing relational supervision and current-behavior decoding to train the shared backbone. Fixed features and learned behavioral encoders provide alternative sources of guidance within the same framework. Inference requires only neural activity. Our contributions are as follows:

1. **BeRG improves motor decoding** across neural backbones by introducing behavioral geometry as a constraint during neural pretraining.
2. **BeRG improves behavioral neighborhood organization and future-velocity readout** from frozen neural representations under both fixed-feature and learned guidance, with the latter providing further gains in future readout.
3. **BeRG improves adaptation to unseen sessions and an unseen animal**, using limited labeled calibration on familiar tasks.

2 Related Work

Neural pretraining and decoding. NDT1 learns population activity with a non-recurrent Transformer (Ye & Pandarinath, 2021); NDT2 extends masked neural pretraining across experimental contexts (Ye et al., 2023). POYO and POYO+ scale decoding across neuronal populations, brain regions, and tasks (Azabou et al., 2023; 2025). NDT3 unifies neural and behavioral sequences through autoregressive modeling (Ye et al., 2025). BeRG adds a complementary signal to neural pretraining: behavioral neighborhoods that guide representations on the decoding path.

Behavioral representations and neural manifolds. PSID and DPAD prioritize behaviorally relevant latent dynamics through linear subspace identification and nonlinear recurrent modeling (Sani et al., 2021; 2024). CEBRA uses behavior- or time-conditioned contrastive sampling for consistent, decodable neural embeddings, including guidance by pretrained visual features (Schneider et al., 2023). BeRG instead matches soft state neighborhoods defined by a separate behavioral encoder during supervised decoding. We analyze both shared hidden states and the decoding subspace.

Privileged information and relational distillation. BLEND supplies behavior to a multimodal neural-reconstruction teacher, then distills outputs, intermediate features, or within-trial neuronal correlations into a neural-only student (Guo et al., 2025). Student training targets reconstruction; behavioral decoding is evaluated downstream. Probabilistic knowledge transfer (PKT) matches neighborhood distributions using KL divergence, supporting handcrafted and learned teachers and cross-modal transfer (Passalis & Tefas, 2018). BeRG builds on these ideas with cross-session state-neighborhood targets from a behavioral encoder that receives no neural observations. Joint relation matching and supervised decoding guide the shared neural model, whose organization, readout capabilities, and new-date calibration are evaluated separately.

3 Approach

BeRG uses behavioral geometry as a training prior for neural decoding (Figure 1). A behavioral encoder defines soft neighborhoods across recordings, and the neural model learns to preserve these relationships in a subspace on its decoding path. Relational guidance and supervised decoding

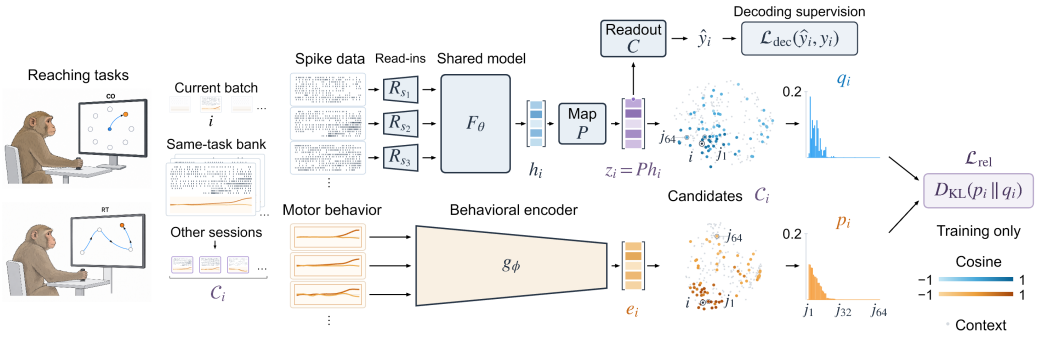

Figure 1: **Overview of Behavioral Relational Guidance (BeRG).** Paired reaching recordings use session-specific neural read-ins and a shared neural backbone. A frozen behavioral encoder supplies soft neighborhood targets p_i over other-session states of the same task. Matching neural neighborhoods q_i in the decoding subspace $z_i = Ph_i$ jointly with decoding trains the shared model; inference uses only neural inputs. Raster ticks preserve binned counts, with schematic within-bin timing. Independent PCA views show real representations; colors denote high-dimensional cosine similarity. The distributions share 64 candidate identities and exclude gray context states.

jointly train the shared neural model; the behavioral encoder provides training targets and is removed for inference.

3.1 Problem Setup and Behavioral Representations

Let i index a time state with recording identity s_i, task identity k_i, and behavioral target $y_i \in \mathbb{R}^{d_y}$. Its neural input is a window X_{w_i} containing token position t_i, and B_i contains the paired behavioral history and available task context. A neural read-in R_{s_i} and shared processor F_θ produce a hidden state, while a separate behavioral encoder g_ϕ produces a behavioral representation:

$$h_i = F_\theta(R_{s_i} X_{w_i})_{t_i} \in \mathbb{R}^d, \qquad e_i = g_\phi(B_i) \in \mathbb{R}^{d_B}. \tag{1}$$

The window notation accommodates both offline and causal decoding; dataset-specific input access is specified in Section 4.

The current decoding target specifies the output at a single time state. Behavioral history and available task context can additionally distinguish states with similar instantaneous outputs but different temporal or task-related conditions. BeRG encodes this information into a behavioral space in which state relationships can be compared. Fixed features and learned encoders provide alternative ways to define these relationships.

We instantiate the behavioral encoder in three ways, differing in how the reference geometry is constructed. Each produces state representations whose similarities define BeRG's relational targets; the neural decoding pathway and relation-matching formulation remain shared.

BeRG-Explicit constructs a fixed representation from the complete behavioral history and task context. The two input blocks are flattened, centered using source-training statistics, and scaled to balance their aggregate contributions before concatenation and normalization. This variant defines behavioral relationships without fitting a behavioral encoder.

BeRG-LIGHT learns a compact behavioral representation from scratch. A temporal Transformer combines kinematic history with time-resolved task context and is trained using behavior-only forecasting and reconstruction objectives. Its learned states define relationships that reflect the temporal structure acquired through this training.

BeRG-MOMENT obtains temporal features from a frozen, pretrained MOMENT-1-small backbone (Goswami et al., 2024). A trainable Fusion Adapter combines these features with task context to produce behavioral states, with the adapter and behavioral heads fitted using predictive and reconstruction objectives.

After encoder preparation, all three mappings remain fixed during neural pretraining. Their state relationships provide supervision alongside current-behavior decoding; the neural model receives no

direct future-behavior prediction loss. Retrospective context, where used, is training-time privileged information; encoder-specific inputs and fitting objectives are detailed in Appendix G.

3.2 Guidance on the Neural Decoding Path

Supervised decoding constrains behavioral predictions but leaves the geometry of intermediate representations underdetermined. We apply behavioral relational guidance to a subspace on the neural decoding path, allowing relational supervision and behavioral prediction to jointly shape the shared neural model:

$$z_i = Ph_i, \qquad \widehat{y}_i = Cz_i + b, \qquad PP^\top = I_r, \tag{2}$$

Here $P \in \mathbb{R}^{r\times d}$ defines orthonormal coordinates for the subspace and $C \in \mathbb{R}^{d_y\times r}$ reads out behavior. We refer to z_i as Common-Port in the analyses. Relational guidance acts on z_i, with gradients propagating through the projection to the shared hidden state h_i. Dimensions, the orthonormal parameterization, and initialization details are provided in Appendix G.

3.3 Cross-Session Neighborhood Matching

BeRG uses paired behavioral and neural observations of each sampled state to link the two representation spaces. This correspondence is defined within a recording and requires neither neuron matching nor matching timestamps across sessions. For each anchor, we compare its relative similarities to candidate states from other sessions of the same task. Behavioral and neural representations need not share coordinates or dimensionality. Following probabilistic relational distillation (Passalis & Tefas, 2018), we express these relationships as distributions over candidates.

Candidates $j \in \mathcal{C}_i$ satisfy $k_j = k_i$ and $s_j \neq s_i$. They are drawn from a refreshed, session-balanced source-training bank, span near to distant behavioral-similarity ranks, and have a cap on each session's contribution. Both modalities use the same candidate identities, so each probability refers to the same paired state.

Let u_i denote the behavioral vector after fixed source-training centering and unit normalization. For learned encoders, $u_i = \mathrm{norm}(e_i - \mu_B)$, with source mean μ_B; Explicit additionally balances its history and context feature blocks. To compare neural states relative to their recording means, we use within-session centering over the current collection $\mathcal{U}$ of anchors and all active-bank rows:

$$\bar{z}_{s,\mathcal{U}} = \frac{1}{|\mathcal{U}_s|}\sum_{j\in\mathcal{U}_s} z_j, \qquad v_i = \mathrm{norm}(z_i - \bar{z}_{s_i,\mathcal{U}}), \tag{3}$$

where $\mathcal{U}_s$ contains rows from recording s and norm denotes numerically stabilized unit normalization. Neural centering is differentiable and confined to the relation branch; decoding uses the uncentered z_i.

For $j \in \mathcal{C}_i$, cosine similarities define behavioral targets p_{ij} and neural probabilities q_{ij}:

$$\begin{aligned} p_{ij} &= \frac{\exp(u_i^\top u_j/\tau_B)}{\sum_{\ell\in\mathcal{C}_i}\exp(u_i^\top u_\ell/\tau_B)}, \\ q_{ij} &= \frac{\exp(v_i^\top v_j/\tau_N)}{\sum_{\ell\in\mathcal{C}_i}\exp(v_i^\top v_\ell/\tau_N)}. \end{aligned} \tag{4}$$

Temperatures τ_B and τ_N control distribution concentration. The behavioral-to-neural KL objective encourages the neural model to preserve the relative affinities specified by the behavioral encoder:

$$\mathcal{L}_{\mathrm{rel}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{C}_i} p_{ij}\log\frac{p_{ij}}{q_{ij}}, \tag{5}$$

where $\mathcal{A}$ is the valid anchor collection and p_{ij} are stop-gradient targets. Sampling quotas, normalization details, and source-training temperature calibration are specified in Appendix G.

3.4 Joint Training and Inference

At neural optimization update t, BeRG combines the supervised decoding loss with relational guidance:

$$\mathcal{L}(t) = \mathcal{L}_{\mathrm{dec}} + a(t)\lambda\mathcal{L}_{\mathrm{rel}}. \tag{6}$$

Here $\mathcal{L}_{\text{dec}}$ is mean squared prediction error, λ is a fixed relation weight, and $a(t)$ is its activation schedule.

Both objectives update the neural read-in and shared processor through the decoding subspace. Decoding updates P, C, and b throughout training. The relation loss directly updates P during an initial phase; afterward, the relation branch evaluates $\text{sg}(P)h_i$. This stops its parameter gradient to P while retaining gradients to h_i. The relation loss has no direct dependence on C or b, and the behavioral encoder remains frozen.

Relational training reuses anchor states from the decoding forward pass. All bank samples are forwarded through the current neural model with gradients retained. These additional forwards contribute to the relation objective without an additional candidate decoding loss. Loss averaging, weights, and schedules are specified per dataset in Appendix G.

Inference uses only the neural prediction path. The behavioral encoder, relation bank, and relation-specific normalization are removed. The linear maps can be folded into $W_{\text{fold}} = CP$ without changing predictions, retaining the neural model's original form of inference.

4 Experiments

Our experiments first test whether behavioral relational guidance improves neural decoding and examine the sources of the gain. We then assess what learned behavioral encoders contribute beyond fixed features through future-behavior readouts and layerwise organization analyses. Finally, we evaluate adaptation to unseen recording dates and an unseen subject, and examine transfer boundaries.

We study decoding and representation organization using the Perich reaching recordings (Perich et al., 2023), comprising 99 sessions from three animals across center-out and random-target tasks. The main comparisons use source-validation intervals from pretraining sessions. These are development endpoints because 19 sessions also inform behavioral-encoder selection. LINK (Temmar et al., 2025b;a) provides longitudinal recordings from one animal, divided chronologically into 94 source, 17 development, and 45 final dates excluded from pretraining. Perich uses offline neural windows, whereas LINK uses causal neural histories.

Within each window-backbone comparison, Base and BeRG share the neural architecture, paired initialization, and training-update budget. Additional candidate forwards increase computation, so equal updates do not imply equal computational cost. We distinguish trained-decoder performance from newly fitted readouts of frozen representations. Perich comparisons report one pretraining seed per condition: seed 11 for window backbones and seed 42 for POYO-Light. LINK repetitions are reported separately. Detailed splits, metrics, optimization settings, and recorded costs are provided in Appendices A, G, and E.

4.1 Decoding Gains from Behavioral Relational Guidance

BeRG-LIGHT improves source-validation decoding across all six tested neural backbones under a common evaluation protocol (Table 1; Figure 2a). These include window and causal Transformers, a temporal convolutional network (TCN), a bidirectional gated recurrent unit (BiGRU), a multilayer perceptron (MLP), and POYO-Light. POYO-Light training differences are specified in Table 1; these runs do not use an official pretrained checkpoint (Appendix D.3).

Neural backbone	Base	+BeRG	ΔR^2
Window Transformer	0.8561	**0.8887**	**+0.0326**
Causal Transformer	0.8321	**0.8639**	**+0.0318**
TCN	0.7934	**0.8413**	**+0.0479**
BiGRU	0.8501	**0.8817**	**+0.0316**
MLP	0.8502	**0.8757**	**+0.0255**
POYO-Light†	0.8817	**0.8999**	**+0.0182**

Table 1: **Current decoding across neural backbones.** Pooled R^2 on matched source-validation queries from 99 Perich sessions; +BeRG: BeRG-LIGHT. One pretraining seed per condition. † POYO-Light uses a longer training budget and BeRG-only weight restarts (Appendix D.3).

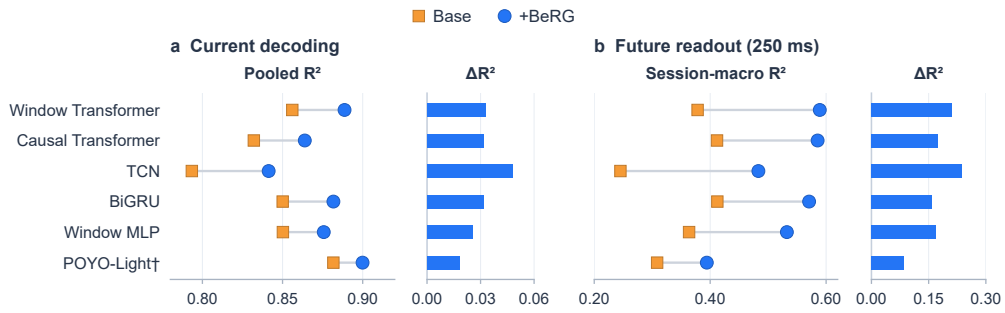

Figure 2: **BeRG improves current decoding and future velocity readout across neural backbones.** **(a)** Native current-velocity decoding (pooled R^2). **(b)** Offline ridge readout of velocity 250 ms after the query center from frozen neural representations (session-macro R^2). Points show scores; bars show BeRG-LIGHT minus Base gains. Both panels use validation data from 99 pretraining-exposed sessions, with different query cohorts. All comparisons use one pretraining seed. Absolute-score axes are truncated. † POYO-Light retains the historical training differences described in Table 1.

Training condition	Current $R^2 \uparrow$	Future $R^2 \uparrow$	Conditional $\Delta R^2 \uparrow$
Base	0.8561	0.3783	+0.1168
Base + extra decoding	0.8735	—	—
BeRG-LIGHT, shuffled	0.8775	—	—
BeRG-Explicit	0.8880	0.4898	+0.2022
BeRG-LIGHT	0.8887	0.5891	+0.2939
BeRG-MOMENT	0.8889	0.5705	+0.2766
LIGHT-FUTURE	0.8505	0.5796	+0.2804

Table 2: **Behavioral encoders and auxiliary objectives.** Current: trained-decoder pooled R^2. Future: frozen final-hidden ridge readout, session-macro R^2 at 250 ms. Conditional: gain over a history-only readout at 250 ms. One matched neural seed; source validation. Extra decoding adds candidate supervision without matched computation. —: not evaluated.

Behavioral encoder comparisons. Fixed behavioral guidance also improves Window Transformer decoding (Table S2A). Explicit and LIGHT achieve comparable current-decoding scores, $R^2 = 0.8880$ and 0.8887, respectively. Encoding the complete behavioral history and context with fixed features therefore already supports strong decoding gains.

Contributions of candidate supervision and relational correspondence. All conditions retain the current-velocity decoding objective. We ask whether the additional gains can be explained by ordinary supervision on the candidate samples, or whether assigning behavioral relations to the corresponding neural states contributes beyond this supervision. We use the shared single-seed setting in Table 2. Adding ordinary decoding supervision to the extra candidate windows improves Base from $R^2 = 0.8561$ to 0.8735, below both Explicit (0.8880) and LIGHT (0.8887). Behavioral relational guidance therefore outperforms this tested use of additional candidate supervision.

Permuting the correspondence between LIGHT's target probabilities and candidate identities reduces R^2 from 0.8887 to 0.8775. Correct relational assignments thus contribute to the observed performance. The shuffled condition retains behavior-dependent candidate selection, so its remaining gain over Base cannot be interpreted as a control with all behavioral structure removed.

Compatibility with an adapted pretrained encoder. Using the adapted MOMENT interface described above, BeRG-MOMENT reaches a current-decoding R^2 of 0.8889 in the same single-seed comparison (Table 2). This supports the use of an adapted time-series foundation model as a behavioral interface. The contribution of its public pretrained weights is not isolated by this comparison.

Because current-decoding accuracy does not clearly distinguish learned from fixed guidance, we next examine future-behavior readouts from their frozen neural representations.

4.2 Future-Behavior Readout from Frozen Neural Representations

Current-velocity decoding measures performance on the pretraining target, but does not establish whether the learned representation supports other behaviorally relevant readouts. Motivated by predictive approaches to identifying behaviorally relevant neural states (Sani et al., 2021), we use future-velocity readout as a complementary probe of temporal behavioral information. We freeze the neural model and fit linear ridge readouts at 100 and 250 ms, testing both fixed and learned guidance against Base and assessing their contributions beyond the supplied velocity history. Each query session is excluded from readout fitting but was included in neural pretraining; this evaluates source-domain readout, with the full protocol provided in Appendix C.1.

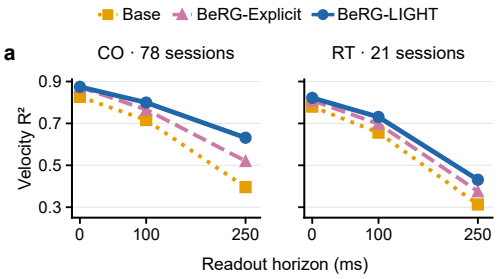

Figure 3: **BeRG improves future velocity readout. (a)** Frozen final-hidden ridge readout in center-out (CO; 78 sessions) and random-target (RT; 21). All panels: one pretraining seed, source validation from pretraining-exposed sessions; a,c show equal-session means.

Across the six tested backbones, LIGHT improves 250-ms future readout over Base by 0.086–0.238 in session-macro R^2 (Figure 2b). These readouts use a different query cohort and aggregation from the native current decoder. We next compare fixed and learned guidance using the Window Transformer.

Learned guidance improves predictive accessibility. Despite comparable current-decoding performance, LIGHT yields better future readout than Explicit, with a larger advantage at 250 than at 100 ms (Figure 3a). Session-level comparisons and task-family analyses show broadly distributed gains across the source recordings (Figure 3a,b). LIGHT also provides a larger increment when neural features are added to a velocity-history readout (Figure 3c), supporting additional linear predictability beyond the measured history. This control does not cover all task context used by the future-supervised behavioral encoder. A separate single-seed comparison with adapted MOMENT likewise shows improved future readout over fixed guidance (Table 2).

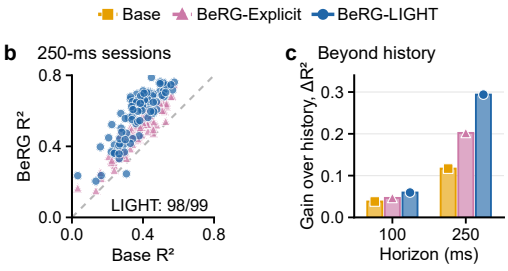

Figure 3: **Future velocity readout (continued). (b)** Paired 250-ms session scores; LIGHT improves over Base in 98 of 99 sessions. **(c)** Incremental R^2 from adding final hidden states to a velocity-history baseline.

Objective choice affects the combination of decoding and readout benefits. Replacing relational guidance with direct prediction of future teacher representations improves future readout over Base, but does not reproduce LIGHT's current-decoding gain in the single-seed comparison (Table 2). Transition-only guidance also improves current decoding, whereas adding transition or conditional-transition guidance does not further improve LIGHT under the tested recipes (Appendix B.3, Table S5). These results support the present relational objective for jointly improving current decoding and future readout, while the limited objective comparisons leave broader conclusions about alternative supervision open.

4.3 Behavioral Organization along the Neural Decoding Pathway

We next examine how proximity in neural representation space reflects behavioral similarity across source recordings. At multiple layers, we predict current velocity from 20 neighbors drawn from other sessions, using standardized Euclidean distance without dimensionality reduction. The full protocol is provided in Appendix C.2.

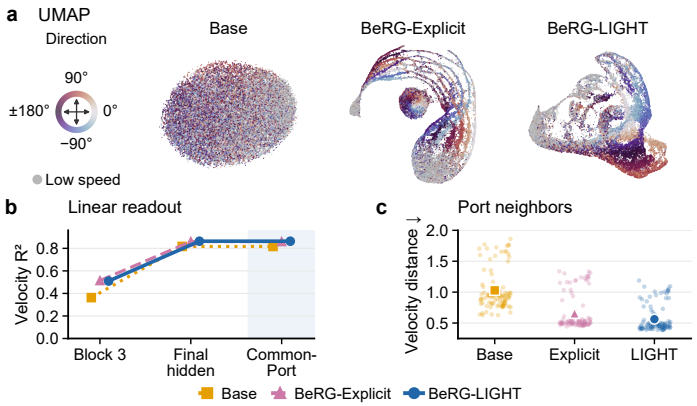

Figure 4: **BeRG improves behavioral organization. (a)** Independent, unsupervised Common-Port UMAPs; color: movement direction; gray: low speed. **(b)** Linear velocity readout. **(c)** Mean normalized true velocity distance to individual cross-session Common-Port neighbors. Large/faint markers: equal-session means/session scores. Seed 11; 99 pretraining-exposed source-validation sessions (Appendix F.2).

Behavioral guidance reorganizes local neighborhoods. Both BeRG variants substantially improve neighborhood velocity readout at Common-Port, where Base already supports accurate linear readout (Figures 4b and S4). The kNN and linear readouts use different query cohorts and provide complementary diagnostics, indicating a pronounced change in local organization at an interface where behavior was already linearly accessible. The mean normalized velocity distance to individual cross-session neighbors also decreases, with LIGHT outperforming Explicit (Figure 4c).

The learned encoder's advantage depends on layer and diagnostic. Improvements extend to the final hidden state within the neural backbone (Figure S4). LIGHT outperforms Explicit at final hidden and Common-Port, whereas Explicit leads at Block 3. This ordering holds under both tested distance metrics in the reported seed (Table S7). Individual-neighbor similarity provides a further distinction: at final hidden, Explicit has a larger mean velocity distance to its neighbors than Base despite better averaged-neighbor readout (Figure S2a). Thus, improved neighborhood prediction does not imply uniformly improved behavioral proximity.

The cross-session neighbor atlas and independently fitted UMAP projections illustrate clearer movement-direction organization with BeRG at Common-Port (Figures S4 and 4a). These single-seed displays provide qualitative context; conclusions about neighborhood quality rest on measurements in the original feature spaces. The findings support improved local behavioral organization, with the largest changes at the decoding interface and additional changes within the backbone.

4.4 Adaptation to New Recording Dates and Subjects

We next evaluate whether BeRG supports supervised adaptation to recordings excluded from pretraining. We consider new recording dates within one animal and a new animal performing familiar tasks, followed by a transfer setting involving a different task.

Calibration on new recording dates. On LINK, we evaluate 45 later recording dates from one animal with physically matched channels. Each date starts independently from its pretrained checkpoint. We fit the read-in and final linear readout using 50 recent labeled trials, keep the shared backbone frozen, and evaluate on subsequent trials. The adaptation protocol is selected on separate development dates. Across three pretraining seeds, mean decoding R^2, with equal weighting across dates, increases from 0.4074 for Base to 0.4449 for BeRG-LIGHT. The paired gain is 0.0375 ± 0.0045 (mean $\pm$ standard deviation across seeds), with positive average gains in all three seeds (Figure S5). Without calibration, however, BeRG-LIGHT performs worse than Base on average in every seed. The benefit therefore concerns adaptation using recent labels. A single-seed sweep also shows positive mean gains at the other tested nonzero label budgets (Figure 5). Because optimization budgets vary with label budgets, these curves characterize the tested calibration settings.

Adaptation to an unseen animal. We evaluate Monkey T, whose recordings are excluded from pretraining, on 12 sessions comprising six center-out and six random-target reaching sessions. Both

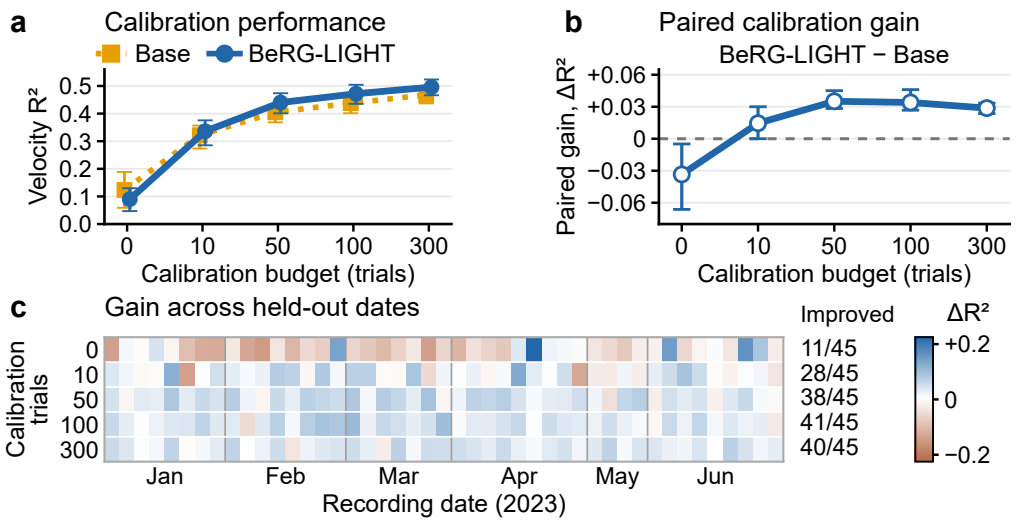

Figure 5: **Calibration improves decoding on held-out dates. (a)** Velocity readout; **(b)** gains over Base; **(c)** all 45 date-level gains and positive-date counts. Points average 45 dates; intervals: 95% moving-block bootstrap. One animal, one pretraining seed; read-in/head adaptation, with 75 shared evaluation trials per date. Without calibration, LIGHT underperforms Base.

task families are represented in source training. Here, only the target read-in is fitted; the shared backbone and pretrained decoder remain frozen. Using 5% of target-training labels, BeRG-LIGHT increases internal-test R^2 from 0.5370 to 0.5716. With all target-training labels, it increases R^2 from 0.7263 to 0.7603. These single-seed results support improved same-task adaptation to the tested unseen animal. Target-validation labels are additionally used for checkpoint selection, so the stated fractions describe the training-label budget rather than the total target-side labeling cost.

Limits under a different transfer setting. On MC-Maze, fitting a target read-in and a separate decoder with the backbone frozen yields lower R^2 for BeRG-LIGHT than Base at both 5% and 100% target-training labels, with differences of -0.0089 and -0.0071, respectively. This setting changes the task, recording, and decoder interface together; the decline cannot be attributed to task differences alone. Detailed transfer results and protocols are provided in Table S4 and Appendices B.1 and B.2.

5 Discussion and Conclusion

BeRG complements multi-recording pretraining with behavioral relationships. Its encoder comparisons show that current-decoding accuracy alone cannot characterize a representation's predictive accessibility or neighborhood organization. The Common-Port results indicate metric reorganization where behavior was already linearly accessible; final-hidden changes extend into the backbone. Neither establishes a global shared manifold or a causal geometry-to-calibration mechanism. A universal encoder ranking is also unsupported: Explicit exceeds LIGHT in final-hidden count readouts and has comparable dropout performance (Table S8).

Behavioral guidance supervises mappings learned from paired recordings. It may favor reuse within the pretraining behavioral domain without supplying mappings for unseen tasks. Calibration and transfer outcomes are compatible with this interpretation but do not isolate it: MC-Maze changes recording and interface alongside task, with a frozen core. Broader paired pretraining and backbone adaptation remain open directions. LINK covers one animal; the reported Perich and cross-subject/cross-task comparisons use single pretraining seeds and do not quantify training-seed variability. Preliminary POYO-Light runs do not establish improvement of a released foundation checkpoint. Teacher fitting and candidate forwards add cost (Appendix E), and unresolved preprocessing support precludes end-to-end online-causality claims. Within these limits, behavioral geometry provides a useful prior for neural decoding, predictive accessibility, and supervised calibration.

AI USE STATEMENT

We used large language models to assist with drafting and polishing manuscript sections and identifying relevant literature. We also used these tools to provide feedback on experimental design, assist with implementation, and support the interpretation of results. The authors take full responsibility for the accuracy, originality, and integrity of the final manuscript.

REFERENCES

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44937–44956. Curran Associates, Inc., 2023. doi: 10.52202/075280-1948. URL https://arxiv.org/abs/2310.16046.

Mehdi Azabou, Krystal Pan, Vinam Arora, Ian Knight, Eva Dyer, and Blake A. Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *International Conference on Learning Representations*, pp. 59654–59677, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/953390c834451505703c9da45de634d8-Paper-Conference.pdf.

Mark Churchland and Matthew Kaufman. MC_Maze: macaque primary motor and dorsal premotor cortex spiking activity during delayed reaching. DANDI Archive, 2022. URL https://dandiarchive.org/dandiset/000128/0.220113.0400. Data set, DANDI:000128, version 0.220113.0400.

Jennifer L. Collinger, Brian Wodlinger, John E. Downey, Wei Wang, Elizabeth C. Tyler-Kabara, Douglas J. Weber, Angus J. C. McMorland, Meel Velliste, Michael L. Boninger, and Andrew B. Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866):557–564, 2013. doi: 10.1016/s0140-6736(12)61816-9. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3641862/.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16115–16152. PMLR, 2024. URL https://arxiv.org/abs/2402.03885.

Zhengrui Guo, Fangxu Zhou, Wei Wu, Qichen Sun, Lishuang Feng, Jinzhuo Wang, and Hao Chen. BLEND: Behavior-guided neural population dynamics modeling via privileged knowledge distillation. In *International Conference on Learning Representations*, volume 2025, pp. 78870–78894, 2025. URL https://arxiv.org/abs/2410.13872.

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://pmc.ncbi.nlm.nih.gov/articles/PMC12089531/.

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, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018. doi: 10.1038/s41592-018-0109-9. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6380887/.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018. URL https://arxiv.org/abs/1803.10837.

Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, Matthew T. Kaufman, Mark Churchland, Mehrdad Jazayeri, Lee E. Miller, Jonathan Pillow, Il Memming Park, Eva L. Dyer, and Chethan Pandarinath. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://arxiv.org/abs/2109.04463.

Matthew G. Perich, Lee E. Miller, Mehdi Azabou, and Eva L. Dyer. Long-term recordings of motor and premotor cortical spiking activity during reaching in monkeys. DANDI Archive, 2023. URL https://dandiarchive.org/dandiset/000688. Data set, DANDI:000688.

Omid G. Sani, Hamidreza Abbaspourazad, Yan T. Wong, Bijan Pesaran, and Maryam M. Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021. doi: 10.1038/s41593-020-00733-0. URL https://www.nature.com/articles/s41593-020-00733-0.

Omid G. Sani, Bijan Pesaran, and Maryam M. Shanechi. Dissociative and prioritized modeling of behaviorally relevant neural dynamics using recurrent neural networks. *Nature Neuroscience*, 27 (10):2033–2045, 2024. doi: 10.1038/s41593-024-01731-2. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC11452342/.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023. doi: 10.1038/s41586-023-06031-6. URL https://arxiv.org/abs/2204.00673.

Hisham Temmar, Nina Gill, Yixuan Wang, 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. LINK: Long-term intracortical neural activity and kinematics. DANDI Archive, 2025a. URL https://dandiarchive.org/dandiset/001201. Data set, DANDI:001201.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-4642. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/cb463f73a35802996546ac8e8b1b2743-Paper-Datasets_and_Benchmarks_Track.pdf.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with Neural Data Transformers. *Neurons, Behavior, Data analysis, and Theory*, 5(3):1–18, 2021. doi: 10.51628/001c.27358. URL https://arxiv.org/abs/2108.01210.

Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context pretraining for neural spiking activity. In *Advances in Neural Information Processing Systems*, volume 36, pp. 80352–80374. Curran Associates, Inc., 2023. doi: 10.52202/075280-3523. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fe51de4e7baf52e743b679e3bdba7905-Paper-Conference.pdf.

Joel Ye, Fabio Rizzoglio, Adam Smoulder, Hongwei Mao, Xuan Ma, Patrick Marino, Raeed Chowdhury, Dalton Moore, Gary Blumenthal, William Hockeimer, Nicolas G. Kunigk, J. Patrick Mayo, Aaron P. Batista, Steven Chase, Adam Rouse, Michael L. Boninger, Charles Greenspon, Andrew B. Schwartz, Nicholas G. Hatsopoulos, Lee E. Miller, Kristofer E. Bouchard, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt. A generalist intracortical motor decoder. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3692. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a00000e6a2208172700510bcd69d48e9-Paper-Conference.pdf.

A Datasets and Evaluation Details

A.1 Datasets and splits

Perich pretraining uses 99 sessions: 78 center-out and 21 random-target sessions, with 68, 3, and 28 sessions from animals C, J, and M, respectively. Monkey T contributes six sessions of each task. MC-Maze from the Neural Latents Benchmark (Pei et al., 2021; Churchland & Kaufman, 2022) uses the single `jenkins_maze_train` recording. Our LINK evaluation uses 156 recording dates from one animal performing the random-target finger task, with 96 electrodes. This subset is divided chronologically into 94 source, 17 development, and 45 final dates.

The Perich behavioral teacher is fitted on training intervals from 80 sessions and selected using validation intervals from the other 19. Neural pretraining uses training intervals from all 99 source sessions. Consequently, aggregate source-validation performance is a development endpoint and is not independent of teacher selection.

For the window backbones, Perich spikes are binned at 50 ms, transformed by $\log(1 + x)$, and standardized per channel using each session's training data. Two-dimensional hand velocity uses normalization fitted on pooled source-training data. We retain the existing within-session training and validation intervals. LINK uses 96 physically matched spiking-band-power channels at 20-ms resolution to predict two finger-group velocities. Its model consumes 50-bin causal histories and predicts at the endpoint. Source fitting uses the first 270 retained trials per source date; trials 271–300 provide source validation. LINK normalization uses source-training data throughout.

Table S1: **Datasets used for pretraining and evaluation.** LINK source, development, and final dates are disjoint and come from one animal. CO denotes center-out; RT and LINK's RD denote random-target tasks; SBP denotes spiking-band power. Counts refer to the subsets used in this study.

Subset	Animals	Sessions / dates	Task	Signal	Role
Perich source	3	99	CO / RT	Spikes	Neural pretraining; source evaluation
Perich Monkey T	1	12	CO / RT	Spikes	Cross-animal target
NLB MC-Maze	1	1	Maze	Spikes	Task / recording / interface shift
LINK source	1	94	RD	SBP	Neural pretraining
LINK development	same	17	RD	SBP	Adaptation protocol selection
LINK final	same	45	RD	SBP	Frozen target-date evaluation

A.2 Decoding metrics and reported seeds

The primary Perich metric is pooled normalized R^2. For scored records $\mathcal{I}$, normalized targets y_{id}, predictions $\hat{y}_{id}$, and the mean $\bar{y}_d$ of output d over these records, it is

$$R^2(\mathcal{I}) = 1 - \frac{\sum_{i\in\mathcal{I}} \sum_d (y_{id} - \hat{y}_{id})^2}{\sum_{i\in\mathcal{I}} \sum_d (y_{id} - \bar{y}_d)^2}. \tag{7}$$

The primary score pools records across sessions before computing this quantity. The secondary macro score computes it within each session and averages sessions equally. This differs from averaging output-wise R^2, as used for LINK and the historical POYO-Light evaluation in Table S10.

Table 1 evaluates all six backbones on the same 99 source sessions, 8,161 one-second windows, and 130,576 scored positions. Targets, source-training global normalization, and the pooled metric are identical. Window backbones consume twenty 50-ms bins and score the last 16 tokens. POYO-Light retains its native event input and is queried at the same bin centers; its frozen reevaluation is detailed in Appendix D.3.

The five window backbones use fixed 20,000-update checkpoints. Base and LIGHT share initialization and update budgets within each of these backbones, but LIGHT adds candidate forwards. All five window backbones report seed 11. POYO-Light uses seed 42 and historical epoch-60 checkpoints after 46,680 updates, with unmatched restart histories. For consistent single-seed reporting, all Window Transformer tables and figures use the fixed historical seed 11, rather than a seed selected by performance. Additional repetitions remain in the experiment archive; displayed Perich

results do not estimate training-seed uncertainty. A shared evaluation does not equate training budgets, training sampling, neural preprocessing, or total computation. The Window Transformer and BiGRU use bidirectional context, the TCN uses symmetric convolutions, and the MLP accesses the full window. The causal Transformer uses a prefix attention mask; this does not independently establish end-to-end causal preprocessing.

Table S2A reports the seed-11 comparison of Base, Explicit, and LIGHT, including session-macro scores. Panel B retains historical same-session test scores at fixed checkpoints from seed 11. These test scores are separate from the source-validation comparison in Table 1.

Table S2: **Additional current-decoding comparisons on Perich.** (A) Fixed and learned guidance with the Window Transformer: the fixed historical pretraining seed 11 on source validation. Pooled R^2 is primary; macro R^2 weights sessions equally, and Δ is the paired pooled gain over Base. (B) Historical same-session test results from one seed; architecture and context vary. Bold marks the best primary score and largest primary gain within each comparison.

Condition	Pooled R^2 ↑	Macro R^2 ↑	Paired Δ vs. Base
A. Window Transformer; source validation; seed 11			
Base	0.8561	0.8423	—
BeRG-Explicit	0.8880	0.8786	+0.0319
BeRG-LIGHT	**0.8887**	0.8793	**+0.0326**
B. Historical same-session test; one pretraining seed			
Processor / context	Base	BeRG-LIGHT	ΔR^2
Window Transformer	0.8493	**0.8824**	**+0.0330**
Causal Transformer	0.8265	**0.8558**	**+0.0294**
Temporal convolution	0.7845	**0.8317**	**+0.0472**

B Calibration, Transfer, and Transition Controls

B.1 LINK calibration

LINK normalization is fitted on source-training data. The 17 development dates determine the adaptation protocol; the 45 final dates are evaluated without target-validation checkpoint selection or a target teacher loss. Final dates had been evaluated before the additional seed repetitions; the repetitions apply a fixed protocol to the same final set. Each date starts independently from its source checkpoint. The final dates span January 10 to June 22, 2023, lying 167–330 days after the last source date. At recent 50, LIGHT improves 38, 40, and 43 of the 45 dates for pretraining seeds 11, 22, and 33, respectively.

Recent 50-trial calibration uses retained trials 251–300, with 300 read-in and final linear readout updates at learning rate 3×10^{-4}; the shared processor and projection P remain frozen. Adaptation uses AdamW with zero weight decay, batches of 48 sampled with replacement, and gradient-norm clipping at 1; all modules remain in evaluation mode. Evaluation uses trials 301–375. For recorded two-dimensional finger position p_t, the decoding label is the forward difference $v_t = (p_{t+1} - p_t)/0.02$, whereas the neural input spans bins $t - 49$ through t. Support and evaluation windows require their complete input history and the label's $t + 1$ position sample to lie within the respective retained-trial intervals; discontinuities are rejected. Scores first average the separate R^2 values for the two finger-group velocities within date, then weight dates equally. Pretraining seeds are 11, 22, and 33, with adaptation seed fixed at 11. The 10/100/300-trial read-in/head conditions use 100/600/600 updates. The full-model 300-trial condition instead uses 300 updates. Varying the label budget thus also varies optimization. Bootstrap and date-display procedures are given in Appendix F.

Table S3: **Calibration on 45 LINK dates excluded from pretraining.** Scores average the two output-wise R^2 values within date, then 45 dates equally, for one animal. (A) Mean ± SD over three pretraining seeds with a fixed adaptation seed. (B) Additional calibration budgets from one pretraining seed. Adaptation updates the input interface and decoder while freezing the shared core, except for the full-model row. Recent 50 uses trials 251–300; every condition is evaluated on trials 301–375.

Labeled trials	Base R^2 ↑	BeRG-LIGHT R^2 ↑	Paired ΔR^2
A. Three pretraining seeds; fixed adaptation seed			
0	**0.1252±0.0316**	0.0979±0.0346	−0.0273±0.0058
50 (recent)	0.4074±0.0076	**0.4449±0.0053**	+0.0375±0.0045
B. Additional budgets; historical pretraining seed 11 only			
10	0.3216	**0.3362**	+0.0145
100	0.4379	**0.4719**	+0.0340
300	0.4673	**0.4961**	+0.0287
300 (full model)	0.4339	**0.4933**	+0.0594

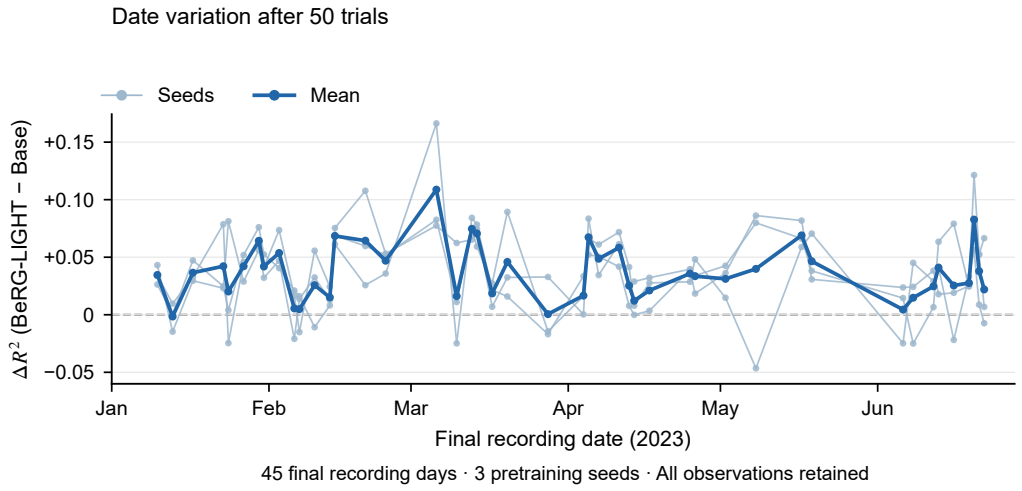

Figure S1: **Date-level variation after recent 50-trial calibration.** Light traces show paired BeRG-LIGHT minus Base ΔR^2 for three pretraining seeds; the dark trace is their within-date mean. All 135 contrasts from 45 dates in one animal are shown, including 14 negative contrasts. The mean gain is positive on 44 of 45 dates. Calendar spacing is preserved, and lines connect observed dates without smoothing. Every date is adapted independently from the source checkpoint. The traces describe date variation, not independent training replicates or an uncertainty band.

B.2 Cross-animal and MC-Maze adaptation

Table S4 reports historical target internal-test scores from pretraining seed 11 and support/adaptation seed 42. Adaptation uses the read-in alone for Monkey T and the read-in plus an independent target decoder for MC-Maze. The shared core remains frozen even at the 100% label budget. Budgets count labeled target-training support; target validation is additionally used for checkpoint selection. MC-Maze changes the task, recording, and decoder interface together, so this comparison does not isolate task identity or evaluate full-backbone fine-tuning.

The archived adaptation protocol partitions each target-training session into complete trial chunks and residual continuous blocks of up to 100 bins, requiring at least 20 bins per chunk. A deterministic session-specific shuffle with seed 42 selects the shortest prefix reaching the requested fraction of eligible training bins; rounding to complete chunks can exceed 5%. The 100% condition uses all eligible chunks. Neural-channel normalization is fitted only to selected support, while behavior normalization remains fixed from source training. Each budget starts from the source model with

newly initialized target read-ins. Monkey T's 12 read-ins are optimized together, with 32 windows from every session per update and a shared frozen decoder. MC-Maze also fits a separate target decoder initialized from the source decoder. AdamW uses learning rate 3×10^{-4}, betas $(0.9, 0.95)$, weight decay 0.01 and gradient-norm clipping at 1. The schedule allows 10,000 updates with 500-update learning-rate warmup. Target validation is evaluated every 250 updates; after at least 2,000 updates, eight evaluations without improvement trigger early stopping. The checkpoint with highest pooled target-validation R^2 is selected for internal-test evaluation. The target score pools predictions across sessions and both physical-velocity coordinates before computing R^2, after reversing source normalization.

Table S4: **Transfer to a new animal and to MC-Maze.** Internal-test R^2 is reported at two target-training label budgets using one pretraining and adaptation seed. Monkey T includes 12 sessions; MC-Maze includes one recording. The shared neural core remains frozen at both budgets, with adaptation restricted to the input interface and, for MC-Maze, an independent target decoder. Target validation is additionally used for checkpoint selection. MC-Maze combines task, recording, and interface changes.

Target	Label budget	Base $R^2 \uparrow$	BeRG-LIGHT $R^2 \uparrow$	ΔR^2
Monkey T	5%	0.5370	**0.5716**	+0.0346
Monkey T	100%	0.7263	**0.7603**	+0.0340
MC-Maze	5%	**0.6662**	0.6573	−0.0089
MC-Maze	100%	**0.8628**	0.8557	−0.0071

A separate exploratory comparison tests whether adding Maze behavior to teacher training improves this transfer without Maze neural pretraining. Matched MOMENT–Fusion teachers use source-only or source-plus-Maze behavior under the same total training budget; both guided neural models are pretrained only on source recordings. With ordinary frozen-core adaptation, the source-plus-Maze condition reduces 5% target-validation R^2 from 0.6835 to 0.6645. At 100% labels, it gives a small increase from 0.8670 to 0.8707. These development endpoints are separate from the internal-test comparison above. They show that the tested behavioral exposure does not improve low-label adaptation, while the full-label effect is small and positive.

B.3 Transition objectives

Objectives. The transition kernel compares behavior-predicted innovations at 100 and 250 ms. A separately prepared frozen behavioral dynamics teacher predicts a 64-dimensional future-trajectory latent mean and uncertainty from past behavior with restricted context. We subtract the latent encoding of a constant-current-velocity trajectory from that predicted mean and divide by source-training RMS scales (0.5% tail winsorization, floor 10^{-4}). The exported target uses the predicted future, not each sample's realized future. All three reported runs use mean-distance kernels with uncertainty-dependent confidence weights:

$$p_{ij}^{B,H} \propto \exp\{-\mathrm{MSE}(m_{i,H}, m_{j,H})/\tau_{B,H}\}, \qquad H \in \{100, 250\}\,\mathrm{ms}.$$

Here $m_{i,H}$ is the normalized innovation. For the neural distribution, let $g_t = [\hat{v}_t, z_{t,3:32}]$ and form source-scaled differences $\Delta g_{t,H} = g_{t+H} - g_t$ from two tokens in the same neural window. Its distance gives equal weight to mean squared differences in the two velocity coordinates and the remaining 30 port coordinates. A softmax of negative distance defines $p^{N,H}$, and confidence-weighted $\mathrm{KL}(p^{B,H} \| p^{N,H})$ is averaged over anchors. The projection and output-head parameters are detached in this transition branch. Transition-only replaces the static relation loss; the combined condition adds this loss to the unchanged LIGHT relation objective.

The conditional variant normalizes each distribution separately within eight groups of eight candidates selected by the static similarity ranks. Group weights multiply normalized static entropy, one minus normalized transition-target entropy, and teacher confidence. Its main term matches the behavioral innovation relations using separately source-scaled current g_t distances; a weaker term matches neural transition differences. This is coarse conditioning on static ranks, not statistical conditional independence. For each horizon, both neural tokens use a common canonical window with the future token at its final position. The current-state term can update P through step 3000, but never the final linear head; its transition term detaches both throughout.

Source-training calibration sets auxiliary coefficients relative to decoding gradients over the read-ins, input/final normalization layers and shared processor, excluding decoder parameters. Per-horizon nominal ratios are 0.05 for transition-only and 0.01 for the static-plus-transition condition. The conditional condition uses 0.015 for each current-state term and 0.005 for each transition term, with an adaptive multiplier updated every 200 steps and bounded to $[0.5, 1.5]$. These are nominal component budgets, not guaranteed realized ratios. Kernel mode selection consulted source validation, whereas scales, temperatures and initial gradient coefficients were fitted on source training. Predictive uncertainty was calibrated on teacher-validation data.

The transition comparisons use seed 11, 99 source sessions, and fixed 20,000-update checkpoints. Listed budgets are nominal auxiliary gradient ratios; the realized conditional-transition ratio exceeded its nominal value late in training. Transition-only guidance improves on Base, while the tested additions do not improve current decoding over BeRG-LIGHT. These settings do not constitute a matched-dose factorial design and do not establish a general failure of dynamic supervision.

Table S5: **Current-velocity decoding with transition-based auxiliary objectives.** Pooled source-validation R^2 is reported for one pretraining seed. State and transition columns give the nominal auxiliary gradient ratios; Δ compares each objective with BeRG-LIGHT. Transition-only guidance improves on Base, while neither tested combination improves on BeRG-LIGHT. The settings do not form a matched-dose factorial comparison.

Objective	State / transition	Pooled $R^2 \uparrow$	Δ vs. LIGHT
Base	0.00 / 0.00	0.8561	−0.0326
BeRG-LIGHT	0.10 / 0.00	**0.8887**	+0.0000
Transition only	0.00 / 0.10	0.8771	−0.0116
BeRG-LIGHT + transition	0.10 / 0.02	0.8877	−0.0010
BeRG-LIGHT + conditional transition	0.10 / 0.04	0.8864	−0.0023

C Predictive Readouts and Representation Analyses

C.1 Frozen behavioral readouts

For each query session, ridge readouts are fitted on source-training representations from the other 98 sessions and evaluated on that session's source-validation data. The backbone has seen training data from every session. The fresh current ridge is distinct from the original trained decoder. Ordinary rows in Table S6 report equal-session means of R^2; conditional rows subtract a history-only reader's R^2 from that of a joint history-plus-hidden reader on matched observations. History comprises 20 velocity bins through the current bin and does not contain the full teacher context. Target-center separations of 100/250 ms place the targets 75/225 ms beyond the complete neural input's right edge. Table S6 reports pretraining seed 11.

At final hidden, the paired LIGHT–Explicit difference is $+0.0992$ for the 250-ms future readout and $+0.0917$ for the corresponding history-conditioned increment. These characterize incremental linear predictability under the specified readers, rather than conditional mutual information or causal effects.

Readouts across neural backbones. Figure 2b uses identical source-training windows and 256 source-validation queries per session across backbones and horizons. The shared training pool contains 698,060 windows, with 698,058 and 698,055 valid targets at 100 and 250 ms, respectively; each fit excludes the query session. Predictors and targets are centered within that fitting set, with an intercept and no feature-wise standardization. Targets retain the frozen normalization fitted to all 99 source-training sessions. The float32 ridge solver uses $\alpha = \max[10^{-3}\,\mathrm{tr}(X_c^\top X_c)/d, 10^{-6}]$, where d is the feature dimension, without validation-based tuning. Features are extracted at the final input-bin center. POYO-Light uses its 128-dimensional output-query representation before Common-Port and the velocity readout, including its learned session-query embedding. These comparisons use seed 11 for window backbones and seed 42 for POYO-Light. The 100/250-ms target centers lie 75/225 ms beyond the full input boundary; both interpolation-support samples also occur after that

Table S6: **Current and future behavioral readout from frozen neural representations.** Values use the fixed historical pretraining seed 11 on source validation. Ordinary rows report session-macro R^2; history-conditioned rows report the ΔR^2 gained by adding neural features to a behavior-history-only readout. Readouts are fitted using other source sessions; all evaluated sessions were present during backbone pretraining. History includes 20 velocity bins through the current bin; future horizons denote target-center separations.

Readout	Base	BeRG-Explicit	BeRG-LIGHT
Final hidden; seed 11			
Fresh current ridge	0.8171	0.8599	**0.8634**
Future 100 ms	0.7036	0.7516	**0.7849**
Future 250 ms	0.3783	0.4898	**0.5891**
History-conditioned 100 ms	0.0381	0.0467	**0.0595**
History-conditioned 250 ms	0.1168	0.2022	**0.2939**
Common-Port; seed 11			
Fresh current ridge	0.8166	0.8600	**0.8633**
Future 100 ms	0.5810	0.6797	**0.7622**
Future 250 ms	0.1788	0.3863	**0.5700**
History-conditioned 100 ms	0.0053	0.0197	**0.0484**
History-conditioned 250 ms	0.0169	0.1286	**0.2753**

boundary. Upstream velocity-filtering support remains unresolved, so these are offline future readouts. Scores weight sessions equally, and all 99 sessions participated in neural pretraining. This protocol differs from native current decoding in both query cohort and aggregation.

C.2 Neighborhood organization

Cross-session velocity prediction uses $k = 20$ nearest neighbors from a bank of 128 evenly indexed training observations from each of the other 98 sessions. Centered cosine subtracts the candidate-bank feature mean from both queries and candidates, then unit-normalizes rows without per-feature variance scaling. Standardized Euclidean distance instead uses the bank's mean and population standard deviations, with scale floor $\max(10^{-5}, 10^{-5} \max_k \sigma_k)$. Both metrics operate in the original feature space without PCA. Predicted velocity is the unweighted mean of the 20 neighbors' velocities. R^2 pools errors and total variation over the two normalized velocity coordinates within session; means weight sessions equally. Panels A/B of Table S7 and panel C all use pretraining seed 11. Explicit performs better at Block 3, whereas LIGHT performs better at final hidden and Common-Port. The large port-neighborhood improvement does not imply an equally large increase in linearly available information or the removal of all session information.

C.3 Neural-activity readouts and input dropout

These evaluations use one pretraining seed and 99 source sessions. For each session and frozen representation, a multi-output log-linear Poisson head is fitted on at most 2,048 source-training windows and evaluated on at most 256 source-validation windows from that session. Windows are sorted by physical time and sampled at evenly spaced indices; training and validation bins are disjoint. Predictor standardization uses training means and population standard deviations, with constant columns removed. For K outputs, fitting minimizes mean Poisson negative log likelihood plus $0.01\|W\|_F^2/(2K)$, with an unpenalized intercept. Float64 L-BFGS-B fits must reach maximum absolute gradient at most 10^{-4} in the original standardized coordinates. Outputs without training spikes retain rate 10^{-6} and remain in scoring. The constant-rate reference uses each output's training mean.

Future counts are predicted separately for each neuron in each of five complete 50-ms bins immediately after the input window (0–250 ms from its right edge). Two windows with incomplete exposure are excluded consistently across conditions, leaving 25,342 validation windows. Reconstruction masks approximately 20% of units throughout the history and fits separate Poisson heads to their endpoint counts; those units were present during backbone pretraining. Three determin-

Table S7: **Behavioral neighborhood organization across neural layers.** Scores are session-macro R^2 for velocity predicted from 20 nearest neighbors drawn from other source sessions. (A,B) The fixed historical pretraining seed 11 under centered cosine and standardized Euclidean distance, respectively. (C) Additional behavioral interfaces evaluated with standardized Euclidean distance and the same pretraining seed. All results use source validation. Comparisons are made within the same layer and distance metric.

Representation	Base	BeRG-Explicit	BeRG-LIGHT
A. Centered cosine; seed 11			
Block 3	0.3698	**0.6079**	0.5988
Final hidden	0.7544	0.8397	**0.8523**
Common-Port	0.2021	0.8329	**0.8491**
B. Standardized Euclidean; seed 11			
Block 3	0.3965	**0.6319**	0.6171
Final hidden	0.8074	0.8416	**0.8554**
Common-Port	0.2358	0.8309	**0.8526**
C. Additional interfaces; seed 11; standardized Euclidean			
Training condition	Block 3	Final hidden	Common-Port
BeRG-MOMENT	0.6240	0.8551	0.8527
LIGHT-FUTURE	0.4662	0.8283	0.8036

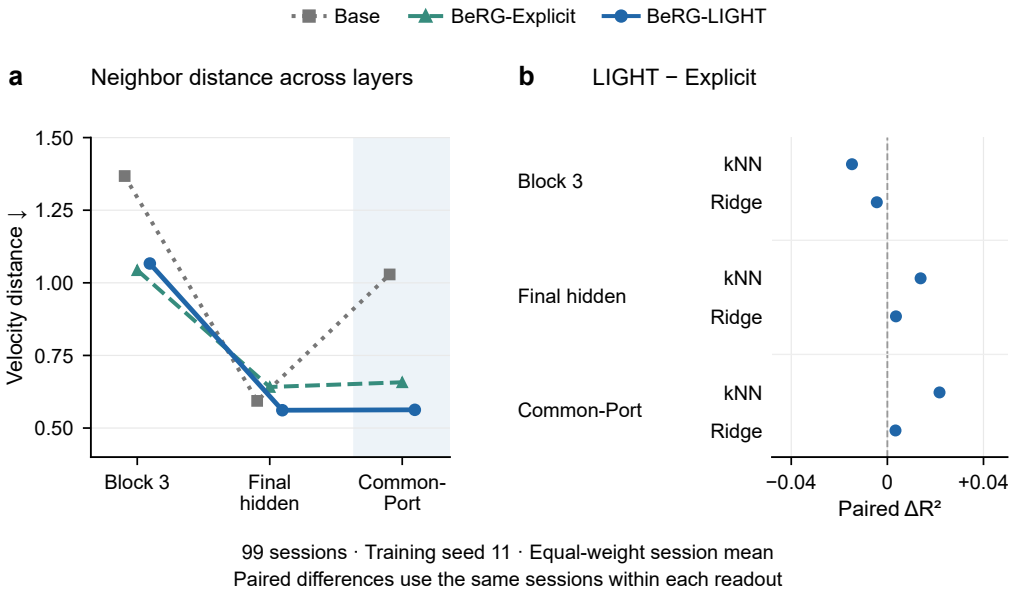

Figure S2: **Layerwise neighborhood distances and learned-encoder effects. (a)** Mean normalized velocity distance to individual cross-session neighbors (lower is better). **(b)** BeRG-LIGHT minus BeRG-Explicit current-velocity R^2, paired within session for each reader. Markers show equal-session means over 99 sessions from pretraining seed 11. kNN uses $k = 20$ and standardized Euclidean distance. kNN and ridge use different query cohorts. All results use validation data from pretraining-exposed sessions.

istic unit permutations per session are shared across models and, for reconstruction, across fitting and evaluation. Dropout fractions are nested within each permutation; mask sizes are rounded to units while retaining at least one visible unit. Bits per spike pool likelihood improvement and spike counts within each session and mask before division by the spike count and $\ln 2$, followed by equal averaging over masks and sessions. Thus five-bin likelihood pooling is distinct from fitting a single summed-count target.

Input dropout fills missing units with zero in source-normalized $\log(1 + x)$ space and evaluates the original decoder without adaptation. Scores average output-wise R^2, then masks and sessions. The three masks are perturbation repeats. At Common-Port, LIGHT–Explicit gains differ across tasks: future counts give -0.0009 for CO and $+0.0258$ for RT; reconstruction gives -0.0045 and $+0.0356$, respectively. Higher masked-input scores alone do not establish a smaller decline from each model's own clean-input score.

Table S8: **Neural-activity readout and robustness to missing neural inputs.** (A) Frozen-representation readouts predict future counts or reconstruct 20% masked units, scored in bits per spike. (B) Behavioral R^2 under neural-input dropout, with changes measured relative to clean input. Results use one pretraining seed and 99 source sessions; perturbation scores average three masks. Higher scores are better. These are custom evaluations, not official Neural Latents Benchmark scores; masks are perturbation repeats, not training seeds.

Endpoint	Base	BeRG-Explicit	BeRG-LIGHT
A. Neural-count readouts; bits per spike (higher is better)			
Future counts / final	0.0477	**0.0884**	0.0808
Future counts / port	0.0513	0.1150	**0.1198**
20% unit reconstruction / final	0.1006	**0.1451**	0.1369
20% unit reconstruction / port	0.0808	0.1347	**0.1387**
B. Neural-input dropout; behavioral readout			
Clean input (R^2)	0.8073	0.8562	**0.8586**
10% missing (R^2)	0.7784	0.8329	**0.8350**
30% missing (R^2)	0.6947	0.7611	**0.7617**
50% missing (R^2)	0.5459	**0.6198**	0.6161
30% missing: Δ vs. clean	-0.1126	$\mathbf{-0.0951}$	-0.0969

D Behavioral Interfaces and POYO-Light Protocols

D.1 Encoder and auxiliary-objective controls

The fixed-versus-learned encoder comparison in Table S2A and the auxiliary controls in Table 2 use the same pretraining seed, 11. Table 2 uses the current 20,000-update recipe with neural seed 11. BeRG-Explicit applies fixed processing to the full LIGHT history/context input. The shuffled condition preserves the candidate set and probability multiset while permuting their correspondence. The shuffled control retains behavior-dependent candidate selection and associated task/session structure. Its score differences from the other conditions are not an additive causal decomposition. Extra decoding adds supervised candidate endpoints and is not an exact computation match. MOMENT uses an adapted pretrained interface with different input packaging from LIGHT.

LIGHT-FUTURE replaces relation matching with direct prediction of the frozen LIGHT teacher's future-trajectory embeddings, retaining current decoding. At 100/250/500 ms, the teacher's future encoder maps the observed displacement-and-velocity prefix to a 64-dimensional target, standardized per horizon and coordinate on valid source-training windows. A head concatenates the endpoint's 32-dimensional port with a fixed 16-dimensional sinusoidal horizon embedding and applies a $48 \to 256 \to 256 \to 64$ MLP (GELU after the first two linear layers; LayerNorm after the first GELU). Smooth L1 loss with $\beta = 1$ averages over coordinates, valid endpoints within session, sessions and available horizons. Each horizon has its own validity mask. The fixed coefficient 3.7916146 is calibrated on four source-training batches to match the reference BeRG shared-gradient

ratio. Auxiliary gradients to P stop after step 3000; decoding continues to update it. The head is discarded at inference.

At 100 ms its final-hidden readout is 0.7880, compared with LIGHT's 0.7849. At 250 ms, LIGHT-FUTURE minus LIGHT is -0.00951, with a descriptive 95% paired-session bootstrap interval of $[-0.02236, 0.00366]$. The interval uses 20,000 draws of 99 sessions with replacement (seed 2026090911), takes the mean paired difference per draw and reports its 2.5th/97.5th percentiles. It describes source-session variation conditional on one trained pair, not training-seed uncertainty. Its current decoding is lower (Table 2). This one setting characterizes an objective tradeoff without establishing a Pareto frontier.

D.2 Behavioral teacher fitting

Teacher forecasting is evaluated at selected checkpoints on the 19 teacher-selection sessions. Each learned configuration uses one selected checkpoint from one teacher-training seed. LIGHT consumes 50×6 history and 50×14 context; MOMENT consumes 48×6 history and the current 14-dimensional context through Fusion. Both learned teachers use future supervision and retrospective context. The frozen MOMENT variant trains Fusion and prediction heads while holding the public backbone fixed; this is the interface used by BeRG-MOMENT. The other variants permit the top two backbone blocks or the full backbone to train. Input differences and checkpoint selection prevent a controlled architecture ranking or an isolated attribution to public pretraining weights.

The four forecast channels comprise two-dimensional displacement relative to the query anchor and two-dimensional velocity, standardized using per-channel means and standard deviations pooled over eligible future targets from the 80 teacher-training sessions (standard deviations are floored at 10^{-4}). Within each session, nMSE averages squared error over evaluated windows, the four standardized channels, and the first 2, 5, or 10 future samples at 50-ms spacing for the 100-, 250-, or 500-ms horizon, respectively; the 19 session scores are then weighted equally. Mean nMSE is the arithmetic mean of these three session-averaged trajectory-prefix scores.

For displacement and velocity at future offset τ, persistence predicts $(0, v_t)$, whereas constant velocity predicts $(\tau v_t, v_t)$. The regularized VAR/AR baseline is implemented as direct multi-output ridge regression from 51 position-and-velocity samples over the preceding 500 ms (including the anchor; 204 inputs) to ten four-channel future samples, without recursive rollout or context inputs. Predictors are standardized on each fitting subset; targets retain the teacher-training normalization. Five-fold session-block cross-validation within the 80 teacher-training sessions selects penalty 0.01 from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. It does not select its penalty on the 19 teacher-selection sessions.

Table S9: **Forecasting performance of behavioral encoders.** Normalized MSE averages future trajectories through 100, 250, and 500 ms on 19 sessions used for teacher selection; Mean nMSE averages these three scores. MOMENT variants differ in which backbone layers are trainable; the frozen-backbone variant supplies the interface used by BeRG-MOMENT. Lower values are better. Input packaging differs between LIGHT and MOMENT. Scores evaluate selected checkpoints and do not isolate the contribution of public pretraining weights.

Behavior model	100 ms	250 ms	500 ms	Mean nMSE↓
Constant velocity	0.1301	0.4206	1.0347	0.5285
Persistence	0.1715	0.5344	1.1011	0.6024
Regularized VAR/AR	0.0357	0.1777	0.4064	0.2066
LIGHT	**0.0164**	**0.0493**	**0.1254**	**0.0637**
MOMENT + Fusion (frozen)	0.0340	0.0710	0.1569	0.0873
MOMENT + Fusion (top 2)	0.0382	0.0742	0.1606	0.0910
MOMENT + Fusion (full)	0.0357	0.0702	0.1543	0.0867

D.3 POYO-Light evaluation and historical training

Evaluation on the shared scoring set. The epoch-60 checkpoints are reevaluated without training or changing weights on the same 8,161 windows and 130,576 scored positions used for the window

backbones. Native event tokenization and the trained unit/session vocabularies are retained. The decoder is queried at 225, 275, . . ., 975 ms within each one-second input window. Targets are copied from the frozen reference export. Predictions are multiplied by their training scale of 20 to restore the original velocity units, then standardized using the shared source-training global means and standard deviations. All legal scoring positions are included, without the historical reach/random-phase evaluation mask or timestamp stitching. The pooled metric is defined in Appendix A.2. Session-macro R^2 improves from 0.8669 to 0.8906, with positive within-session gains in 93 of 99 sessions.

Historical training conditions. POYO-Light uses seed 42 and 99 source sessions. Its epoch-60 checkpoints represent 46,680 training updates, versus 20,000 for the window backbones. The BeRG arm resumes from weights at epochs 35 and 50 without restoring optimizer or RNG states, whereas Base trains continuously. The historical unit-dropout random generator was not explicitly seeded, so the pre-restart prefix is not a strict random-stream match. The common scoring protocol does not match these training trajectories, input representations, training sampling, or computational budgets. These runs do not evaluate an official pretrained POYO checkpoint. Complete epoch-60 training times are unavailable for both arms.

Preserved historical evaluation. Table S10 retains the original evaluation with its separate sampling and output-wise aggregation. The old evaluator uses overlapping windows with a 0.5-second stride, applies the phase mask, and averages predictions at repeated timestamps. Its epoch-60 scores use 394,120 deduplicated queries and average the two output-wise R^2 values. These values are distinct from the unified endpoint in Table 1. The exploratory epoch-35 comparison predates the restarts but is a post-hoc prefix, not a prespecified endpoint.

Table S10: **Historical-protocol current-decoding results with POYO-Light.** These one-seed source-validation results use the original evaluation, distinct from Table 1. Micro scores average output-wise R^2 over pooled query records; macro scores average sessions equally. The epoch-60 comparison has unmatched restart histories: BeRG resumed twice from weights alone, while Base trained continuously. Epoch 35 is an exploratory prefix before the restarts. These runs do not evaluate an official pretrained POYO checkpoint.

Source-validation endpoint	Base	BeRG-LIGHT	ΔR^2
E60 micro	0.9036	0.9157	+0.0121
E60 session macro	0.9047	0.9191	+0.0144
E35 original prefix	0.8800	0.8962	+0.0162

E Information Access and Computational Resources

E.1 Information available to each component

Teacher parameter fitting and relation-bank construction use source-training data; teacher checkpoint selection additionally uses the validation partitions specified in Appendix A.1 and Appendix G. Teachers and banks are removed for neural inference. Perich's retrospective trial/cue context and future teacher targets are training-time privileged information. BeRG-Explicit preserves the complete 1,000-dimensional LIGHT input through fixed centering and block scaling fitted on source-training rows. Although Explicit does not consume future target tensors, it inherits the shared anchor-eligibility rule requiring valid behavior through 500 ms after the anchor. The teacher-selection split and previously evaluated LINK final dates retain the interpretation described above.

The temporal support of dataset-provided Perich velocity preprocessing and the upstream LINK voltage-to-SBP transformation has not been independently established. Interval and identity checks do not by themselves establish end-to-end online causality.

E.2 Recorded computation and reusable teachers

Recorded RTX 4090 wall times are not a controlled throughput benchmark. Perich timers include in-loop diagnostics and checkpoint writes but exclude prior setup and subsequent fixed evaluation.

Table S11: **Information available during teacher fitting, neural pretraining, and target adaptation.** Behavioral teachers and relation banks are used only during training. Perich teacher selection and retrospective context are distinguished from neural pretraining inputs; LINK source, development, and final dates have separate roles. This inventory describes the evaluated offline protocols. Retrospective teacher context is privileged information; the inventory does not establish end-to-end online causality.

Component	Fitting / selection	Information and boundary
Perich LIGHT teacher	80 sessions: official train; 19 others: official validation for selection	50×6 behavior + 50×14 context; future fitting targets
Perich MOMENT interface	Same 80/19 division; Frozen-backbone interface selected at branch update 19K	48×6 behavior + current 14 D context; public backbone frozen
Perich neural model	Official train intervals of all 99 source sessions	Decoding targets and training-only relation bank; source-valid is development
BeRG-Explicit	Source-train centering and block scales; 698,060 rows audited	Fixed 1,000 D history/context features; full reconstruction verified
LINK teacher / bank	94 source dates only; source-valid teacher selection	50 by 6 behavior history; zero context; 1,799,577 bank anchors
LINK target adaptation	17 development dates fix protocol; 45 final dates score	Recent labels only; no target teacher loss or target-validation selection

LINK timers also include periodic source validation and final deployment checks. Runners and hardware contention differ.

Reusable teacher fitting takes 2,504.597 s for Perich LIGHT (17,000 updates), 2,295.593 s for Perich MOMENT's frozen-backbone interface (5,000 warmup plus 20,000 branch updates), and 433.178 s for LINK LIGHT (6,000 updates). These costs are incurred once when a teacher is reused across neural seeds. Public MOMENT pretraining and unrecorded cache preparation are excluded; missing costs are not zero.

Table S12: **Recorded neural-training wall time on an RTX 4090.** Perich uses the fixed historical pretraining seed 11; LINK retains three pretraining seeds. Times are in hours and include the diagnostics and evaluations run inside each training loop; separately prepared teachers and caches are excluded. Runner differences and hardware contention were not controlled. —: no run reported. Teacher fitting and unrecorded cost components are detailed in Appendix E.2.

Dataset / seed	Updates	Base	BeRG-LIGHT	BeRG-Explicit
A. Recorded neural-training wall time; hours				
Perich / 11	20,000	0.650	2.523	2.817
LINK / 11	40,000	0.452	1.113	—
LINK / 22	40,000	0.454	1.088	—
LINK / 33	40,000	0.452	1.106	—
B. Additional Perich neural conditions; seed 11; hours				
Training condition	Updates			Recorded time (h)
BeRG-MOMENT	20,000			1.698
LIGHT-FUTURE	20,000			1.000
Matched shuffle	20,000			2.325
Extra decoding	20,000			1.721

E.3 Additional exposure and calibration costs

Repeated windows do not constitute additional independent recordings, and equal optimizer updates do not match computation. Extra-candidate decoding adds supervision as well as neural forwards. LIGHT-FUTURE's direct latent supervision has no candidate-neural branch. For LINK seeds 22 and 33, median adaptation-plus-scoring time is approximately 9.15–9.28 s per date. The 50 labeled trials last a median 66.78 s, whereas median elapsed time through trial 300 is 403.86 s. Label volume,

acquisition latency, and computation are separate costs. No new inference-latency measurement is implied by removing the teacher.

Table S13: **Training exposure beyond the main neural batches.** Counts distinguish nominal input windows from measured auxiliary exposures and are reported per run. Repeated windows do not represent additional independent recordings. Equal optimizer updates do not imply equal computation or supervision across objectives. Anchor exposures and candidate-window counts are different quantities.

Quantity	Count per run	Interpretation
Perich main windows	2,560,000	Nominal $20{,}000 \times 4 \times 32$; 51,200,000 input bins
Perich relation candidates	256 / active task / update	After update 200; up to two tasks; complete candidate trace unrecorded
Perich extra decoding	8,225,536 extra windows	Measured cumulative counter; additional endpoint labels
Explicit / shuffle anchors	5,043,447 each	Measured seed-11 relation-anchor exposures; not candidate-window count
LINK main windows	1,920,000	Nominal $40{,}000 \times 48$; 96,000,000 input bins
LINK extra candidates	10,240,000 windows	Nominal $40{,}000 \times 256$ additional neural forwards
LIGHT-FUTURE targets	6,388,857	Recorded valid horizon-anchor exposures over 100/250/500 ms

F Figure Construction and Statistical Procedures

F.1 Future Velocity Readouts

Readers and aggregation. Figure 3, Table S6, and Figure S3 evaluate frozen Window Transformer representations from 20,000-update checkpoints on 99 source-validation sessions. For each query session, ridge readers fit source-training observations from the other 98 sessions and evaluate 256 query windows. All 99 sessions contributed to neural pretraining. Velocity coordinates use pooled source-training normalization, including the training partition of the query session; only reader fitting excludes that session. Ordinary current/future ridge readers center predictors and targets on the fitting set without feature-wise standardization and use an unpenalized intercept with $\alpha = \max[10^{-3}\,\mathrm{tr}(X_c^\top X_c)/d, 10^{-6}]$. This is also the ordinary ridge rule used for the cross-backbone comparison in Figure 2b. The history-conditioned readers below instead use support-standardized predictors and a sample-count-scaled penalty. Within each session, R^2 pools squared errors and total variation over both normalized coordinates before division. Each coordinate is centered at its own query-session mean when computing total variation. All Window Transformer readouts use pretraining seed 11. Aggregate curves and bars weight sessions equally; the paired scatter retains all 99 session scores without averaging across training seeds. Task-specific summaries pair each guided model with Base within session before averaging within center-out (78 sessions) or random-target (21 sessions) tasks. No training-seed uncertainty is estimated.

Prediction horizons and conditioning. The 0-ms point uses a newly fitted ridge reader, distinct from the decoder trained with the neural backbone. Horizons denote separations between bin centers. Neural inputs contain twenty 50-ms bins and end 25 ms after the query center; targets at 100 and 250 ms therefore lie 75 and 225 ms beyond the complete input, respectively. Conditional readout measures $R^2(\text{velocity history} + \text{final hidden}) - R^2(\text{velocity history})$. History contains twenty consecutive velocity bins, including the current bin. Both readers use identical support and query observations, predictor standardization fitted only on the support set, and a fixed relative ridge penalty of 10^{-3} (normal-equation penalty $n_{\mathrm{support}}10^{-3}$), with an unpenalized intercept and no additional target standardization. This comparison controls the supplied velocity history, rather than the complete behavioral teacher input. Teacher training included future behavior and offline context; the readout analysis does not establish prediction without future supervision during pretraining.

Velocity-bin error maps. The error maps in Figure S3 use all 25,344 saved query predictions at 250 ms from training seed 11. For each query, we subtract the guided model's Euclidean prediction

error in the two normalized velocity coordinates from Base's error, then average these paired reductions within the true-velocity bin. Positive values indicate lower error than Base. This calculation averages individual error magnitudes, rather than taking the magnitude of an average residual. Both coordinates use boundaries $-\infty, -1.5, -0.5, 0.5, 1.5, +\infty$, producing 25 shared bins. Open outer intervals retain every query. Windows receive equal weight within a bin, so session contributions depend on their bin populations. The bins contain 279–13,042 windows each; equal displayed tile areas represent categorical intervals, not equal populations or physical extents. Both maps use the same linear range, $[-0.90, 0.90]$, without clipping. These maps describe regional error changes from one training seed; bins and windows are not independent training repetitions.

F.2 Representation Organization

Neighborhood atlas. The atlas uses the same 48 moving query anchors for all three models at training seed 11. Eight equal-width direction bins each contribute the six eligible queries with the lowest SHA-256 hashes of their row identities; the selected anchors are then ordered by movement direction. Eligibility requires speed above the pooled training-candidate 20th percentile. For each query session, the candidate bank contains 128 fixed, evenly indexed training observations from each of the other 98 sessions, giving 12,544 candidates. Feature means and population standard deviations are fitted to this other-session bank. Each model retrieves its 20 nearest neighbors by Euclidean distance in the standardized, original 32-dimensional Common-Port space, without PCA. Behavioral labels determine anchor sampling and cell colors but do not enter neighbor ranking. Direction is computed after restoring the original velocity-coordinate scales. The low-speed threshold is approximately 0.5591 in those coordinates; cells at or below it are hatched. Grayscale represents an unwrapped direction angle, whose -180° and $+180^\circ$ endpoints denote the same physical direction.

Unsupervised projection. UMAP uses the same 25,344 Common-Port query states for each model, including all low-speed states. Each model's features are standardized using its fixed training-candidate bank, without session alignment, and projected independently. The common configuration uses two output dimensions, Euclidean distance, 50 neighbors, minimum distance 0.2, spectral initialization, 300 epochs and projection seed 20260911. Behavioral labels are used only for coloring the projection. The fitted projections permit same-session neighbors, whereas the atlas explicitly excludes them. UMAP therefore complements the original-space analysis; orientations, shapes and distances across independently fitted maps are not quantitative comparisons of model quality.

Layerwise readouts and neighbor distances. Quantitative panels use the frozen 20,000-update models from training seed 11. Cross-session kNN readout uses $k = 20$ with standardized Euclidean distance in the original representation space, without PCA. The linear ridge and kNN readers evaluate different query cohorts, despite covering the same 99 sessions; their scores are not paired comparisons between reader types. The neighbor-distance measure averages the Euclidean distance between each query's true normalized two-dimensional velocity and that of each individual neighbor, over 20 neighbors and 256 queries per session. It includes both direction and speed. Aggregate symbols weight all 99 sessions equally. Faint points show individual session scores from seed 11; no training-seed uncertainty is estimated. All panels use validation observations from pretraining-exposed sessions.

F.3 LINK Calibration

Calibration and scoring. The chronological evaluation uses one animal, with 94 source, 17 development and 45 final recording dates. Each final date starts independently from its corresponding 40,000-update source checkpoint. Calibration updates the read-in and output head while keeping the processor and Common-Port frozen. For budget B, calibration uses the most recent B of the first 300 retained trials; evaluation always uses retained trial ordinals 301–375. Nonzero budgets use learning rate 3×10^{-4} and adaptation seed 11. Budgets 0, 10, 50, 100 and 300 use 0, 100, 300, 600 and 600 optimization updates, respectively. Label count and optimization effort therefore vary together across the displayed calibration recipes. Within each date, scoring averages separately

computed R^2 values for the index and middle–ring–small finger-group velocities; date scores are then equally weighted.

Uncertainty and date-level displays. The five-budget curves and heatmap use pretraining seed 11. Curve intervals are 95% percentile intervals from 10,000 non-circular moving-block bootstrap draws, with blocks of five successive observed dates and bootstrap seed 20260906. Absolute-score intervals are method specific; difference intervals resample the date-paired differences. Blocks follow observed-date order, not five-day calendar spans. These intervals describe date variation conditional on one animal and one trained model pair. The separate 0- and 50-trial comparison reports means and sample standard deviations across pretraining seeds 11, 22 and 33; the other budgets do not have three-seed replication. The heatmap retains all 225 date-by-budget differences and uses a common linear scale spanning ± 0.225, without clipping. Date columns are equally spaced chronological categories, and counts indicate the number of dates with positive paired gain. The supplementary 50-trial date plot retains all 135 date-by-seed differences and their 45 within-date means, using actual calendar spacing. Connecting lines do not indicate continual adaptation or evaluation between recorded dates. These repeated observations do not constitute independent animals or additional pretraining repetitions.

F.4 Mechanism illustration

The representation views in Figure 1 use real representations from a frozen 20,000-update window model and its frozen teacher. The 64 matched candidates are selected from a 256-state same-task bank, with at most four candidates per other session; the example contains 46 candidate sessions. The remaining 192 states are gray context only. Teacher features use global source-training centering; student port states use within-session centering over the reconstructed context. Both are unit-normalized. Independent PCA fits on 512 context rows (256 endpoint anchors plus 256 bank states) retain 45.7% and 37.3% of teacher and student variance, respectively. Coordinates are not aligned. Colors and probabilities use the original high-dimensional cosine scores, with teacher temperature 0.110761 and student temperature 0.1. Probabilities retain a common candidate order and exclude gray context states. The plot illustrates one frozen model, not a before/after comparison. Architecture blocks and feature strips are schematic. The center-out and random-target task drawings are conceptual illustrations, not measured trajectories or a specification of apparatus or animal counts. Three input pairs illustrate recording-specific read-ins; they do not imply a simultaneous batch or aligned neurons or timestamps across recordings. Raster thumbnails preserve the original unit/bin counts over twenty 50-ms bins. Ticks are evenly spaced within each bin for display and do not represent measured sub-bin spike times. Sampling-card counts are schematic. The full active bank is forwarded and used for centering; the displayed per-anchor candidate set is not the only forwarded subset.

G Model and Optimization Specifications

G.1 Neural architecture and optimization

The Perich Window Transformer and LINK causal Transformer use bias-free read-ins, six Transformer blocks, width 256, feed-forward width 1024, dropout 0.1, and a 32-dimensional row-orthonormal decoder subspace. Perich uses separate read-ins for each session's recorded units; LINK shares the physical-channel mapping across dates. Each Perich update first averages squared error over valid output elements within each session microbatch, then averages four microbatches of 32 windows each. Each LINK update samples one source date and averages squared error over 48 endpoints and their two output coordinates. These main batches exclude the additional neural forwards used for relation candidates. Table S14 lists the dataset-specific settings.

G.2 Decoder parameterization and relation gradients

Let $A \in \mathbb{R}^{r \times d}$ be the unconstrained projection parameter. Thin QR of $A^\top$ produces $Q \in \mathbb{R}^{d \times r}$ with orthonormal columns. After fixing column signs using the diagonal of the triangular factor, we use $P = Q^\top$. Initialization includes the original decoder's row space and completes the remaining basis directions with seeded orthonormal vectors, preserving the original linear prediction. For original

Table S14: **Settings for the main neural pretraining comparisons.** These settings apply to the Perich Window Transformer and LINK causal Transformer; other neural backbones are specified separately below. The auxiliary schedule is separate from learning-rate warmup. The Perich relation coefficient is inherited and fixed; LINK calibrates its coefficient at each initialization. Additional neural candidate forwards are not included in the main-batch counts.

Setting	Perich Window Transformer	LINK Transformer
Input bins / bin width	20 / 50 ms	50 / 20 ms
Scored positions	Last 16 tokens	Last token
Read-in	Session-specific	Shared channels
Attention	Within-window bidirectional	Causal
Layers / width / heads	6 / 256 / 8	6 / 256 / 4
Feed-forward width / dropout	1024 / 0.1	1024 / 0.1
Subspace / output dimensions	32 / 2	32 / 2
Main batch	4 sessions $\times$ 32 windows	1 date $\times$ 48 windows
Updates	20,000	40,000
Optimizer	AdamW	AdamW
Adam (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.999)$
Peak learning rate	3×10^{-4}	3×10^{-4}
Weight decay / gradient clip	0.01 / 1.0	0.01 / 1.0
LR warmup	1,000 updates	100 updates
Cosine terminal LR	0	10% of peak
Reported pretraining seeds	11	11, 22, 33
Bank rows / active task	256	256
Candidates / anchor	64	64
Candidate cap / session	4	4
Neural temperature	0.1	0.1
Relation coefficient λ	0.0997179114	Initialization-calibrated
Relation activation	Off through 200; ramp to 1,000	Full from first update
Direct relation updates to P	Through step 3,000	Through step 3,000

decoder matrix W_0, this construction sets $C_0 = W_0 P_0^\top$ and satisfies $C_0 P_0 = W_0$ up to numerical precision; the original bias is retained. Base uses the same parameterization and paired initialization.

After step 3000, the relation branch evaluates $z_i^{\mathrm{rel}} = \mathrm{sg}(P)h_i$, whereas decoding always uses $z_i = Ph_i$. Here sg denotes stop-gradient. Thus the projection parameter continues learning through decoding, and relation gradients can still propagate to h_i. We add gradients directly; no gradient-conflict projection is applied. The relation loss has no direct dependence on C or b, and behavioral encoder parameters remain frozen throughout neural training.

G.3 Anchor and candidate sampling

Perich relation anchors are selected from the currently decoded valid tokens, after the first four window positions. Anchor identities are deduplicated by session and timestamp and sampled with speed/direction balancing up to 64 anchors per active session. During cache construction, behavioral samples are aligned to the nearest neural-bin center within 26 ms, discarding rows with conflicting known trial identities and deduplicating neural identities. Decoded anchors subsequently match these cached neural timestamps exactly. Their hidden states retain the original decoding-window context. Candidate windows instead place their last bin's center at the cached candidate timestamp; the bin extends another 25 ms to the right. LINK uses all 48 sampled last-token states as anchors.

Each active task bank contains 256 distinct source-training rows, refreshed with session-balanced quotas and a seeded cyclic shuffle. After excluding the anchor's session, candidate selection scans behavioral-similarity ranks starting at zero-based positions 0, 8, 24, 48, 80, 120, 160, and 208, selecting eight distinct legal rows per scan while enforcing the cap of four per session. Cap-aware spillover and wraparound complete the 64-row selection, spanning at least 16 other sessions. The complete 256-row bank is forwarded and enters within-session neural centering, including rows not selected for a particular anchor's softmax. Anchor states reuse the decoding forward pass, and all bank neural

states retain gradients. The main recipe adds no candidate decoding loss. The teacher temperature is calibrated on a separate fixed source-training ledger to median $\exp[-\sum_j p_{ij} \log p_{ij}] = 8$. This target need not hold for each refreshed training bank. Perich LIGHT uses $\tau_B = 0.1107609734$.

Relation normalization uses $\mathrm{norm}(x) = x/\max(\|x\|_2, 10^{-8})$. Learned behavioral features subtract a fixed mean over aligned source-training representations before normalization. Neural features use the differentiable per-session means of the concatenated anchors and all active-bank rows, as in Eq. 3. Only the relation branch uses these transformations; decoding uses raw z_i. Non-finite or zero neural centered norms raise an error. Learned behavioral rows and final Explicit feature rows with norms at most 10^{-8} also raise an error; these checks do not silently remove rows from the loss.

The KL loss sums over candidates and then averages over valid anchors, as in Eq. 5; it has no additional τ^2 multiplier. The neural temperature is fixed at $\tau_N = 0.1$. All temperatures remain fixed during neural training after source calibration of τ_B.

G.4 Relation weighting

The Window-Transformer recipe fixes $\lambda = 0.0997179114$ across LIGHT, Explicit, MOMENT, and pretraining seeds. It inherits a source-calibrated reference coefficient targeting a relation-to-decoding gradient-norm ratio of 0.1; the current arms are not independently reweighted to maintain that ratio. Its activation multiplier is

$$a(t) = \begin{cases} 0, & t \le 200, \\ (t - 200)/800, & 200 < t < 1000, \\ 1, & t \ge 1000. \end{cases} \tag{8}$$

For each LINK initialization, four source-training batches are evaluated without parameter updates and the coefficient is set to

$$\lambda = \frac{0.1}{\mathrm{median}_{b=1}^{4} \left(\|\nabla_{\theta_{\mathrm{core}}} \mathcal{L}_{\mathrm{rel},b}\|_2 / \|\nabla_{\theta_{\mathrm{core}}} \mathcal{L}_{\mathrm{dec},b}\|_2 \right)}. \tag{9}$$

Calibration disables dropout and computes gradient norms over the Transformer blocks and input/final normalization layers, excluding the read-in and decoder parameters. LINK uses this fixed coefficient with $a(t) = 1$ from the first update. Its 100-update learning-rate warmup does not delay relational guidance.

G.5 Behavioral encoders

Inputs. Each Perich behavioral sample contains two-dimensional position, velocity and acceleration. LIGHT uses 50 samples at 20-ms spacing, from 980 ms before the behavioral anchor through the anchor. Acceleration is a backward difference of the supplied velocity on its original 10-ms grid. The 14 context coordinates comprise task indicators (CO/RT; two), phase indicators (hold/preparation, movement, return; three), phase validity, goal coordinates (two), goal validity, time relative to the go cue, cue-time validity, whether the cue has occurred, trial progress, and progress validity. CO goal coordinates encode the sine and cosine of target direction; RT coordinates encode target position relative to hand position. Kinematic, goal and cue-time features use means and standard deviations from the 80 teacher-training sessions, with standard deviations floored at 10^{-4}. Indicators and trial progress retain their native encoding; invalid context values are zero with validity indicators. Trial progress is $(t - t_{\mathrm{start}})/(t_{\mathrm{end}} - t_{\mathrm{start}})$; phase labels and pre-cue times may use retrospective trial information. This input timing does not establish causal preprocessing of the supplied velocity.

LIGHT. Perich LIGHT uses four temporal layers, width 192, six attention heads, feed-forward width 768, and a 64-dimensional exported state. Its input is 50 six-dimensional kinematic samples at 20-ms spacing plus 50 corresponding 14-dimensional context vectors. Behavior-only fitting combines trajectory forecasting, future-latent prediction, future-trajectory autoencoding, masked-history reconstruction, and state-health regularization. Their full-stage weights are 1, 0.5, 0.25, 0.1, and 0.01, respectively. The first 5000 updates omit the latent-prediction term. The future trajectory grid is 50 ms, with 100/250/500-ms horizons. Training stops at 17,000 total updates, including

the initial 5000-update warmup; checkpoint selection identifies update 9000 using the 19 teacher-selection sessions. The Perich context includes retrospective task information and is training-time privileged information; it is not required at neural inference.

Explicit. BeRG-Explicit retains the same complete normalized history and context input: 300 and 700 coordinates, respectively. Each flattened block is centered with aligned source-training means and divided by its population RMS Euclidean norm before concatenation and unit normalization. These fixed transformations equalize total block energy rather than applying per-feature z-scoring or dimension reduction. Its own behavioral relations determine both candidate ranks and target probabilities. Each block's RMS denominator has a 10^{-8} lower bound.

MOMENT. The MOMENT interface keeps the public MOMENT-1-small backbone frozen while training Fusion and behavioral heads. It uses the last 48 samples of the same six-dimensional history and the current 14-dimensional context. Second- and eighth-layer features are each projected to 64 dimensions and concatenated. Six eight-sample patches per channel yield 36 history tokens with channel, patch, physical-time, dataset and task embeddings. A context token encodes context values and validity; a learned query aggregates the tokens using a two-layer, width-128, four-head Fusion Transformer with feed-forward width 256 and dropout 0.1. Projecting its query output gives the 64-dimensional behavioral state. Fitting comprises a 5000-update warmup and 20,000-update adaptation branch; the selected checkpoint is at branch update 19,000. The whole interface is then frozen for neural pretraining. Its input packaging differs from LIGHT's.

LINK teacher. LINK LIGHT uses 50 six-dimensional behavioral history samples on a nominal 20-ms grid, zero context, and 64-dimensional states. Its channels are two-dimensional position, backward velocity and backward acceleration, with differences divided by actual adjacent timestamp intervals. These historical derivatives differ from the forward-difference decoding label. Predictive supervision uses 100/200/500-ms trajectory prefixes, corresponding to 5/10/25 samples on its native grid. Fitting uses only the 94 source dates, runs for 6000 updates, and selects update 5000 by source-validation forecasting error. Exporting learned current states does not pass future tensors into the history encoder, but the encoder weights have received future supervision.

Behavioral fitting losses. For Perich, $Y^{(h)}$ contains normalized displacement relative to the anchor and future velocity over the horizon's trajectory prefix, and $u^{(h)}$ is its encoding by a jointly trained future encoder. Trajectory forecasting applies Huber loss ($\delta = 1$) to a prediction of $Y^{(h)}$ from the current behavioral state. Future-latent prediction minimizes diagonal-Gaussian negative log likelihood of $\mathrm{sg}(u^{(h)})$ under the predicted mean and standard deviation, with log standard deviations clipped to $[-5, 3]$. Future reconstruction applies Huber loss from a decoder of $u^{(h)}$ to $Y^{(h)}$, updating the future encoder. Losses average samples and coordinates within each prefix, then horizons equally. LIGHT's separate masked-history pass replaces two randomly positioned three-sample spans (overlap allowed) with learned mask tokens; Huber loss is scored only on masked positions and coordinates. Context remains available subject to context dropout.

For current unnormalized behavioral states $s_i \in \mathbb{R}^d$, centered batch matrix S_c and $C = S_c^\top S_c / \max(1, n-1)$, state-health regularization is

$$\mathcal{L}_{\text{health}} = \frac{1}{d}\sum_k \max\left(0, 1 - \sqrt{\frac{1}{n}\sum_i (s_{ik} - \bar{s}_k)^2 + 10^{-4}}\right) + \frac{0.01}{d}\sum_{k\neq l} C_{kl}^2.$$

MOMENT uses LIGHT's trajectory, latent, reconstruction and health terms with weights $1, 0.5, 0.25, 0.01$, without masked-history loss, throughout both fitting stages. LINK uses all five LIGHT terms and weights, omitting the latent term during its first 300 updates. LINK future targets contain absolute position and backward velocity, normalized by source-training history statistics, rather than Perich's anchor-relative displacements.

Behavioral fitting optimization. Teachers use AdamW with peak learning rate 3×10^{-4}, betas $(0.9, 0.95)$, weight decay 0.05 and gradient-norm clipping at 1. Perich LIGHT uses batches of 512, 1,500 learning-rate warmup updates and cosine decay toward 10% of peak over its planned 30,000-update budget. The objective transition after step 5000 does not restart this schedule; actual fitting

stops at 17,000 updates, with update 9000 selected. Frozen-backbone MOMENT uses batches of 512 and restarts the optimizer and schedule for its 5,000-update warmup and 20,000-update branch. Learning-rate warmup lasts 250 and 1,000 steps respectively, followed by cosine decay to 10% of peak. LINK LIGHT uses batches of 128, 150 learning-rate warmup updates and cosine decay to 10% of peak at step 6000. These schedules are separate from neural learning-rate warmup and relation activation.

G.6 SUPPORTING COMPARISONS

Causal-Transformer and temporal-convolution comparisons use seed 11, with source-validation and historical same-session test results reported separately. The convolutional processor has 24 layers, width 136, and kernel size 5, with the same read-in and decoder framework. Current encoder/objective controls also use seed 11. Monkey T/MC-Maze use pretraining seed 11 and support/adaptation seed 42. POYO-Light uses seed 42 and the common scoring protocol, with its historical training conditions detailed in Appendix D.3.

G.7 BIGRU AND WINDOW MLP

Both backbones use the 20-bin neural-window interface described in Appendix A.2. Session-specific bias-free read-ins project inputs to 256 dimensions, followed by shared layer normalization. The BiGRU has two bidirectional GRU layers with 256 hidden units per direction and dropout of 0.1 between layers. A linear map projects the concatenated 512-dimensional outputs to 256 dimensions. Recurrent states reset to zero for each window. The window MLP flattens the 20×256 inputs in temporal order and applies linear maps of sizes $5120 \to 512 \to 512 \to 5120$. Layer normalization, GELU, and dropout of 0.1 follow each of its first two maps. Outputs are reshaped to 20×256. Both models apply final per-token layer normalization, the 32-dimensional decoding subspace, and the two-dimensional velocity readout. Including read-ins and the decoding interface, BiGRU and MLP contain 4,602,434 and 8,012,098 parameters.

Both architectures train from scratch for 20,000 updates with seed 11. Within each architecture, Base and LIGHT share initial weights, main-data sampling, and dropout streams. Each update samples 32 windows from each of four sessions. AdamW uses learning rate 3×10^{-4}, $\beta = (0.9, 0.95)$, weight decay 0.01, gradient clipping at 1.0, and 1,000 warmup updates followed by cosine decay. The frozen LIGHT encoder and source-training relation bank are reused. Four fixed training batches calibrate the relation-to-decoding shared-core gradient-norm ratio to 0.1, giving fixed coefficients $\lambda = 0.10187969$ for BiGRU and 0.04026179 for MLP. The relation schedule follows Eq. 8; direct relational updates to the projection stop after step 3,000, while decoding continues to train it. Final checkpoints are evaluated with stride 20 and at most 256 source-validation windows per session.

H SUPPLEMENTARY VISUAL ANALYSES

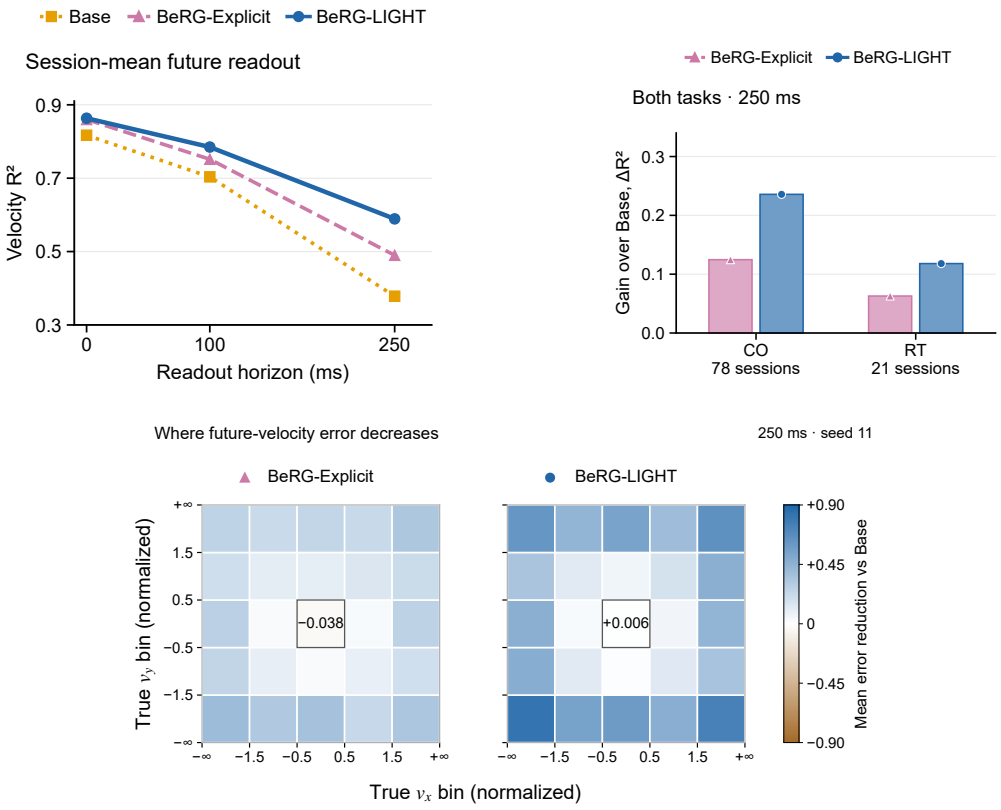

Figure S3: **Additional future-readout analyses.** Top left: ridge readout across horizons, equally averaged over all 99 sessions. Top right: paired 250-ms R^2 gains over Base in CO and RT. Both use pretraining seed 11 and weight sessions equally. Bottom: mean paired reduction in Euclidean prediction error versus Base within true-velocity bins at 250 ms (seed 11, all 25,344 queries); positive values indicate improvement. Both maps share one linear scale. Equal tile areas do not represent equal sample counts. All results use validation data from pretraining-exposed sessions.

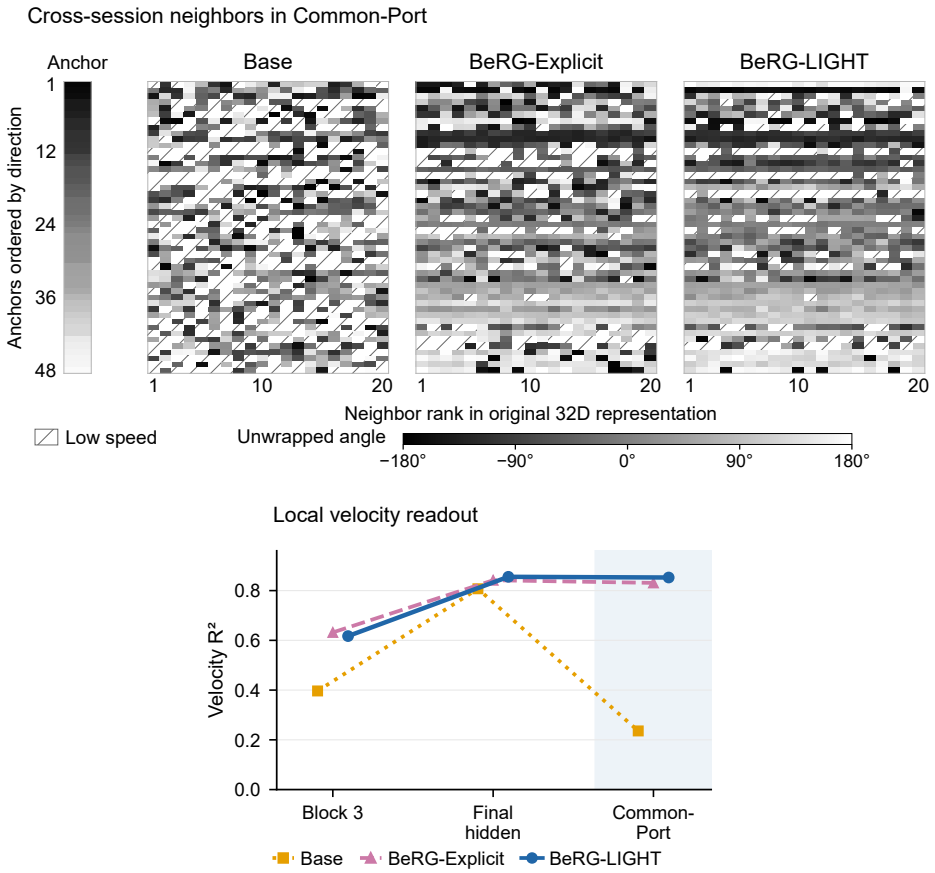

Figure S4: **Cross-session neighborhood atlas and local velocity readout.** Top: the same 48 anchors are ordered by true movement direction. Each row shows the 20 nearest states from other sessions in original 32-dimensional Common-Port space. Grayscale encodes direction and hatching denotes low speed; $-180°$ and $+180°$ represent the same direction. This atlas uses training seed 11. Bottom: cross-session kNN velocity readout ($k = 20$). Markers show equal-session means over all 99 sessions from the same pretraining seed 11. Results use validation data from pretraining-exposed sessions.

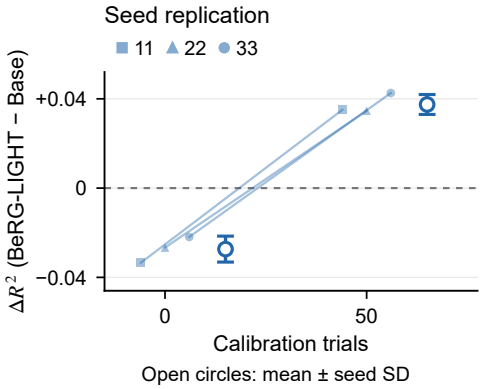

Figure S5: **Calibration gains across pretraining seeds.** Faint symbols show date-mean BeRG-LIGHT minus Base gains for three pretraining seeds at 0 and 50 calibration trials; open circles show their mean ± sample SD. Each seed averages the same 45 held-out dates from one animal. Horizontal offsets separate seeds and summary points.