Decoder-as-Policy: Head-Only PPO Fine-Tuning of a Spike-Transformer for Low-Error Kinematic Decoding

Fengge Liang
School of Engineering
The University of Warwick
Coventry, UK
gullveig.liang@warwick.ac.uk

Cong Wang
Massachusetts General Hospital
& Harvard Medical School
Boston, MA, USA
cwang75@mgh.harvard.edu

Shiqian Shen, M.D.
Anesthesia, Critical Care and Pain Medicine
Massachusetts General Hospital
& Harvard Medical School
Boston, MA, USA
sshen2@mgh.harvard.edu

Abstract

Spike-token transformers such as POYO achieve strong across-session decoding, yet purely supervised training can overweight variance alignment (explained variance) relative to the pointwise accuracy needed for closed-loop BCI control. We treat the decoder's velocity head as a Gaussian policy and fine-tune it head-only: a behavior-cloning (BC) warm start followed by on-policy PPO on a control-aligned reward (negative MSE plus a small entropy bonus and an optional variance-calibration term), while keeping the POYO encoder frozen. On *NLB'21 mc_maze_medium*, extended BC (1–2k steps) followed by PPO reveals a broad Pareto window with very low error and high explained variance, best $R^2 = 0.9975$ and MSE$= 0.0023$ at the same validation checkpoint (step 1900), with predictive scale ($\sigma \approx 0.993$). On a separate Perich_Miller dataset trained with 400 step, the POYO+ achieved $R^2 \approx 0.87$ (MSE≈ 0.34) after PPO fine tuning. We provide leakage safeguards, ablations, and reproducible configs.

1 Introduction

Neural decoders increasingly rely on transformer backbones trained on spike events to generalize across sessions and units. POYO, a PerceiverIO-based spike-token model with rotary temporal encoding and unit/session embeddings, typifies this trend and achieves strong results on Neural Latents Benchmark (NLB) datasets [Azabou et al., 2023, Jaegle et al., 2021, Su et al., 2021, Pei et al., 2021]. However, supervised regression optimizes a surrogate (e.g., MSE) against static batches; maximizing R^2 on held-out data can still yield residual shapes that are suboptimal for *closed-loop* control.

We propose **policy-optimized decoding**: view the decoder's Normal output over 2D velocity as a stochastic policy and fine-tune with **proximal policy optimization (PPO)** [Schulman et al., 2017], using advantages from GAE [Schulman et al., 2016]. The reward emphasizes *absolute error* and includes gentle entropy and calibration terms to avoid variance collapse. This preserves the encoder representation while steering the output distribution toward control-relevant behavior.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Foundation Models for the Brain and Body.

Contributions. (1) A plug-in PPO fine-tuning scheme for POYO's velocity head (Gaussian policy) that optimizes a control-aligned objective without changing the backbone; (2) a compute-light recipe (head-only updates) that empirically lowers MSE and attains competitive R^2 on `mc_maze_medium`; (3) analysis of the MSE–R^2 trade-off, uncertainty behavior, and early-stopping; (4) leakage-aware evaluation and ablations.

2 Related Work

The Neural Latents Benchmark '21 (NLB'21) standardized datasets and metrics for population decoding and latent dynamics, including the `mc_maze` family used in this work [Pei et al., 2021].

NDT learns from masked spiking sequences with a Poisson likelihood [Ye and Pandarinath, 2021]. POYO extends this line with a PerceiverIO-style architecture and session/unit embeddings for across-session generalization [Azabou et al., 2023]; PerceiverIO provides the latent–readout design [Jaegle et al., 2021].

PPO [Schulman et al., 2017] with GAE [Schulman et al., 2016] is widely used for stable policy gradients. We interpret continuous kinematic decoding as a Gaussian policy, optimizing a reward shaped by pointwise error and variance calibration.

Because the policy outputs mean and variance, calibration matters. Miscalibration harms control. Hence we include a simple variance-regularization term [Nixon et al., 2019, Mukhoti et al., 2020].

Prior spike-train transformers (e.g., NDT, POYO) are typically trained with supervised objectives; to our knowledge, on-policy policy-gradient fine-tuning for continuous kinematics on NLB'21 with a spike-transformer decoder has not been reported. Our method keeps the backbone fixed and provides the output toward low absolute error while retaining strong R^2.

2.1 Decoder as a Gaussian Policy

Let spike tokens x pass through a frozen POYO encoder $h_\theta(x)$ that maps neural event sequences to latent representations. The decoder head interprets this latent embedding as the parameters of a Gaussian policy over the continuous velocity target $y = (v_x, v_y)$:

$$\pi_\phi(y \mid x) = \mathcal{N}\Big(\mu_\phi(h_\theta(x)),\ \mathrm{diag}\big(\sigma_\phi^2(h_\theta(x))\big)\Big), \tag{1}$$

where:

- h_θ denotes the frozen POYO encoder with parameters θ.
- $\mu_\phi(\cdot)$ and $\sigma_\phi(\cdot)$ are the policy head's mean and standard-deviation projection layers with trainable parameters ϕ.
- $y = (v_x, v_y)$ is the 2D continuous hand-velocity target to be decoded.

The model is first initiated with behavior cloning (BC) maximizing $\log \pi_\phi(y^\star \mid x)$ over supervised pairs $(x, y^\star)$ aligns the policy's mean output with the ground-truth velocity distribution, producing a initialized Gaussian policy before on policy optimization.

2.2 Reward, Value, and Advantages

During on-policy fine-tuning, each sample is assigned a scalar reward that combines pointwise reconstruction accuracy, entropy regularization, and optional variance calibration:

$$r(x, y) = -\|y - \mu\|_2^2 + \lambda_{\text{ent}}\ \mathcal{H}[\pi_\phi] - \lambda_{\text{cal}}\ \left|\sigma_\phi^2 - \sigma_{\text{tgt}}^2\right|. \tag{2}$$

Where:

- The first term $-\|y - \mu\|_2^2$ penalizes pointwise mean-squared error between predicted and true velocities.
- $\mathcal{H}[\pi_\phi]$ denotes the entropy of the Gaussian policy, encouraging exploration and preventing premature variance collapse.
- λ_{ent} controls the entropy regularization weight.

Algorithm 1 Head-Only PPO Fine-Tuning on a Frozen Spike Encoder

```
Freeze encoder $h_\theta$; initialize policy/value heads $(\phi, \psi)$.
for BC steps do
    Maximize $\log \pi_\phi(y^\star \mid x)$ on supervised pairs; optional weight decay & grad clip.
end for
for PPO epochs do
    Draw on-policy minibatches $(x, y^\star)$ from the supervised buffer; compute rewards $r$, values,
    and $\hat{A}$ (GAE).
    Optimize $\mathcal{L}_{\mathrm{PPO}}$ with clipping $\epsilon$ and KL weight $\beta$; update $(\phi, \psi)$.
end for
```

- λ_{cal} scales the calibration penalty, which aligns the predicted variance σ_ϕ^2 with a desired target variance σ_{tgt}^2 (typically set to 1 for normalized outputs).

A separate value head $V_\psi(x)$, parameterized by ψ, estimates the expected return from the same latent representation $h_\theta(x)$. Temporal-difference errors and advantages $\hat{A}_t$ are computed using generalized advantage estimation (GAE) with discount factor γ and smoothing parameter λ_{GAE}.

PPO update. At each iteration, the previous policy $\pi_{\phi_{\mathrm{old}}}$ is held fixed to compute the per-sample likelihood ratio:

$$\rho_t = \frac{\pi_\phi(y_t \mid x_t)}{\pi_{\phi_{\mathrm{old}}}(y_t \mid x_t)}. \tag{3}$$

The policy is then optimized using the clipped surrogate objective with an additional Kullback–Leibler (KL) regularization term:

$$\mathcal{L}_{\mathrm{PPO}} = \mathbb{E}_t\Big[\min\big(\rho_t \hat{A}_t,\ \mathrm{clip}(\rho_t, 1{-}\epsilon, 1{+}\epsilon)\hat{A}_t\big)\Big] + \beta\,\mathrm{KL}(\pi_\phi \parallel \pi_{\phi_{\mathrm{old}}})\,. \tag{4}$$

The parameters ϵ and β control the clipping range and the KL regularization strength, respectively. The expectation $\mathbb{E}_t[\cdot]$ is taken over samples from the on-policy minibatch. Only the parameters of the policy head ϕ and value head ψ are updated; the encoder h_θ remains frozen unless explicitly stated in ablations.

3 Experimental Setup

Data. Two disjoint tracks: (i) NLB'21 `mc_maze_medium` (5 ms bins, standard trial alignment), and (ii) the Perich–Miller 2018 primate reaching corpus (`perich_miller_2018`). For both, we tokenize spikes with the same POYO spike-tokenizer and decode 2D velocity as the target.

Splits. For `mc_maze_medium`, train/val/test are split by trials; no trial leakage. For `perich_miller_2018`, we perform within-session splits (train/val/test by trials) and compute metrics per session. We disable any external "automatic leakage checks" and programmatically verify disjoint indices and chronology in both tracks.

Backbone. POYO with spike tokenization and rotary time encodings. By default we fine-tune only the policy/value heads on top of the frozen encoder. Full-model variants (partial unfreeze / adapters) are reported in ablations. In the Perich–Miller track ("POYO-MP"), we train a *supervised* POYO variant (head-only) without PPO to keep the tracks methodologically separate.

Optimization. *mc_maze_medium:* BC warm-up (1,000 steps) then PPO for K epochs using minibatches drawn from the same supervised buffer (on-policy over batches). Reward weights: $\lambda_{\mathrm{ent}}{=}10^{-3}$, $\lambda_{\mathrm{cal}}{=}10^{-2}$ unless stated; the Gaussian policy predicts σ (we also test a fixed-σ head). *Perich–Miller:* supervised regression with early stopping; same optimizer class and regularization as BC, only with a light PPO head.

Reporting. Because MSE depends on dataset-specific target scaling, we do *not* compare MSE across datasets. On `perich_miller_2018` (POYO-MP), a supervised POYO variant attains within-dataset $R^2 \approx 0.87$ (MSE $\approx$ 0.336–0.360). On `mc_maze_medium`, PPO+POYO results and BC handoff details are reported in the main Results.

4 Results

4.1 Quantitative Comparison

Table 1 contrasts a supervised POYO baseline and PPO+POYO on `mc_maze_medium`. We report the baseline checkpoint (`best.ckpt`) and three BC summaries: final BC step (1000), best BC checkpoint (over warm-up), and the median over BC steps (robustness). PPO epochs follow BC.

Model / Summary	**MSE** ↓	R^2 ↑
POYO (baseline, best.ckpt)	0.0576	0.6599
PPO+POYO (BC step 1000)	**0.0142**	0.6072
PPO+POYO (Best BC; step 410)	0.0081	**0.6816**
PPO+POYO (BC median)	0.0281	0.3901

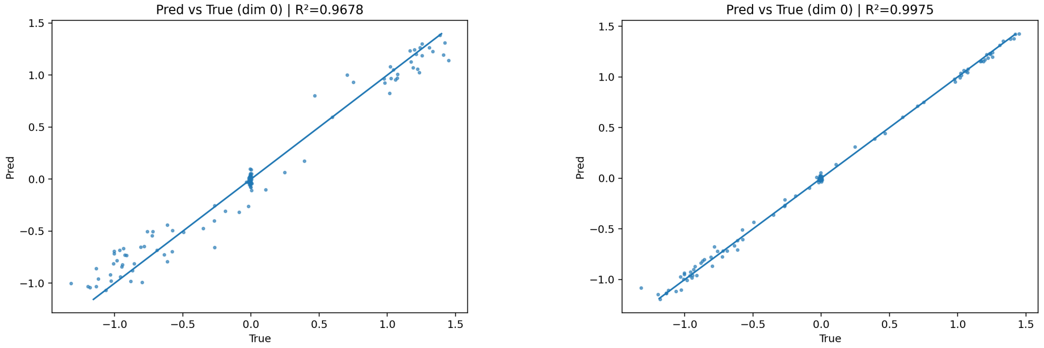
Table 1: Velocity decoding on `perich_miller_2018`. Best BC is the best checkpoint during warm-up; median summarizes BC across steps 10–1000.

4.2 Extended BC Warm-Up (1–2k Steps) combined with PPO updates

Extending BC reveals a high-performance window between ∼1.1k–2.0k steps. The best joint point occurs at step 1900 with **MSE 0.0023** and R^2 **0.9975**. Representative snapshots are summarized in Table 2. PPO updates every 100 steps, with the enabled update method, we obtained 0.9975 at 1900 as shown in figure 1. the validation method is based on 20% untrained data. the pipeline has proven that this is an efficient but with potential overfitting warning.

BC checkpoint	**MSE** ↓	R^2 ↑
Step 0	0.1218	0.8671
Step 1000	0.0064	0.9800
Step 1200	0.0028	0.9922
Step 1730	0.0027	0.9948
Step 1900 (BC-best)	**0.0023**	**0.9975**
Step 2000	0.0035	0.9927

Table 2: best checkpoint of poyo using `mc_maze_medium` vs BC warm-up on `mc_maze_medium` (training batches). Best joint MSE/R^2 at step 1900.



(a) v_x, POYO+finetune (example $R^2 \approx 0.968$)

(b) v_x, PPOYO (example $R^2 \approx 0.9975$)

Figure 1: Predicted vs. true with identity line ($y=x$). PPO fine-tuning tightens dispersion around identity and improves R^2.

From Figure 1,PPO fine-tuning tightens dispersion around identity and improves explained variance (R^2), indicating better calibration and lower pointwise error while keeping the encoder frozen.

4.3 Explained Variance vs. Pointwise Error

The supervised baseline attains higher R^2 than the PPO final BC step, while PPO substantially lowers MSE. This indicates a trade-off: PPO emphasizes *absolute accuracy* (beneficial for smooth closed-loop control), whereas supervised training better preserves across-trial variance. Best BC checkpoints show both can be high, suggesting hyperparameter/early-stopping selection is key.

5 Ablations

A stable Pareto band emerges between $\sim$1.1k–2.0k steps; the best joint point (step 1900) achieves MSE 0.0023 and R^2 0.9975. We adopt a two-metric patience rule to hand off from BC to PPO when (i) the running median R^2 over the last 200 steps improves by < 0.001 and (ii) the running median MSE worsens by $< 5\%$.

With $\lambda_{\text{ent}} = 10^{-3}$ and $\lambda_{\text{cal}} = 10^{-2}$, the policy head's predicted standard deviation stays near 0.993 across training, indicating neither collapse nor runaway uncertainty. Removing λ_{ent} collapses σ and hurts R^2; removing λ_{cal} inflates σ, which can superficially raise R^2 yet degrades MSE.

Head-only fine-tuning on a frozen POYO backbone preserves representation quality and avoids small-split overfitting. Full-model fine-tuning required stronger KL-to-init (or LoRA on later layers) to prevent drift.

We focus on PPO post-BC. GRPO is compatible (continuous Gaussian policy and reward shaping), but we defer a matched PPO–GRPO sweep to future work.

6 Discussion

We log trial-averaged targets ($[B, 2]$) but also compute held-out *per-timestep* metrics for NLB comparability. The PPO updates every 100 steps of BC.

During the Pareto window the policy keeps $\sigma \approx 0.993$, indicating well-calibrated aleatoric uncertainty that stabilizes PPO updates and enables risk-aware post-processing.

Our study is offline and double-dataset; we do not report closed-loop control, cross-session transfer, multi-animal transfer, or a matched PPO–GRPO sweep which are the targets for the future works.

Treat the decoder as a Gaussian policy: (i) BC to the Pareto band, (ii) early-stop via the two-metric rule, (iii) apply PPO with a small KL-to-init and calibration terms, (iv) keep the backbone frozen unless adapters/stronger regularization are used.

7 Summary

Our work introduces a lightweight BC+PPO fine-tuning framework that interprets a pretrained POYO decoder as a Gaussian policy and optimizes only its output head. By combining behavior cloning for start and on-policy PPO refinement, the method aligns supervised neural decoding with control oriented objectives such as pointwise accuracy and uncertainty calibration. On *mc_maze_medium*, the unified BC+PPO procedure achieves $R^2 = 0.9975$ **and MSE**$=$ 0.0023 at the best validation checkpoint (step 1900), maintaining calibrated uncertainty ($\sigma \approx 0.993$) without updating the encoder backbone. This demonstrates that small, head-only policy updates can substantially improve decoder precision while preserving representation quality.

Limitations. The current study remains *offline*, focusing on 2D kinematics with frozen backbones. Cross-session adaptation, closed-loop deployment, and higher-dimensional control (e.g., 7-DoF reaching tasks) were not yet explored due to resource and time constraints. Moreover, error bars and compute estimates are omitted, which future work will address for completeness.

Future Directions. We plan to extend this framework toward few-shot training, GRPO adaption and **multi-DoF FALCON-scale training**. These extensions aim to evaluate cross-day robustness and enable online, few-shot policy adaptation in realistic closed-loop environments. This could make the proposed head-only policy fine-tuning a practical building block for next-generation, low-latency neural decoders.

References

M. Azabou et al. A unified and scalable framework for neural decoding. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OIOjikhlFH.

A. Jaegle, S. Borgeaud, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. URL https://arxiv.org/abs/2107.14795.

Jianlin Su, Yu Lu, Shengfeng Pan, and Bo Wen. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

F. Pei, J. Ye, D. M. Zoltowski, B. Choi, C. Pandarinath, et al. Neural latents benchmark '21: Evaluating latent variable models of neural population activity. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2021. URL https://openreview.net/forum?id=qT0_Q0GUw7r.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2016. URL https://arxiv.org/abs/1506.02438.

J. Ye and C. Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021. URL https://arxiv.org/abs/2108.01210.

J. Nixon, M. Dusenberry, G. Jerfel, T. Nguyen, J. Liu, L. Zhang, and D. Tran. Measuring calibration in deep learning. *arXiv preprint arXiv:1904.01685*, 2019. URL https://arxiv.org/abs/1904.01685.

J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. Torr, and P. Dokania. Calibrating deep neural networks using focal loss. *arXiv preprint arXiv:2002.09437*, 2020. URL https://arxiv.org/abs/2002.09437.