WHEN SENSORS FAIL: TEMPORAL SEQUENCE MODELS FOR ROBUST PPO UNDER SENSOR DRIFT

Kevin Vogt-Lowell, Theodoros Tsiligkaridis, Rodney Lafuente-Mercado, Surabhi Ghatti
MIT Lincoln Laboratory

Shanghua Gao, Marinka Zitnik
Harvard

Daniela Rus
MIT

ABSTRACT

Real-world reinforcement learning systems must operate under distributional drift in their observation streams, yet most policy architectures implicitly assume fully observed and noise-free states. We study robustness of Proximal Policy Optimization (PPO) under temporally persistent sensor failures that induce partial observability and representation shift. To respond to this drift, we augment PPO with temporal sequence models, including Transformers and State Space Models (SSMs), to enable policies to infer missing information from history and maintain performance. Under a stochastic sensor failure process, we prove a high-probability bound on infinite-horizon reward degradation that quantifies how robustness depends on policy smoothness and failure persistence. Empirically, on MuJoCo continuous-control benchmarks with severe sensor dropout, we show Transformer-based sequence policies substantially outperform MLP, RNN, and SSM baselines in robustness, maintaining high returns even when large fractions of sensors are unavailable. These results demonstrate that temporal sequence reasoning provides a principled and practical mechanism for reliable operation under observation drift caused by sensor unreliability.

1 INTRODUCTION

Real-world reinforcement learning (RL) systems, from robotic control to autonomous driving, depend on sensor feedback that is often unreliable. Failures, communication dropouts, or transient corruption lead to partial observability and degraded performance (Jaakkola et al., 1994; Kaelbling et al., 1998). Standard RL agents, especially those based on multilayer perceptrons (MLPs), assume fully observed states and thus suffer sharp reward losses when inputs become unreliable (Morad et al., 2023; Pleines et al., 2023).

In practical systems, sensor outages exhibit both temporal persistence and correlations between related components (Vuran et al., 2004; Das et al., 2016). To capture these effects during analysis and evaluation, we model sensor reliability using a standard stochastic failure process that accounts for individual- and group-level dependencies. This framework allows for a systematic study of robustness without introducing new assumptions about failure dynamics.

Building on this setting, our focus is on robust sequence-based PPO agents that can leverage temporal structure to infer missing information and maintain performance under partial observability. Our key contributions are as follows.

- *Sequence-Model PPO Architectures.* We integrate Transformer- and State Space–based encoders into PPO, enabling agents to exploit temporal dependencies to enhance decision-making.
- *Theoretical Robustness Analysis.* We derive a high-probability bound on infinite-horizon reward degradation under stochastic observation failures, quantifying how robustness scales with policy smoothness and failure persistence.

- *Empirical Evaluation.* We show that Transformer PPO agents achieve substantially higher reward retention under severe sensor dropout compared to MLP, RNN, and SSM baselines across MuJoCo tasks.

Together, these results establish sequence modeling as a key mechanism for robustness in online RL, demonstrating how temporal reasoning mitigates the brittleness of standard policy architectures in realistic, unreliable environments.

2 Related Work

Partial Observability in RL In the context of addressing partial observability in reinforcement learning, early efforts focused on adding memory through recurrence to value-based agents. Deep Recurrent Q-Networks (DRQN) (Hausknecht & Stone, 2015) have been proposed which replace DQN's first fully-connected layer with an LSTM to integrate information over time from single-frame inputs. DRQN not only matches DQN's performance on fully observable Atari games but also degrades more gracefully when evaluated under partially observed conditions. Zambaldi et al. (2019) introduced a relational module within a model-free RL agent that encodes observations as sets of entity vectors and applies iterative message-passing (self-attention) to reason over object relations, yielding substantial gains in sample efficiency, generalization to novel scenarios, and interpretability on tasks like StarCraft II and Box-World. RLBenchNet (Smirnov & Gu, 2025) provides an empirical comparison of sequence models under PPO on control and memory-based environments. However, this work is purely empirical and does not offer a theoretical characterization of performance degradation under partial observability. Moreover, the masking mechanisms used in their partially observable settings are simplistic (e.g., permanently removing velocity components or shrinking observation windows) and do not model realistic sensor failures with temporal persistence, correlation, or distributional drift. As a result, these benchmarks do not capture the structured, time-correlated observation failures encountered in real-world systems.

Transformer Models The Transformer architecture, introduced by Vaswani et al. (2017), revolutionized sequence modeling by relying entirely on self-attention mechanisms, dispensing with recurrence and convolution. Its ability to model long-range dependencies with parallelizable computation has made it the foundation of numerous advances in natural language processing and beyond (Devlin et al., 2019; Brown et al., 2020). Modified Transformers, such as UniTS (Gao et al., 2024) and Transformer-XL (Dai et al., 2019), have pushed Transformers even further by enabling universal time series representations and dependency learning beyond fixed-size context windows, respectively.

State Space Models Recent advances in sequence modeling have positioned structured state space models (SSMs) as an alternative to Transformer attention for capturing long-range dependencies with favorable scaling. S4 introduced the "structured state space sequence" formulation, showing that carefully parameterized continuous-time linear systems can be made computationally efficient while retaining strong, long-context performance (Gu et al., 2022). Building toward simpler and more scalable SSM/RNN hybrids, Linear Recurrent Units (LRUs) "resurrect" efficient recurrent-style sequence modeling by using a structured (diagonal-style) recurrence that supports fast parallel training while maintaining strong long-range capability (Orvieto et al., 2023a). Mamba further advanced this line by introducing selective (input-dependent) state space updates, enabling token-wise, content-adaptive computation while preserving linear-time scaling in recurrent mode (Gu & Dao, 2024). Most recently, LinOSS (Rusch & Rus, 2025) proposed an oscillatory SSM derived from stable discretizations of forced second-order (harmonic oscillator) dynamics, yielding stable long-horizon behavior under a particularly simple condition—a nonnegative diagonal state matrix—and integrating efficiently via associative parallel scans.

3 Sensor Failure Model

We model correlated observation failures using a two-layer Markov process that captures both individual and group-level reliability dynamics. Each sensor follows a binary Markov chain for local hardware reliability, while sensor groups share a higher-level process representing subsystem dependencies (e.g., shared communication buses or power lines). This structure captures temporal

persistence and spatial correlation while remaining analytically tractable with closed-form steady-state probabilities (Huang & Dey, 2007; Liu et al., 2006; Mo et al., 2013).

For sensor i, let $z_i(t) \in \{0, 1\}$ denote its operational status, evolving as a two-state Markov chain with failure probability p_{fail} and recovery probability p_{recover}. The steady-state probability of operation is $\pi_z = p_{\text{recover}}/(p_{\text{fail}} + p_{\text{recover}})$. Each group j has a binary variable $y_j(t) \in \{0, 1\}$ with analogous dynamics governed by $p_{\text{fail}}^{\text{group}}$ and $p_{\text{recover}}^{\text{group}}$, yielding steady-state $\pi_y = p_{\text{recover}}^{\text{group}}/(p_{\text{fail}}^{\text{group}} + p_{\text{recover}}^{\text{group}})$. For sensor i in group j, the effective operational status is $x_i(t) = z_i(t) \cdot y_j(t)$, requiring both individual and group processes to be up. Under independence, the effective steady-state probability is $\pi_x = \pi_z \cdot \pi_y$. The effective failure and recovery probabilities are:

$$p_{\text{fail}}^{\text{eff}} = 1 - (1 - p_{\text{fail}})(1 - p_{\text{fail}}^{\text{group}}), \tag{1}$$

$$p_{\text{recover}}^{\text{eff}} \approx p_{\text{recover}} \cdot p_{\text{recover}}^{\text{group}}. \tag{2}$$

A wide array of failure dynamics can be simulated with this model, including fast individual failures, fast group failures, mixed dynamics, and slow recovery with prolonged outages.

4 Sequence-based PPO Agents

Motivation. Standard PPO agents often use a feed-forward MLP policy $\pi_\theta(a_t \mid s_t)$ that maps the *current* observation/state to an action. In partially observable settings or under sensor dropouts, conditioning on only s_t can induce brittle behavior because the policy ignores temporal context. We therefore couple PPO with sequence encoders that summarize recent history (Transformers) or maintain a recurrent hidden state (RNNs/SSMs).

4.1 Transformer-based PPO agent

History buffer. For each parallel environment, we maintain a circular buffer of the most recent L observations $\mathcal{B}_t = (o_{t-L+1}, \ldots, o_t)$, where L is the configured sequence length. At time t, we form a length-L sequence $X_t \in \mathbb{R}^{L \times d_{\text{in}}}$ by rolling the buffer so that the oldest valid element appears first, and we construct a padding mask $M_t \in \{0, 1\}^L$ indicating which positions are invalid (e.g., before the buffer is filled or after an environment resets).

Encoder. We first project each observation to the model dimension d and add *sinusoidal* positional encodings:

$$\tilde{X}_t = \text{PE}(X_t W_{\text{in}} + b_{\text{in}}), \qquad \tilde{X}_t \in \mathbb{R}^{L \times d}. \tag{3}$$

We then apply a Transformer encoder with temporal self-attention using the key-padding mask M_t:

$$H_t = \text{TransformerEnc}(\tilde{X}_t; M_t), \qquad H_t = (h_{t,1}, \ldots, h_{t,L}),\ h_{t,i} \in \mathbb{R}^d. \tag{4}$$

Attention pooling. To obtain a fixed-size feature vector that can be fed into separate actor and critic heads, we apply learned attention pooling over time:

$$e_{t,i} = w^\top h_{t,i} + b, \qquad \alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j:\, M_{t,j}=0} \exp(e_{t,j})}, \qquad z_t^{\text{attn}} = \sum_{i:\, M_{t,i}=0} \alpha_{t,i} h_{t,i}. \tag{5}$$

4.2 RNN/SSM-based PPO Agent

Overview. We unify recurrent neural networks (RNNs; e.g., GRU/LSTM) and recurrent state-space models (SSMs; e.g., LRU) under a common *recurrent latent-state encoder* that maintains memory across time. At each step, the policy conditions on a contextual feature z_t computed from the current input and a carried hidden state h_{t-1}. This allows the PPO policy to use temporal context under partial observability while keeping the PPO objective unchanged.

Generic recurrent encoder. Let o_t denote the observation and $x_t \in \mathbb{R}^d$ represent an input embedding:

$$x_t = g_{\text{enc}}(o_t). \tag{6}$$

We model both RNNs and SSMs as a stateful mapping

$$(h_t,\ z_t) = \mathcal{E}_\psi(h_{t-1}, x_t; d_t), \tag{7}$$

where h_t is the recurrent memory state, z_t is the emitted feature, and $d_t \in \{0, 1\}$ is the episodic done flag. Similarly to the Transformer-based agent, the feature vector z_t is then passed to separate actor and critic heads.

5 THEORY

We prove a high-probability bound on the infinite-horizon reward degradation for the RL agent under the stochastic sensor failure model.

The notation is as follows. $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ is a discounted Markov decision process (MDP) with discount $0 < \gamma < 1$. The random one-step reward at (s, a) is $r(s, a)$ and its expectation is $R(s, a) := \mathbb{E}[r(s, a)]$. A stochastic policy π maps observations to action distributions. Given an observation map $h : \mathcal{S} \to \mathbb{R}^d$, actions are drawn as $a_t \sim \pi(\cdot \mid h(S_t))$. The *action–value function* (or Q-function) of π is

$$Q^\pi(s, a) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t\, r(S_t, a_t) \;\middle|\; S_0 = s,\ a_0 = a,\ S_{t+1} \sim P(\cdot \mid S_t, a_t),\ a_{t\geq 1} \sim \pi(\cdot \mid h(S_t))\right]. \tag{8}$$

Equivalently, Q^π satisfies the Bellman expectation equation

$$Q^\pi(s, a) = R(s, a) + \gamma\, \mathbb{E}_{S' \sim P(\cdot|s,a)}\Big[\, \mathbb{E}_{a' \sim \pi(\cdot|h(S'))}\big[\, Q^\pi(S', a')\,\big]\Big]. \tag{9}$$

The *state value function* is

$$V^\pi(s) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t\, r(S_t, a_t) \;\middle|\; S_0 = s\right] = \mathbb{E}_{a\sim\pi(\cdot|h(s))}[\, Q^\pi(s, a)\,], \tag{10}$$

and the *advantage* is $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$.

A mask $m_t \in \{0, 1\}^d$ zeroes failed sensors: $h_{m_t}(s_t) = m_t \odot h(s_t)$. The mask process $\{M_t \equiv m_t\}_{t\geq 0}$ is generated by the two-layer Markov model (per-sensor chains and group chains). The stationary marginal probability that sensor i is "up" is denoted $\pi_{x,i}$ (for identical sensors we write π_x). For the two-layer model this marginal is the product formula: $\pi_x = \frac{p_{\text{recover}}}{p_{\text{fail}} + p_{\text{recover}}} \cdot \frac{p_{\text{grouprecover}}}{p_{\text{groupfail}} + p_{\text{grouprecover}}}$.

We make the following assumptions.

Assumption 5.1 (Bounded sensor outputs)**.** *For all $s \in \mathcal{S}$ and $i \in \{1, \ldots, d\}$, $|h_i(s)| \leq B_i < \infty$.*

Assumption 5.2 (Policy smoothness — Wasserstein Lipschitzness)**.** *Let $(\mathcal{A}, \|\cdot\|)$ be the action space with the metric inducing the 1-Wasserstein distance W_1 on $\mathcal{P}(\mathcal{A})$. There exists $L_\pi \geq 0$ such that for all observations $o, o' \in \mathbb{R}^d$,*

$$W_1\big(\pi(\cdot \mid o),\, \pi(\cdot \mid o')\big) \;\leq\; L_\pi\, \|o - o'\|_1.$$

Assumption 5.3 (Q^π is Lipschitz in action)**.** *There exists $L_Q \geq 0$ such that for all $s \in \mathcal{S}$ and $a, a' \in \mathcal{A}$,*

$$\big|Q^\pi(s, a) - Q^\pi(s, a')\big| \;\leq\; L_Q\, \|a - a'\|.$$

Assumption 5.4 (Augmented-chain geometric ergodicity)**.** *Fix the policy π. The augmented process $\Xi_t = (S_t, M_t)$ is an irreducible, aperiodic Markov chain on $\mathcal{S} \times \{0, 1\}^d$ with stationary law π_Ξ, and is geometrically ergodic. Write its total-variation mixing time at tolerance $1/8$ as*

$$\tau \;:=\; t_{\text{mix}}(1/8) \;:=\; \inf\Big\{t \geq 0 :\ \sup_{\xi_0} \big\|\mathcal{L}(\Xi_t \,|\, \Xi_0 = \xi_0) - \pi_\Xi\big\|_{\text{TV}} \leq \tfrac{1}{8}\Big\}.$$

Assumption 5.5 (Mask exogeneity / independence)**.** *Under the stationary law of Ξ_t, M_t is independent of S_t and has stationary distribution π_M with marginals $\mathbb{P}(M_{t,i} = 1) = \pi_{x,i}$. Consequently, for all i,*

$$\mathbb{E}\big[(1 - M_{t,i})\, |h_i(S_t)|\big] \;=\; (1 - \pi_{x,i})\, \mathbb{E}_{S\sim d_\pi}\big[|h_i(S)|\big] \;:=\; (1 - \pi_{x,i})\, h_i,$$

where d_π is the stationary state distribution under π.

Loss variables. Define the *one-step counterfactual gap in future return* at time t as

$$\Delta_t \;:=\; \mathbb{E}_{a\sim\pi(\cdot|h(S_t))}\big[Q^\pi(S_t,a)\big] \;-\; \mathbb{E}_{a\sim\pi(\cdot|h_{M_t}(S_t))}\big[Q^\pi(S_t,a)\big],$$

its nonnegative version $X_t := |\Delta_t|$, and the discounted cumulative loss

$$S \;:=\; \sum_{t=0}^{\infty}\gamma^t X_t, \qquad \mu_S := \mathbb{E}[S].$$

Let $C_{\max} \;:=\; L_Q\, L_\pi \sum_{i=1}^{d} B_i$. Δ_t measures, in units of discounted *future* return, how much is lost at time t by drawing the action from the masked-action distribution $\pi(\cdot \mid h_{M_t}(S_t))$ instead of the full-observation distribution $\pi(\cdot \mid h(S_t))$, while evaluating consequences with the same Q^π.

Theorem 5.6 (High-probability reward-degradation bound)**.** *Assume 5.1–5.5. Fix $\delta \in (0,1)$. Then, with probability at least $1-\delta$,*

$$S \;\le\; \mu_S \;+\; C_{\max}\min\left\{\sqrt{\frac{2\tau}{1-\gamma^2}\,\ln\frac{2}{\delta}} \;+\; \frac{4}{3}\,\tau\,\ln\frac{2}{\delta}, \frac{1}{1-\gamma}\right\}.$$

Moreover, the mean satisfies

$$\mu_S \;\le\; \frac{L_Q L_\pi}{1-\gamma}\sum_{i=1}^{d}(1-\pi_{x,i})\,h_i, \qquad h_i := \mathbb{E}_{S\sim d_\pi}\big[|h_i(S)|\big].$$

Interpretation of the bound. The mean term μ_S captures the average loss and scales linearly with each sensor's down-time $(1-\pi_{x,i})$ and with the policy and critic sensitivities L_π and L_Q. Notably, only the marginal up-rates $\{\pi_{x,i}\}$ enter this term, so correlations between sensors do not directly affect the expected degradation. The stochastic deviation around this mean has two components: a square-root term $C_{\max}\sqrt{\frac{2\tau}{1-\gamma^2}\ln\frac{2}{\delta}}$ that grows with slower mixing (larger τ), longer effective horizon (larger γ), and desired confidence (smaller δ), and a linear correction $\frac{4}{3}C_{\max}\tau\ln\frac{2}{\delta}$ from the Bernstein inequality that is typically dominated by the square-root term unless δ is extremely small or τ is large. The leverage constant $C_{\max} = L_Q L_\pi \sum_i B_i$ represents the worst-case per-step impact, factoring out policy smoothness, critic smoothness, and observation scale, and thus globally scales both deviation terms.

Dependence on the mask process. In the two-layer mask model, the stationary per-sensor up-rate factorizes as $\pi_x = \pi_z\,\pi_y$ where $\pi_z = \frac{p_{\text{recover}}}{p_{\text{fail}}+p_{\text{recover}}}$ and $\pi_y = \frac{p_{\text{group recover}}}{p_{\text{group fail}}+p_{\text{group recover}}}$; higher recovery rates or lower failure rates increase π_x and reduce the mean term μ_S. The burstiness of outages, quantified by the mixing time τ, can be bounded conservatively using the spectral gaps of the sensor and group layers: $\tau \lesssim \frac{\ln 4}{\min\{g_{\text{sensor}},\, g_{\text{group}}\}}$ where $g_{\text{sensor}} := p_{\text{fail}} + p_{\text{recover}}$ and $g_{\text{group}} := p_{\text{group fail}} + p_{\text{group recover}}$. Slower group or sensor dynamics (smaller spectral gap) increase τ, which widens both the square-root and linear deviation terms in the high-probability bound.

6 Empirical Results

Our experimental RL environments are based on MuJoCo (Todorov et al., 2012), an RL physics engine for simulating continuous control tasks involving complex, articulated robots. These environments focus on locomotion, balance, and agility, providing rich benchmarks for developing and testing RL algorithms in high-dimensional, continuous control tasks.

We experiment on four standard MuJoCo continuous-control benchmarks: HalfCheetah-v4, Hopper-v4, Walker2d-v4, and Ant-v4. We train eight PPO-based agents under full observability and partial observability induced by our sensor failure model. The suite of models tested comprised an MLP baseline and the following sequence-based models:

- **RNNs/SSMs**: Linear Recurrent Unit (LRU) (Orvieto et al., 2023b) and Gated Recurrent Unit (GRU) (Cho et al., 2014)
- **Transformers**: Transformer (Vaswani et al., 2017), UniTS (Gao et al., 2024), and Gated Transformer-XL (GTrXL) (Dai et al., 2019)

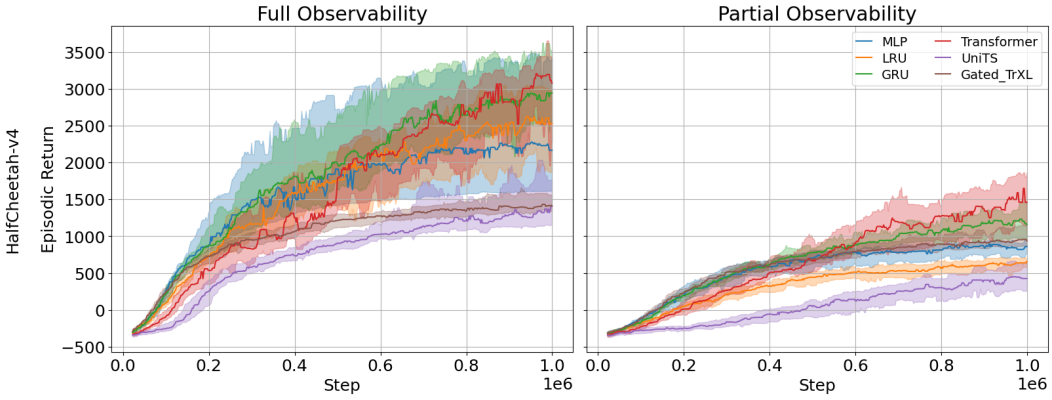

Figure 1: Sample PPO training curves on HalfCheetah-v4 under full (left) and 60% partial (right) observability. Lines represent median episodic return and shaded regions denote inter-quartile ranges across 8 random seeds. Training curves generated under partial observability rise more slowly and plateau at lower returns than those produced using fully observed states.

For our sensor model, we partition observations into three sensor groups with failure and recover probabilities $p_{\text{fail}} = 1\%$, $p_{\text{recover}} = 90\%$, $p_{\text{fail}}^{\text{group}} = 55\%$, and $p_{\text{recover}}^{\text{group}} = 90\%$, yielding an effective recovery rate $p_{\text{recover}}^{\text{eff}}$ of 60%. Failed sensors are implemented by masking normalized features to zero and appending a binary mask to differentiate dropped inputs from valid zero-mean inputs.

All models share a fixed PPO configuration using CleanRL defaults for continuous-control PPO (Huang et al., 2022), as well as matched architectural capacities (where applicable) and randomly initialized priors. Detailed hyperparameter choices can be found in A.3.2. Agents are trained for one million steps within a single environment instance across 8 random seeds, and minibatches for sequence-based agents comprise trajectory segments of length 16 for truncated backpropagation through time. For models dependent on hidden states, an initial burn-in of 8 timesteps was used to initialize hidden states prior to each gradient computation. For the Transformer- and UniTS-based agents, encoder dropout is enabled during training for stability. Performance during training was measured as the average episodic return across the batch collected per epoch, and the final training curves shown in Figures 1 and 3 were generated by plotting the smoothed medians and inter-quartile ranges of the aggregated results across all seeds.

For evaluation, episodic returns were computed over 100 episodes across all eight seeds using a deterministic version of the learned policy, where actions were taken as the policy mean rather than sampled. The evaluation results per model were then aggregated, and the median performance with 95% confidence intervals was estimated via bootstrap resampling. Evaluation results are presented in Figure 2.

Because MuJoCo environments are mostly Markovian, agents do not require much memory to achieve high rewards under full observability, as each observation contains sufficient information for optimal control. The MLP frequently achieves the highest returns under full observability, benefiting from its architectural simplicity and straightforward feature transformations. Performance among sequence-based policies in the fully observed regime proves more task dependent. On HalfCheetah, multiple sequence models are competitive: the Transformer achieves the strongest performance, and simpler recurrent models like the GRU and LRU also perform well, suggesting that both explicit and latent memory can be beneficial in certain cases even when the state is fully observed. However, on more challenging environments, sequence models underperform to varying degrees, indicating that additional temporal structure and architectural complexity can also be a liability when the current observation is already sufficient. Overall, Figure 2 indicates that providing temporal context under full observability often does not reliably improve returns and fails to outperform the MLP baseline.

Under the partial observability induced by the sensor failure model, all agents exhibit reduced performance relative to the fully observed setting, confirming that intermittent feature dropout substantially increases task difficulty. Yet, the degree of degradation strongly depends on the architecture and task. Without temporal context, the MLP experiences the most significant drops in performance

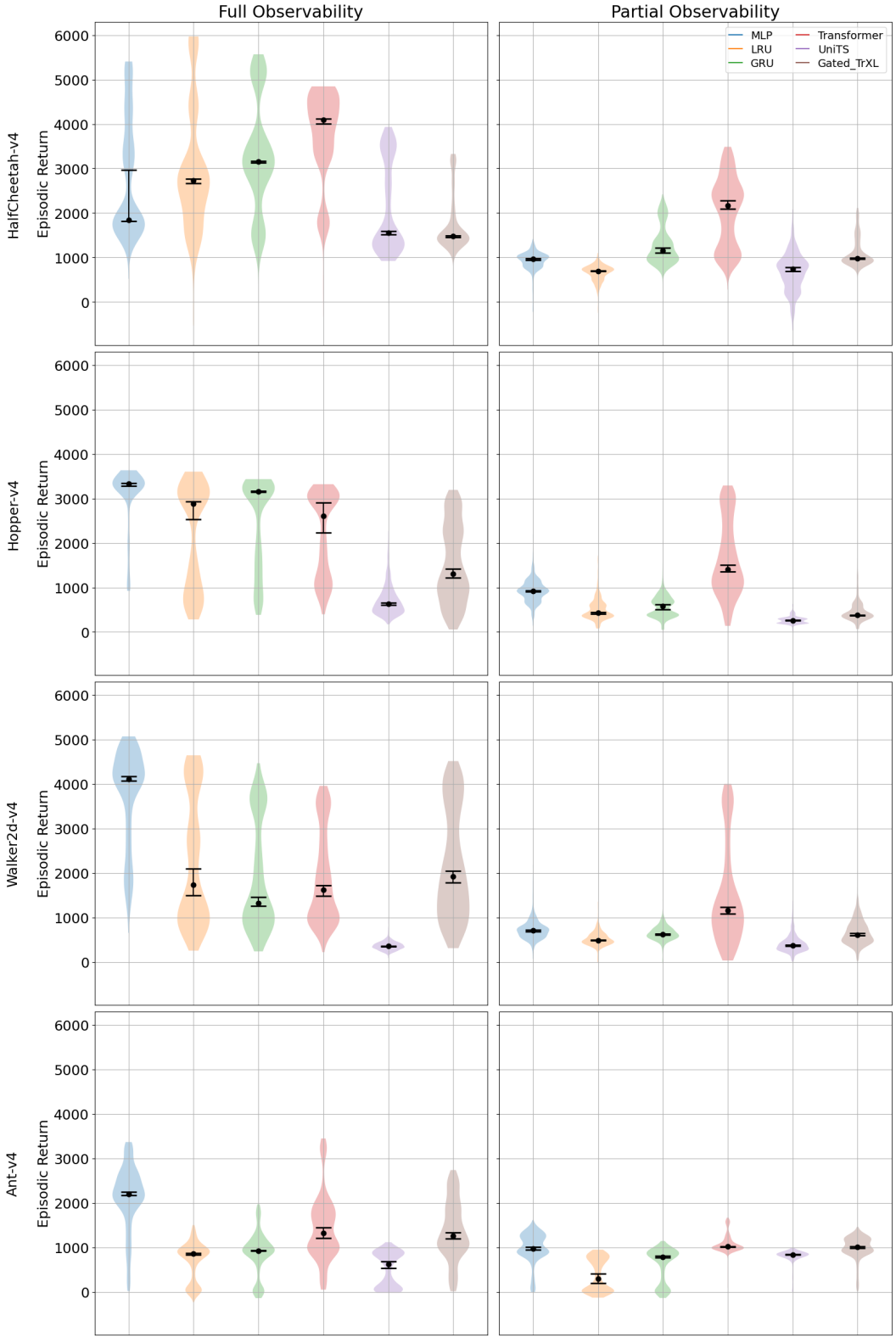

Figure 2: Evaluation episodic returns for PPO agents on MuJoCo environments under full (left) and 60% partial (right) observability, with task complexity roughly increasing from top to bottom. Each violin shows the distribution of pooled episodic returns from 100 episodes across 8 random seeds. Black markers denote the median with 95% bootstrapped CI. While all models suffer performance degradation under partial observability, the Transformer agent demonstrates greater robustness.

on average, particularly in Hopper and Walker2d (Figure 2). Agents that rely on memory primarily through latent recurrence – whether implemented via RNNs, state space dynamics, or the gated recurrent attention mechanisms seen in GTrXL – also demonstrate a substantial lack of robustness. In very few settings, these latent-memory models do outperform the MLP by a small margin, indicating that recurrence can occasionally compensate for missing features, but more often they exhibit noticeably heavier low-return tails. The Transformer policy, on the other hand, proves to be the most robust of all models under partial observability, scoring the highest evaluation medians across all environments and maintaining relatively stable performance. Of the tasks evaluated, Ant proves comparatively challenging under partial observability, and most sequence-based agents do not consistently translate memory over the high-dimensional state into improved returns.

7 Discussion

In reinforcement learning settings with missing state information and online PPO, the combination of non-stationarity, partial observability, and the need for flexible memory across variable timescales creates significant architectural challenges. RNNs, SSMs, and even Gated Transformer-XL struggle in this regime because their recurrent dynamics impose strict constraints on memory evolution, treating inputs uniformly and assuming regular input streams. When state information is missing, these assumptions about smoothness and regularity can be violated, causing the recurrent dynamics to diverge or lose critical information, with limited ability to selectively retrieve specific past observations. In contrast, stateless Transformers process all variables jointly within a single sequence and leverage self-attention mechanisms that allow each output to directly attend to all available past tokens, effectively learning temporal correlations by having state dimensions participate in the attention computation. This architecture provides inherent robustness to missing data: the self-attention mechanism can dynamically identify and utilize whichever past observations are present while naturally skipping over gaps, enabling Transformers to maintain performance even under irregular observability by flexibly referencing earlier relevant inputs without being constrained by assumptions of temporal regularity. Our experiments support these points.

A notable exception among the sequence models is UniTS, which underperforms across all settings. We hypothesize that this result is due largely to an inductive bias mismatch. While the Transformer processes all variables jointly, UniTS processes each variable independently during sequence attention, and cross-variable attention happens separately. Though this factorization is beneficial for creating unified representations of diverse time series, it may hinder learning of joint temporal patterns in continuous control RL, since cross-variable interactions are largely deferred to variable-level attention.

8 Conclusion

In this work, we studied the robustness of Proximal Policy Optimization under temporally persistent sensor failures, framing observation drift as partial observability with structured temporal correlations. By augmenting PPO with sequence-based policy architectures, we showed that agents can leverage history to infer missing information and maintain performance when observations are unreliable.

We derived high-probability bounds on infinite-horizon reward degradation under a stochastic sensor failure model, clarifying how robustness depends on policy smoothness, critic sensitivity, sensor availability, and failure persistence. Empirically, experiments on MuJoCo benchmarks confirm these insights: while sequence models offer limited benefit under full observability, they substantially improve robustness under sensor dropout. Stateless Transformer-based PPO agents in particular consistently outperform MLP, RNN, and state-space baselines, achieving higher returns and more stable behavior when large fractions of sensors are unavailable.

Overall, our results demonstrate that temporal sequence modeling provides a principled and effective mechanism for robust reinforcement learning in unreliable environments, highlighting attention-based architectures as a promising direction for real-world deployment under observation drift.

ACKNOWLEDGMENTS

The authors acknowledge the MIT Lincoln Laboratory Supercomputing Center for providing the high-performance computing resources that have contributed to the research results reported in this paper.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of War for Research and Engineering under Air Force Contract No. FA8702-15-D-0001 or FA8702-25-D-B002. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of War for Research and Engineering. © 2026 Massachusetts Institute of Technology.

Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

REFERENCES

T. Brown et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Sankar Narayan Das, Sudip Misra, Bernd E. Wolfinger, and Mohammad S. Obaidat. Temporal-correlation-aware dynamic self-management of wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 12(6):2127–2138, 2016. doi: 10.1109/TII.2016.2594758.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

S. Gao, T. Koker, O. Qween, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik. Units: A unified multi-task time series model. In *NeurIPS*, 2024.

A. Gu and T. Dao. Linear-time sequence modeling with selective state spaces. In *COLM*, 2024.

A. Gu, K. Goel, U. Shalit, and K. Simonyan. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI*, 2015.

M. Huang and S. Dey. Stability of kalman filtering with markovian packet losses. In *Automatica*, 2007.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G. M. Araújo. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Tommi Jaakkola, Satinder Singh, and Michael Jordan. Reinforcement Learning Algorithm for Partially Observable Markov Decision Problems. In G. Tesauro, D. Touretzky, and T. Leen (eds.), *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://proceedings.neurips.cc/paper_files/paper/1994/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

Q. Liu, K. He, and L. Jiang. Optimal lqg control and stability of networked robot system with data dropout. 2006.

Y. Mo, E. Garone, and B. Sinopoli. Lqg control with markovian packet loss. In *European Control Conference (ECC)*, 2013.

Steven Morad, Ryan Kortvelesy, Matteo Bettini, Stephan Liwicki, and Amanda Prorok. Popgym: Benchmarking partially observable reinforcement learning, 2023. URL https://arxiv.org/abs/2303.01859.

A. Orvieto, L. Smith, A. Gu, A. Fernando, C. Gulcehre, R. Pascanu, and S. De. Resurrecting recurrent neural networks for long sequences. In *ICML*, 2023a.

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

D. Paulin. Concentration inequalities for markov chains by marton couplings and spectral methods. *Electronic Journal of Probability*, 20(79):1–32, 2015.

Marco Pleines, Matthias Pallasch, Frank Zimmer, and Mike Preuss. Memory gym: Partially observable challenges to memory-based agents. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jHc8dCx6DDr.

K. Rusch and D. Rus. Oscillatory state-space models. In *ICLR*, 2025.

Ivan Smirnov and Shangding Gu. Rlbenchnet: The right network for the right reinforcement learning task. *arXiv preprint arXiv:2505.15040*, 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

A. Vaswani et al. Attention is all you need. In *NeurIPS*, 2017.

Mehmet C. Vuran, Özgür B. Akan, and Ian F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245–259, 2004. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2004.03.007. URL https://www.sciencedirect.com/science/article/pii/S1389128604000519.

V. Zambaldi, D. Raposo, A. Santoro, et al. Deep reinforcement learning with relational inductive biases. In *ICLR*, 2019.

A APPENDIX

A.1 PROOF OF THEOREM 5.6

The proof of Theorem 5.6 is based on two lemmas.

Lemma A.1 (Pointwise Wasserstein bound on the per-step loss)**.** *Under Assumptions 5.2–5.3,*

$$|\Delta_t| \;=\; \big|\mathbb{E}_{a\sim\pi(\cdot|h(S_t))}Q^\pi(S_t,a) - \mathbb{E}_{a\sim\pi(\cdot|h_{M_t}(S_t))}Q^\pi(S_t,a)\big| \;\le\; L_Q\, W_1\big(\pi(\cdot \mid h(S_t)), \pi(\cdot \mid h_{M_t}(S_t))\big)$$

$$\le\; L_Q L_\pi \,\|h(S_t) - h_{M_t}(S_t)\|_1 \;=\; L_Q L_\pi \sum_{i=1}^{d} (1 - M_{t,i})\,|h_i(S_t)| \;\le\; C_{\max}.$$

Proof. *Kantorovich–Rubinstein duality gives $|\mathbb{E}_\nu f - \mathbb{E}_{\nu'} f| \le \mathrm{Lip}(f)\, W_1(\nu,\nu')$ for 1-Lipschitz f; here $f(a) = Q^\pi(S_t,a)$ has $\mathrm{Lip}(f) \le L_Q$ by Assumption 5.3. Apply Assumption 5.2 and the mask identity $\|h - h_M\|_1 = \sum_i (1-M_i)|h_i|$, then Assumption 5.1.* □

The next lemma states a Markov-chain Bernstein-type concentration inequality.

Lemma A.2. *Paulin (2015)[Bernstein for geometrically ergodic chains] Let Ξ_t satisfy Assumption 5.4. Let $\{f_t\}_{t\ge0}$ be bounded functions with $|f_t(\xi)| \le b_t$ and define $S_n = \sum_{t=0}^{n-1} f_t(\Xi_t)$ and $Z_n = S_n - \mathbb{E}S_n$. Then, for any $u > 0$,*

$$\mathbb{P}(Z_n \ge u) \;\le\; \exp\!\left(-\frac{u^2}{2\tau\sum_{t=0}^{n-1} b_t^2 + \frac{4}{3} b_{\max}\tau\, u}\right), \quad b_{\max} := \max_{0\le t\le n-1} b_t.$$

Proof of Theorem 5.6. Step 0: deterministic pointwise bound. By Lemma A.1, $X_t = |\Delta_t| \le C_{\max}$ and hence $0 \le \gamma^t X_t \le \gamma^t C_{\max}$ for all t. This also implies the upper cap bound $S \le \frac{C_{max}}{1-\gamma}$. Trivially, $S \le \mu_S + \frac{C_{max}}{1-\gamma}$.

Step 1: apply Bernstein to the weighted sequence. Let $f_t(\Xi_t) := \gamma^t X_t$. Then $|f_t| \le b_t$ with $b_t := \gamma^t C_{\max}$ and $b_{\max} = C_{\max}$. Define $S_n := \sum_{t=0}^{n-1} \gamma^t X_t$ and $Z_n := S_n - \mathbb{E}[S_n]$. By Lemma A.2,

$$\mathbb{P}(Z_n \ge u) \;\le\; \exp\!\left(-\frac{u^2}{2\tau\sum_{t=0}^{n-1} \gamma^{2t} C_{\max}^2 + \frac{4}{3} C_{\max}\tau\, u}\right).$$

Step 2: infinite-horizon limit. Since $\sum_{t=0}^{\infty} \gamma^{2t} = \frac{1}{1-\gamma^2}$, letting $n \to \infty$ and using monotone convergence yields

$$\mathbb{P}(S - \mu_S \ge u) \;\le\; \exp\!\left(-\frac{u^2}{2\tau\, C_{\max}^2/(1-\gamma^2) + \frac{4}{3} C_{\max}\tau\, u}\right).$$

Step 3: solve the quadratic tail. Set the right-hand side to $\delta/2$ and solve the Bernstein quadratic inequality for u. A standard inversion gives the convenient choice

$$u \;=\; C_{\max}\sqrt{\frac{2\tau}{1-\gamma^2}\,\ln\frac{2}{\delta}} \;+\; \frac{4}{3}\, C_{\max}\,\tau\,\ln\frac{2}{\delta},$$

which implies $\mathbb{P}\big(S - \mu_S \ge u\big) \le \delta/2$ (an analogous lower-tail bound also holds, but we only need the upper tail).

Step 4: bound the mean via independence. By Lemma A.1,

$$\mu_S \;=\; \mathbb{E}\left[\sum_{t\ge0} \gamma^t X_t\right] \;\le\; \frac{L_Q L_\pi}{1-\gamma} \sum_{t\ge0} \gamma^t\, \mathbb{E}\left[\sum_{i=1}^{d} (1 - M_{t,i})\,|h_i(S_t)|\right].$$

Under stationarity and Assumption 5.5, $\mathbb{E}[(1 - M_{t,i})\,|h_i(S_t)|] = (1 - \pi_{x,i})\,h_i$, hence $\mu_S \le \frac{L_Q L_\pi}{1-\gamma}\sum_{i=1}^{d}(1-\pi_{x,i})\,h_i$. Combining Steps 1–3 with this mean bound yields the stated inequality. □

Remark A.3 (Signed version)**.** *Defining $S_\Delta := \sum_{t\ge0}\gamma^t\Delta_t$ and repeating the proof with Δ_t (instead of $|\Delta_t|$) gives the same deviation term and a two-sided bound for S_Δ about its mean. Since $|S_\Delta| \le S$, the result in Thm. 5.6 is a conservative degradation guarantee.*

A.2 GRU AND LRU SPECIFICS

GRU (RNN). (Cho et al., 2014) A GRU is obtained by letting $h_t \in \mathbb{R}^d$ and defining gate-controlled nonlinear updates:

$$\begin{aligned} r_t &= \sigma(W_r x_t + U_r h_{t-1}), \\ u_t &= \sigma(W_u x_t + U_u h_{t-1}), \\ \tilde{h}_t &= \tanh(W_h x_t + U_h(r_t \odot h_{t-1})), \\ h_t &= (1 - u_t) \odot h_{t-1} + u_t \odot \tilde{h}_t, \\ z_t &= h_t, \end{aligned}$$

optionally wrapped with residual connections and gating blocks in a stacked encoder.

LRU (diagonal linear SSM). (Orvieto et al., 2023a) An LRU-style SSM can be expressed by choosing h_t in a transformed (complex) latent space and using a linear recurrence with diagonal dynamics:

$$h_t = \Lambda h_{t-1} + B x_t,$$

where Λ controls retention/decay (analogous to a learned, state-dependent "carry" behavior), and z_t is produced by a learned readout followed by stabilizing blocks (e.g., residual/GLU/normalization) in a stacked encoder.

A.3 TRAINING DETAILS

A.3.1 RNN/SSM INTEGRATION

Episode boundary handling. To prevent leakage of information across episodes, we reset memory when an environment terminates:

$$h_{t-1} \leftarrow (1 - d_t)\, h_{t-1},$$

before applying the update in Eq. 7. This reset is applied during both rollout collection and training-time unrolling.

Truncated backpropagation and burn-in. During PPO updates, we unroll $\mathcal{E}_\psi$ over short subsequences. Optionally, we use a *burn-in* prefix to update hidden state without gradients and then compute PPO losses on the subsequent segment using the warmed state:

$$h_\tau \leftarrow \mathcal{E}_\psi(h_0, x_{1:\tau}; d_{1:\tau}) \quad \text{(no grad)}, \qquad \text{loss on } x_{\tau+1:\tau+K}.$$

This is implemented for our recurrent agents to ensure hidden states update alongside model weights each epoch.

A.3.2 HYPERPARAMETERS AND TRAINING CURVES

Table 1: Proximal Policy Optimization (PPO) Hyperparameters

Hyperparameter	**Value**
Total timesteps	1×10^6
Learning rate	3×10^{-4}
Number of steps	2048
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Number of minibatches	32
Update epochs	10
Clipping coefficient	0.2
Entropy coefficient	0.0
Value function coefficient	0.5
Maximum gradient norm	0.5

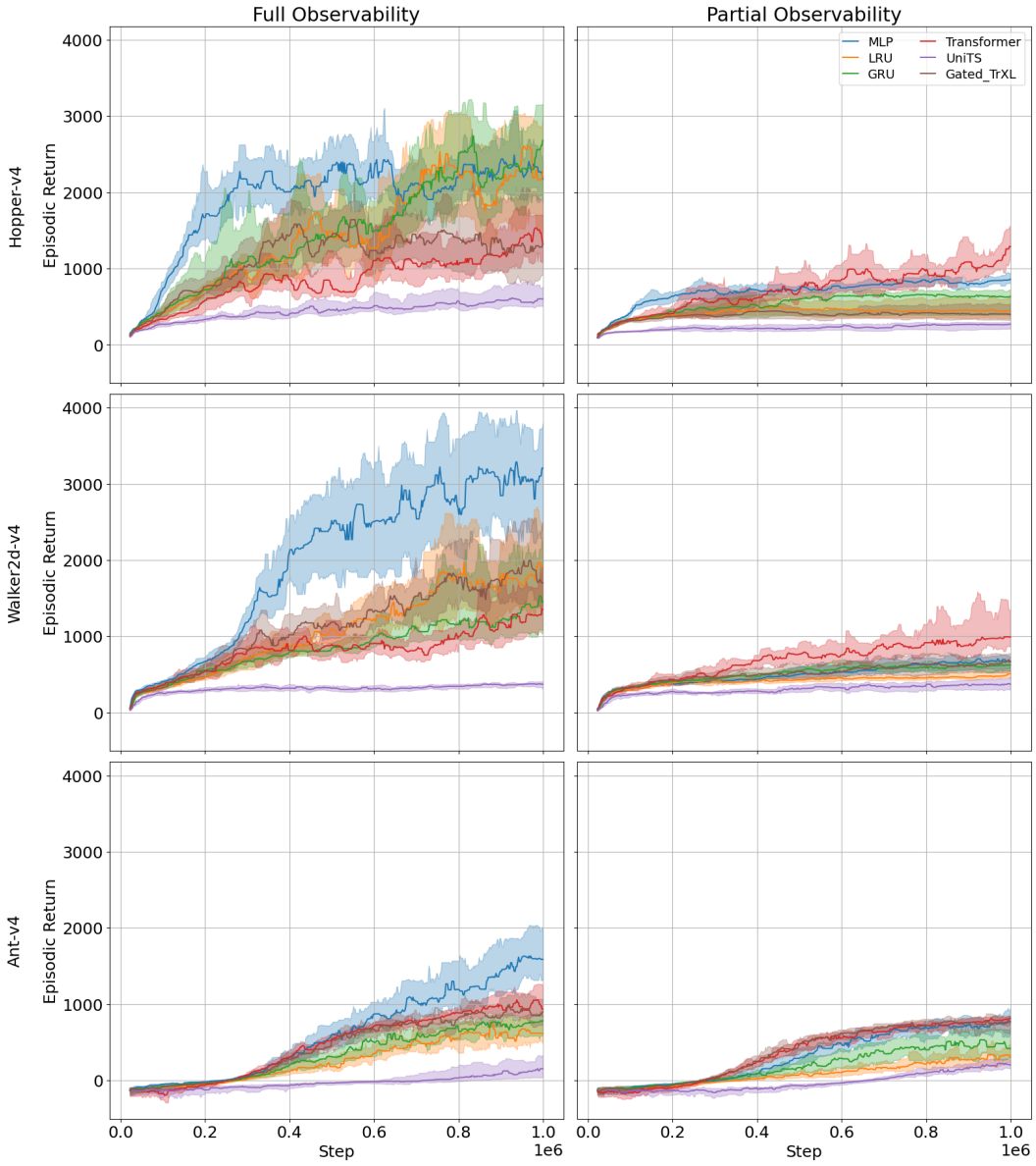

Figure 3: PPO training curves for Hopper-v4, Walker2d-v4, and Ant-v4 under full (left) and 60% partial (right) observability. Lines represent median episodic return and shaded regions denote inter-quartile ranges across 8 random seeds.

Table 2: Architectural Hyperparameters per Model

Model	**Hidden**	**Layers**	**Heads**	**Dropout**	**Additional**
Transformer	128	2	2	0.1	–
Gated TransformerXL	128	2	2	0	Memory length 8
UniTS	128	2	2	0.1	Patch length 1; Prompt tokens 6
GRU	128	4	–	0	–
LRU	128	4	–	0	$r_{\min}$ 0.9; $r_{\max}$ 0.999; Max phase 6.28
MLP	128	–	–	–	–