Transformers Provably Implement In-Context Reinforcement Learning with Policy Improvement

Haodong Liang, Lifeng Lai
Department of Electrical and Computer Engineering
University of California, Davis
{hdliang,lflai}@ucdavis.edu

Abstract

We investigate the ability of transformers to perform in-context reinforcement learning (ICRL), where a model must infer and execute learning algorithms from trajectory data without parameter updates. We show that a linear self-attention transformer block can provably implement policy-improvement methods, including semi-gradient SARSA and actor-critic, via explicit parameter constructions. Beyond existence, we design a teacher-mimicking training procedure, analyze its gradient-flow dynamics, and establish the first convergence guarantee in the ICRL literature: under suitable richness conditions on the training MDP distribution, gradient flow converges locally and exponentially to an optimal parameter manifold corresponding to the desired RL update. Empirically, training transformers on randomly generated tabular MDPs confirms these predictions: the learned models recover the parameter structure of our explicit constructions and, when deployed on unseen MDPs, deliver strong in-context control performance. Together, these results illuminate how transformer architectures internalize and execute classical reinforcement learning algorithms in context, bridging mechanistic understanding and training dynamics in ICRL.[1]

1 Introduction

In-context learning (ICL) refers to a model's ability to learn and perform a new task by conditioning on examples provided in its input, without any parameter updates [Brown et al., 2020, Garg et al., 2022, Dong et al., 2024]. The transformer architecture [Vaswani et al., 2017] has empirically demonstrated remarkable in-context learning capabilities across a wide range of tasks, among which Large Language Models (LLMs) are the most prominent example [Brown et al., 2020, Chowdhery et al., 2023, OpenAI et al., 2024].

This phenomenon has motivated substantial interest in understanding the mechanisms that enable transformers to learn in-context. A growing body of theoretical work has sought to understand this by asking: *which algorithms can a transformer execute in context, and why can it implement them?*

For supervised learning problems, this question now has a fairly complete answer: trained transformers can be shown to implement variants of gradient descent and closed-form least squares on the in-context examples, with explicit constructions and, in special cases, end-to-end training-dynamics guarantees [Garg et al., 2022, von Oswald et al., 2023, Akyürek et al., 2023, Ahn et al., 2023, Mahankali et al., 2024, Bai et al., 2023, Zhang et al., 2024, Liang et al., 2025].

Beyond the supervised learning scenario, recent works have begun to extend this lens to in-context reinforcement learning (ICRL)[Laskin et al., 2023, Lin et al., 2024, Moeini et al., 2025], where an agent is given a trajectory in its context and must internally execute a policy evaluation or policy

[1] Code is available at https://github.com/harris-liang/Transformer-in-context-RL.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

improvement step — core operations in the reinforcement learning framework [Sutton and Barto, 2018]. A central question in this line of work is:

Can a transformer block implement – and be trained to implement standard RL algorithms such as temporal-difference (TD) methods, SARSA, or actor-critic?

While these works, including Laskin et al. [2023], Lee et al. [2023], Brooks et al. [2023], and Raparthy et al. [2023], primarily focus on empirical experiments and demonstrate strong performance in various ICRL settings, they lack theoretical guarantees on the underlying mechanisms. Toward a theoretical answer, Wang et al. [2025] take a first step by showing that a linear-attention block admits a closed-form structure that implements policy evaluation via TD learning. Two questions, however, remain open. First, their analysis is restricted to policy evaluation, leaving open whether a transformer can also implement *policy-improvement* algorithms – a component that is no less central to RL. Second, existing analyses establish only the existence of an *invariant* set: once the parameters lie in the set, the expected update keeps them there. They do not show whether the training actually drives the parameters into this set in the first place, nor identify the conditions under which it does.

In this work, we close both gaps: we move from policy evaluation to *policy improvement* by showing that a single linear-attention block can implement classical on-policy control updates – SARSA and actor-critic – and we move from invariance to *convergence* by identifying sufficient conditions under which gradient training provably drives the transformer parameters into the theoretical manifold that exactly realizes the target update.

Our existence theorem is similar in form to Wang et al. [2025] in that both exhibit a closed-form weight configuration $(\mathbf{P}^\star, \mathbf{V}^\star)$ for a single linear-attention block, but there are substantial differences. In Wang et al. [2025], their transformer outputs only the value estimate at the query sample, so the construction is tailored to a fixed policy evaluation problem and cannot be reused to drive further interaction. Our construction instead directly outputs the updated parameters themselves – the critic parameter $\mathbf{w}$ for SARSA, and both the critic $\mathbf{w}$ and the actor $\boldsymbol{\lambda}$ for actor-critic – which can then be utilized to update policy, sample new trajectories, and close the in-context policy-improvement loop.

On the training-dynamics side, the optimum is no longer a single point but a continuous manifold of equivalent constructions, so global strong-convexity arguments do not apply directly; we instead establish a *local* Polyak–Łojasiewicz inequality on a tubular neighborhood of the manifold which, together with explicit richness conditions on the training-MDP distribution, yields exponential convergence of the gradient flow to a single point on the manifold.

Our contributions are summarized as follows:

- **Existence Proof.** We construct explicit parameters $\boldsymbol{\theta}^\star = (\mathbf{P}^\star, \mathbf{V}^\star)$ for a single-head linear self-attention block such that $\mathsf{TF}_{\boldsymbol{\theta}^\star}$ exactly realizes the batch SARSA semi-gradient update (Theorem 3.1) and the batch actor-critic update (Theorem 3.3).
- **Local convergence.** Under explicit richness assumptions on the training-MDP distribution (Assumption 1), we prove that by mimicking the teacher updates, the gradient flow on the population loss converges locally to a single point on the manifold (Theorem 3.2), with an analogous guarantee established for the actor-critic setting (Theorem I.1 in Appendix I). To our knowledge, these are the first convergence guarantees in the ICRL literature.
- **Empirical validation.** We design ICRL training algorithms (Algorithm 1, Algorithm 2) for both the SARSA and actor-critic settings, and show that the resulting trained transformers recover the theoretical block structure of the attention parameters $(\mathbf{P}^\star, \mathbf{V}^\star)$ (Section 4.1). Furthermore, when deployed on unseen MDPs, the trained transformers closely track their analytical teachers and drive policies toward oracle-level returns (Section 4.2).

Additional related work: We provide a detailed discussion of more related work in Appendix A.

2 Preliminaries

2.1 Transformer architecture

Following prior mechanistic analyses of in-context learning [von Oswald et al., 2023, Mahankali et al., 2024, Wang et al., 2025], we adopt linear attention for theoretical tractability. We consider

a one-layer, single-head transformer block with parameters $\boldsymbol{\theta} = \{(\mathbf{P}, \mathbf{V})\} \subseteq \mathbb{R}^{D\times D}$. Given input $\mathbf{H} \in \mathbb{R}^{D\times n}$, the output of the attention layer is given by

$$\mathbf{H}_{\text{out}} := \mathbf{H} + \frac{1}{n}(\mathbf{V}\mathbf{H})(\mathbf{H}^\top \mathbf{P}\mathbf{H}) \in \mathbb{R}^{D\times n}, \tag{1}$$

where $\mathbf{P} := \mathbf{Q}^\top \mathbf{K}$ absorbs the query and key projections. The transformer output is given by $\mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}) := f_{\text{read}}(\mathbf{H}_{\text{out}})$ for a fixed readout $f_{\text{read}} : \mathbb{R}^{D\times n} \to \mathbb{R}^{D_{\text{out}}}$.

2.2 Markov decision process

A Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, reward $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount $\gamma \in [0, 1)$, where $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$. The goal is to find a policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maximizing the (expected) return $G_t := \sum_{k=0}^{\infty} \gamma^k r_{t+1+k}$. The state- and action-value functions under π are

$$v_\pi(s) := \mathbb{E}_\pi[G_t \mid s_t = s], \qquad q_\pi(s, a) := \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a].$$

2.3 SARSA

SARSA is an on-policy value-based algorithm. Given a feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and parameter $\mathbf{w} \in \mathbb{R}^d$, we consider linear function approximation $q_{\mathbf{w}}(s, a) = \mathbf{w}^\top \boldsymbol{\phi}(s, a)$, and the batch semi-gradient SARSA update over an n-step trajectory is

$$\mathbf{w}_{\text{SARSA}} = \mathbf{w} + \frac{\alpha}{n} \sum_{i=0}^{n-1} \big[r_{i+1} + \gamma \mathbf{w}^\top \boldsymbol{\phi}(s_{i+1}, a_{i+1}) - \mathbf{w}^\top \boldsymbol{\phi}(s_i, a_i)\big] \boldsymbol{\phi}(s_i, a_i), \tag{2}$$

where $\alpha > 0$ is the step size and the bracketed quantity is the temporal-difference (TD) error.

2.4 Actor-Critic

Actor-Critic combines a parametric policy (the actor) and a parametric value function (the critic). Given a policy feature map $\boldsymbol{\phi}_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$, a value feature map $\boldsymbol{\phi}_V : \mathcal{S} \to \mathbb{R}^d$, and parameters $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\mathbf{w} \in \mathbb{R}^d$, with softmax policy $\pi_{\boldsymbol{\lambda}}(a \mid s) \propto \exp(\boldsymbol{\lambda}^\top \boldsymbol{\phi}_\pi(s, a))$ and linear critic $v_{\mathbf{w}}(s) = \mathbf{w}^\top \boldsymbol{\phi}_V(s)$, the batch update over an n-step trajectory is

$$\begin{aligned} \mathbf{w}_{\text{AC}} &= \mathbf{w} + \frac{\beta}{n} \sum_{i=0}^{n-1} \big[r_{i+1} + \gamma \mathbf{w}^\top \boldsymbol{\phi}_V(s_{i+1}) - \mathbf{w}^\top \boldsymbol{\phi}_V(s_i)\big] \boldsymbol{\phi}_V(s_i), \\ \boldsymbol{\lambda}_{\text{AC}} &= \boldsymbol{\lambda} + \frac{\alpha}{n} \sum_{i=0}^{n-1} \gamma^i \big[r_{i+1} + \gamma \mathbf{w}^\top \boldsymbol{\phi}_V(s_{i+1}) - \mathbf{w}^\top \boldsymbol{\phi}_V(s_i)\big] \mathbf{g}_{\boldsymbol{\lambda}}(s_i, a_i), \end{aligned} \tag{3}$$

where $\alpha, \beta > 0$ are step sizes, and $\mathbf{g}_{\boldsymbol{\lambda}}(s, a) := \boldsymbol{\phi}_\pi(s, a) - \sum_b \boldsymbol{\phi}_\pi(s, b)\pi_{\boldsymbol{\lambda}}(b \mid s)$ is the score function.

3 Main Results

We demonstrate that a single-head transformer block can learn to perform in-context policy improvement using classical on-policy RL algorithms. Figure 1 illustrates the ICRL pipeline. Given a formulated input prompt $\mathbf{H}(z)$ constructed from the training sample z, we show that there exists $\boldsymbol{\theta}^\star$ such that the transformer block $\mathsf{TF}_{\boldsymbol{\theta}^\star}$ directly outputs the updated parameter $\mathbf{w}$, which is then used to update the value function or policy. In Section 3.1 we establish the existence of such implementations of SARSA, and in Section 3.2 we analyze the training dynamics of the proposed training procedure and identify sufficient conditions for convergence to the optimum parameter manifold. We then extend these results to actor-critic in Section 3.3.

3.1 Transformers can implement SARSA

Let $z = (\mathcal{M}, \boldsymbol{\phi}, \mathbf{w}, \tau)$ denote a training sample, where $\mathcal{M}$ is the training MDP task, $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is the feature map, $\mathbf{w} \in \mathbb{R}^d$ is the parameter of the behavior policy π, and $\tau = (s_0, a_0, r_1, \ldots, s_n, a_n)$

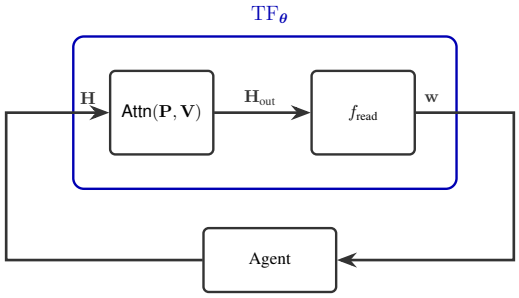


Figure 1: A simplified pipeline of transformer's ICRL implementation. The agent samples a trajectory under the current policy, which is fed into the transformer block. The transformer outputs an updated parameter $\mathbf{w}$, which is then used to update the value function and policy.

is the resulting n-step trajectory in $\mathcal{M}$. For notational convenience, we define the following quantities for $i = 0, \ldots, n-1$:

$$\boldsymbol{\phi}_i := \boldsymbol{\phi}(s_i, a_i) \in \mathbb{R}^d, \quad \boldsymbol{\phi}_i^+ := \boldsymbol{\phi}(s_{i+1}, a_{i+1}) \in \mathbb{R}^d, \quad \delta_i := r_{i+1} + \gamma \mathbf{w}^\top \boldsymbol{\phi}_i^+ - \mathbf{w}^\top \boldsymbol{\phi}_i \in \mathbb{R},$$

$$\mathbf{x}_i := \begin{bmatrix} \boldsymbol{\phi}_i \\ \gamma \boldsymbol{\phi}_i^+ \\ r_{i+1} \end{bmatrix} \in \mathbb{R}^{2d+1}, \quad \tilde{\mathbf{w}} := \begin{bmatrix} 1 \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+1}.$$

We also define the following trajectory statistics:

$$\hat{\boldsymbol{\Sigma}}(z) := \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{x}_i \mathbf{x}_i^\top, \quad \mathbf{R}(z) := \frac{1}{n} \sum_{i=0}^{n-1} \boldsymbol{\phi}_i \mathbf{x}_i^\top, \quad \mathbf{b}(z) := \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{x}_i \, \delta_i.$$

Consider the training prompt as follows:

$$\mathbf{H}(z) = \begin{bmatrix} \mathbf{x}_0 & \cdots & \mathbf{x}_{n-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{w}} \end{bmatrix} \in \mathbb{R}^{D \times (n+1)}, \tag{4}$$

where $D = 3d + 2$. We now establish the existence of a transformer block that implements the batch semi-gradient SARSA update (2).

Theorem 3.1 (Implement semi-gradient SARSA with a transformer block)**.** *Let the output projection function be $f_{read}(\cdot) = \cdot_{-d:,-1}$, which extracts the last d elements of the last column. Given the input prompt* (4)*, for any fixed $\alpha > 0$, there exists $\boldsymbol{\theta}^\star = \{(c\mathbf{P}^\star, c^{-1}\mathbf{V}^\star)\} \subseteq \mathbb{R}^{D \times D}$ with $c \neq 0$ and*

$$\mathbf{P}^\star = \begin{bmatrix} *_{(2d+1)\times(2d+1)} & \begin{bmatrix} \mathbf{0}_{d\times 1} & -\mathbf{I}_d \\ \mathbf{0}_{d\times 1} & \mathbf{I}_d \\ 1 & \mathbf{0}_{1\times d} \end{bmatrix} \\ *_{(d+1)\times(2d+1)} & \mathbf{0}_{(d+1)\times(d+1)} \end{bmatrix}, \quad \mathbf{V}^\star = \begin{bmatrix} *_{(2d+1)\times(2d+1)} & *_{(2d+1)\times(d+1)} \\ \begin{bmatrix} *_{1\times d} & *_{1\times d} & * \\ \alpha \mathbf{I}_d & \mathbf{0}_{d\times d} & \mathbf{0}_{d\times 1} \end{bmatrix} & \begin{bmatrix} *_{1\times(d+1)} \\ \mathbf{0}_{d\times(d+1)} \end{bmatrix} \end{bmatrix}, \tag{5}$$

where the asterisks denote arbitrary entries that do not affect the transformer output. With these parameters, $\mathsf{TF}_{\boldsymbol{\theta}^\star}(\mathbf{H}(z))$ directly implements the batch semi-gradient SARSA update (2)*.*

The proof is deferred to Appendix E. Theorem 3.1 guarantees the existence of a parameter configuration $\boldsymbol{\theta}^\star$ that realizes the semi-gradient SARSA update, but does not prescribe how to obtain it. To bridge this gap, we need to develop a training algorithm.

3.2 Algorithm and training dynamics analysis

We now design an ICRL algorithm to train the transformer to implement SARSA, as outlined in Algorithm 1. This procedure is inspired by Laskin et al. [2023]. At each iteration, trajectory data are sampled from the MDP, the corresponding SARSA update is computed as a target, and the transformer parameters are updated to minimize the discrepancy between the transformer's prediction and the target update. The empirical loss at each step approximates the population loss defined below:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_z \left[\frac{1}{2} \left\| \mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}(z)) - \mathbf{w}_{\text{SARSA}}(z) \right\|^2 \right], \tag{6}$$

Algorithm 1 IC-SARSA Transformer Training

Input: Transformer parameter $\boldsymbol{\theta}^{(0)} = (\boldsymbol{P}^{(0)}, \boldsymbol{V}^{(0)})$, trajectory window length n, number of frames per MDP T, number of MDPs K, ϵ-greedy parameter ϵ, learning rate η, and SARSA step size α
for $k = 0, \ldots, K-1$ **do**
  Sample a MDP $\mathcal{M}^{(k)} = (\mathcal{S}, \mathcal{A}, P^{(k)}, R^{(k)}, \gamma^{(k)})$, and generate corresponding feature map $\phi^{(k)} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. Initialize parameter $\mathbf{w}^{(0)}$ and state $s_0^{(0)}$ randomly
  for $t = 0, \ldots, T-1$ **do**
    Estimate state-action value function $Q^{(k)}_{\mathbf{w}^{(t)}}(s,a) = \mathbf{w}^{(t)\top}\boldsymbol{\phi}^{(k)}(s,a)$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$
    Sample a n-step trajectory $(s_0^{(t)}, a_0^{(t)}, r_1^{(t)}, s_1^{(t)}, a_1^{(t)}, \ldots, r_n^{(t)}, s_n^{(t)}, a_n^{(t)})$ from the MDP $\mathcal{M}^{(k)}$, according to ϵ-greedy policy derived from $Q^{(k)}_{\mathbf{w}^{(t)}}$
    Formulate $\mathbf{H}^{(t)}$ as in (4) with the sampled trajectory and current parameter $\mathbf{w}^{(t)}$
    Compute $\mathbf{w}^{(t+1)} = \mathrm{TF}_{\boldsymbol{\theta}^{(t)}}(\mathbf{H}^{(t)}) := \mathbf{w}^{(t)} + \frac{1}{n}\left[\mathbf{V}^{(t)}\mathbf{H}^{(t)}\mathbf{H}^{(t)\top}\mathbf{P}^{(t)}\mathbf{H}^{(t)}\right]_{-d:,-1}$
    Compute $\mathbf{w}^{(t+1)}_{\mathrm{SARSA}}$ according to (2)
    Compute loss $\hat{L}^{(t)} = \frac{1}{2}\|\mathbf{w}^{(t+1)} - \mathbf{w}^{(t+1)}_{\mathrm{SARSA}}\|^2$
    Update $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \hat{L}^{(t)}$
    Update $s_0^{(t+1)} \leftarrow s_n^{(t)}$, $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t+1)}_{\mathrm{SARSA}}$
  end for
  Update $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}^{(T)}$
end for
Return Trained transformer parameter $\boldsymbol{\theta}$

where $\mathbf{w}_{\mathrm{SARSA}}(z)$ is the target parameter given by the batch semi-gradient SARSA update (2). We now analyze the training dynamics under this loss objective.

We consider gradient flow, which captures the behavior of gradient descent with infinitesimal step size, and is described by the following ordinary differential equation (ODE):

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\nabla \mathcal{L}(\boldsymbol{\theta}(t)). \tag{7}$$

We first identify the invariant parameter groups during training.

Proposition 3.1 (Invariant parameter groups)**.** *Partition the attention parameters into four sub-blocks:*

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}, \tag{8}$$

where $\mathbf{P}_{11}, \mathbf{V}_{11} \in \mathbb{R}^{(2d+1)\times(2d+1)}$, $\mathbf{P}_{12}, \mathbf{V}_{12} \in \mathbb{R}^{(2d+1)\times(d+1)}$, $\mathbf{P}_{21}, \mathbf{V}_{21} \in \mathbb{R}^{(d+1)\times(2d+1)}$, *and* $\mathbf{P}_{22}, \mathbf{V}_{22} \in \mathbb{R}^{(d+1)\times(d+1)}$. *Let* $\bar{\mathbf{V}}_{21}$ *and* $\bar{\mathbf{V}}_{22}$ *denote the last* d *rows of* $\mathbf{V}_{21}$ *and* $\mathbf{V}_{22}$*, respectively. Under gradient flow* (7)*, the following parameters are invariant:*

(i) The blocks $\mathbf{P}_{11}, \mathbf{P}_{21}, \mathbf{V}_{11}, \mathbf{V}_{12}$ *and the first rows of* $\mathbf{V}_{21}, \mathbf{V}_{22}$ *do not affect the transformer output and remain unchanged throughout training.*
(ii) If $\mathbf{P}_{22}^{(0)} = \mathbf{0}$ *and* $\bar{\mathbf{V}}_{22}^{(0)} = \mathbf{0}$*, then* $\mathbf{P}_{22}^{(t)} = \mathbf{0}$ *and* $\bar{\mathbf{V}}_{22}^{(t)} = \mathbf{0}$ *for all* $t \geq 0$.

The proof is deferred to Appendix F. Throughout the remainder of this section, we restrict our analysis to initializations satisfying $\mathbf{P}_{22}^{(0)} = \mathbf{0}$ and $\bar{\mathbf{V}}_{22}^{(0)} = \mathbf{0}$. By Proposition 3.1(ii), this constraint is preserved by gradient flow, and combined with (i) it reduces the training dynamics on $(\mathbf{P}, \mathbf{V})$ to dynamics on the effective parameter $\boldsymbol{\theta}_{\mathrm{eff}} := (\mathbf{P}_{12}, \bar{\mathbf{V}}_{21}) \in \boldsymbol{\Theta}_{\mathrm{eff}}$, where $\boldsymbol{\Theta}_{\mathrm{eff}} = \mathbb{R}^{(2d+1)\times(d+1)} \times \mathbb{R}^{d\times(2d+1)}$ is the effective parameter space. From (5), the optimum structures of these sub-blocks are

$$\mathbf{P}_{12}^{\star} := \begin{bmatrix} \mathbf{0}_{d\times 1} & -\mathbf{I}_d \\ \mathbf{0}_{d\times 1} & \mathbf{I}_d \\ 1 & \mathbf{0}_{1\times d} \end{bmatrix} \in \mathbb{R}^{(2d+1)\times(d+1)}, \qquad \bar{\mathbf{V}}_{21}^{\star} := [\alpha \mathbf{I}_d \quad \mathbf{0}_{d\times d} \quad \mathbf{0}_{d\times 1}] \in \mathbb{R}^{d\times(2d+1)},$$

subject to the scaling freedom $(\mathbf{P}_{12}^{\star}, \bar{\mathbf{V}}_{21}^{\star}) \mapsto (c\mathbf{P}_{12}^{\star}, c^{-1}\bar{\mathbf{V}}_{21}^{\star})$ for any $c \neq 0$. Without loss of generality we restrict to $c > 0$ (the sign-flipped branch is handled by symmetry), and to a compact

sub-interval $I = [c_-, c_+] \subset (0, \infty)$ bounded away from the degenerate limits $c \to 0, \infty$. On this interval, we define the *optimal parameter manifold*

$$\mathfrak{M}_I := \left\{ \bar{\boldsymbol{\theta}}_{\text{eff}}(c) := (c\mathbf{P}_{12}^\star,\ c^{-1}\bar{\mathbf{V}}_{21}^\star) \ :\ c \in I \right\} \subset \boldsymbol{\Theta}_{\text{eff}}, \tag{9}$$

and the normal space at $\bar{\boldsymbol{\theta}}_{\text{eff}}(c)$ as

$$N_c\mathfrak{M}_I := \left\{ (\mathbf{U}, \mathbf{W}) \in \boldsymbol{\Theta}_{\text{eff}} : \langle \mathbf{U}, \mathbf{P}_{12}^\star \rangle_F - c^{-2}\langle \mathbf{W}, \bar{\mathbf{V}}_{21}^\star \rangle_F = 0 \right\}. \tag{10}$$

We now state the richness assumptions on the training-MDP distribution that ensure local convergence to the optimal parameter manifold (9).

Assumption 1 (Richness of training MDPs)**.** Let Π denote the class of behavior policies used to generate training trajectories. There exist positive constants $B_\phi, B_r, B_{\tilde{w}}, \kappa_{\tilde{w}}, \kappa_R, \kappa_q$ and $\rho \in [0, 1)$ such that, for every $\pi \in \Pi$:

(1) **Boundedness.** Almost surely, $\|\boldsymbol{\phi}(s, a)\| \le B_\phi$, $|r| \le B_r$, and $\|\tilde{\mathbf{w}}\| \le B_{\tilde{w}}$.
(2) **Parameter excitation.** $\mathbb{E}[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top] \succeq \kappa_{\tilde{w}}\mathbf{I}_{d+1}$.
(3) **Bellman-regressor excitation.** $\mathbb{E}[\mathbf{R}(z)^\top \mathbf{R}(z) \mid \mathbf{w}] \succeq \kappa_R \mathbf{I}_{2d+1}$ for every $\mathbf{w} \in \mathbb{R}^d$.
(4) **TD-target excitation.** $\mathbb{E}[\mathbf{b}(z)\mathbf{b}(z)^\top] \succeq \kappa_q \mathbf{I}_{2d+1}$.
(5) **No destructive cancellation.** For every $c \in I$ and $(\mathbf{U}, \mathbf{W}) \in N_c\mathfrak{M}_I$,

$$\mathbb{E}_z\langle \mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}},\ \mathbf{W}\mathbf{b}(z)\rangle \ge -\rho\sqrt{\mathbb{E}_z\|\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 \cdot \mathbb{E}_z\|\mathbf{W}\mathbf{b}(z)\|^2}.$$

At a high level, these are mild conditions on the training distribution. Conditions (1)–(2) can be enforced by construction: normalizing features and clipping rewards gives boundedness, and drawing $\mathbf{w}$ from an isotropic distribution gives parameter excitation. Conditions (3)–(5) are structural properties of the trajectory distribution that hold generically under a random feature map $\boldsymbol{\phi}$, random rewards, and ϵ-greedy exploration ($\epsilon > 0$). In Appendix J we discuss these conditions further and make the genericity claim quantitative: we develop a Monte Carlo procedure that estimates $(\kappa_{\tilde{w}}, \kappa_R, \kappa_q, \rho)$ directly from the training prompts and use it to check the conditions numerically for the MDP family of our experiments (Section 4).

Theorem 3.2 (Local convergence to the optimal parameter manifold)**.** *Under Assumption 1, let*

$$B_\Sigma := 2B_\phi^2 + B_r^2, \quad C_Q := \frac{1}{2}B_\Sigma B_{\tilde{w}},$$
$$m_0 := (1-\rho)\min\left\{c_+^{-2}\alpha^2\kappa_R\kappa_{\tilde{w}}, c_-^2\kappa_q\right\}, \quad M_0 := B_\Sigma^2 B_{\tilde{w}}^2\left(c_-^{-2}\alpha^2 + 2c_+^2\right).$$

Define the tube

$$\mathcal{T}_r(I) := \left\{ \bar{\boldsymbol{\theta}}_{\mathit{eff}}(c) + \Delta : c \in I,\ \Delta \in N_c\mathfrak{M}_I,\ \|\Delta\| < r \right\}, \tag{11}$$

where $0 < r < \frac{\sqrt{m_0}}{3C_Q}$. *Fix a compact subinterval* $I_0 \subset \operatorname{int}(I)$. *Let*

$$\eta_{I_0,r} := \operatorname{dist}\big(\mathfrak{M}_{I_0}, \boldsymbol{\Theta}_{\mathit{eff}} \setminus \mathcal{T}_r(I)\big), \tag{12}$$

Suppose the initial parameter $\boldsymbol{\theta}_{\mathit{eff}}(0)$ *satisfies the following condition:*

$$\operatorname{dist}(\boldsymbol{\theta}_{\mathit{eff}}(0), \mathfrak{M}_{I_0}) < \min\left(r, \frac{\eta_{I_0,r}}{1 + \frac{K_r\left(\sqrt{M_0} + C_Q r\right)}{\sqrt{2}\mu_r}} \right), \tag{13}$$

where $K_r := \sqrt{2}B_\Sigma B_{\tilde{w}}\left(\left(c_+\sqrt{2d+1} + r\right)^2 + \left(c_-^{-1}\alpha\sqrt{d} + r\right)^2\right)^{1/2}$. *Consider the gradient flow* (7) *with initial point* $\boldsymbol{\theta}_{\mathit{eff}}(0)$. *Then the gradient flow solution exists for all* $t \ge 0$, *remains in* $\mathcal{T}_r(I)$, *and satisfies*

$$\mathcal{L}(\boldsymbol{\theta}_{\mathit{eff}}(t)) \le e^{-2\mu_r t}\big(\mathcal{L}(\boldsymbol{\theta}_{\mathit{eff}}(0))\big),$$

where $\mu_r := \frac{(m_0 - 3C_Q\sqrt{m_0}r)^2}{\left(\sqrt{M_0} + C_Q r\right)^2}$. *Furthermore,* $\boldsymbol{\theta}_{\mathit{eff}}(t)$ *converges to a point* $\boldsymbol{\theta}_{\mathit{eff}}(\infty) \in \mathfrak{M}_I$, *and*

$$\|\boldsymbol{\theta}_{\mathit{eff}}(t) - \boldsymbol{\theta}_{\mathit{eff}}(\infty)\| \le \frac{K_r}{\mu_r}\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\mathit{eff}}(0))}\, e^{-\mu_r t}.$$

Theorem 3.2 establishes that, when initialized sufficiently close to the optimal manifold $\mathfrak{M}_I$, gradient flow provably converges to a point on that manifold at an exponential rate governed by μ_r. The key insight is that under Assumption 1 on the training MDPs, the loss function exhibits local contractivity: the manifold itself is a level set, but perturbations perpendicular to it decay exponentially. To briefly summarize our proof strategy, we first define the tube (11), which specifies the region around $\mathfrak{M}_I$ where a local Polyak–Łojasiewicz (PL) inequality holds (Lemma C.2). Next, the buffer distance (12) quantifies the slackness between the inner manifold patch $\mathfrak{M}_{I_0}$ and the boundary of the tube, providing the margin a trajectory can travel before risking exit. The initialization condition (13) then bounds the cumulative gradient-flow displacement by this margin $\eta_{I_0,r}$, which ensures that the trajectory never leaves the tube and the PL inequality remains active for all $t \geq 0$. The complete proof is deferred to Appendix G.

3.3 Transformers can implement Actor-Critic

Having established both existence and convergence for the SARSA-based ICRL implementation, we now ask whether an analogous construction extends to the actor-critic method (3), in which the transformer must jointly update a value function and a parameterized policy. To accommodate this, we construct the input prompt as follows:

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{\phi}_V(s_0) & \cdots & \boldsymbol{\phi}_V(s_{n-1}) & \mathbf{0} \\ \gamma\boldsymbol{\phi}_V(s_1) & \cdots & \gamma\boldsymbol{\phi}_V(s_n) & \mathbf{0} \\ r_1 & \cdots & r_n & 0 \\ \gamma^0\mathbf{g}_{\boldsymbol{\lambda}}(s_0,a_0) & \cdots & \gamma^{n-1}\mathbf{g}_{\boldsymbol{\lambda}}(s_{n-1},a_{n-1}) & \mathbf{0} \\ 0 & \cdots & 0 & 1 \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\lambda} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{w} \end{bmatrix} \in \mathbb{R}^{D\times(n+1)}, \tag{14}$$

where $D = 3d + 2m + 2$. Compared to the SARSA prompt (4), the construction in (14) additionally encodes the discounted score function terms $\gamma^i\mathbf{g}_{\boldsymbol{\lambda}}(s_i,a_i)$ and the current actor parameter $\boldsymbol{\lambda}$. The following theorem shows that a single transformer block can still exactly implement the batch actor-critic update.

Theorem 3.3 (Implementing Actor-Critic with a transformer block)**.** *Let the output projection function be $f_{read}(\cdot) = \cdot_{-(d+m):,-1}$, which extracts the last $(d+m)$ entries of the final column. Given the input prompt* (14)*, for any $\alpha, \beta > 0$ there exist parameters $\boldsymbol{\theta}^\star = \{(c\mathbf{P}^\star, c^{-1}\mathbf{V}^\star)\} \subseteq \mathbb{R}^{D\times D}$ with $c \neq 0$ and*

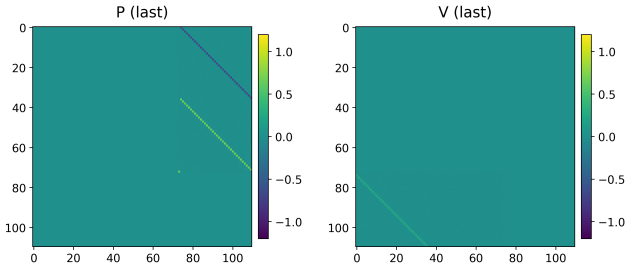
$$\mathbf{P}^\star = \begin{bmatrix} *_{(2d+m+1)\times(2d+m+1)} & \begin{bmatrix} \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & -\mathbf{I}_{d\times d} \\ \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & \mathbf{I}_{d\times d} \\ 1 & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times d} \\ \mathbf{0}_{m\times 1} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times d} \end{bmatrix} \\ *_{(d+m+1)\times(2d+m+1)} & \mathbf{0}_{(d+m+1)\times(d+m+1)} \end{bmatrix},$$
$$\mathbf{V}^\star = \begin{bmatrix} *_{(2d+m+1)\times(2d+m+1)} & *_{(2d+m+1)\times(d+m+1)} \\ \begin{bmatrix} *_{1\times d} & *_{1\times d} & *_{1\times 1} & *_{1\times m} \\ \mathbf{0}_{m\times d} & \mathbf{0}_{m\times d} & \mathbf{0}_{m\times 1} & \alpha\mathbf{I}_{m\times m} \\ \beta\mathbf{I}_{d\times d} & \mathbf{0}_{d\times d} & \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} \end{bmatrix} & \begin{bmatrix} *_{1\times(d+m+1)} \\ \mathbf{0}_{m\times(d+m+1)} \\ \mathbf{0}_{d\times(d+m+1)} \end{bmatrix} \end{bmatrix}, \tag{15}$$

such that the transformer $\mathsf{TF}_{\boldsymbol{\theta}^\star}$ directly implements the batch actor-critic update (3)*.*

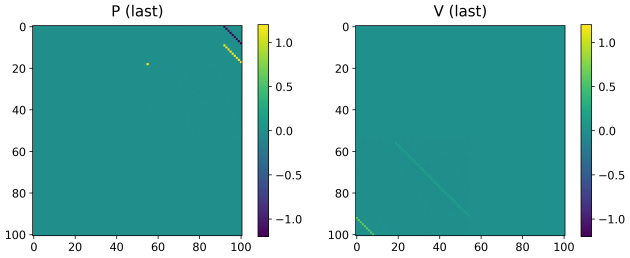
The proof of Theorem 3.3 is provided in Appendix H. Similar to SARSA, we design a training procedure to learn the transformer parameters $\boldsymbol{\theta}$ in Algorithm 2 in Appendix B. The convergence analysis of Section 3.2 extends to the actor-critic setting: in Appendix I, we show that under analogous richness conditions on the training-MDP distribution (Assumption 2), the gradient flow of the population imitation loss converges locally and exponentially to the optimal parameter manifold induced by (15) (Theorem I.1). Empirical validation of convergence is provided in Section 4.

4 Experiments

We empirically demonstrate that gradient-based training of a single linear self-attention block converges to the optimal parameter manifold characterized in Theorem 3.1 and Theorem 3.3, using Algorithm 1 and Algorithm 2 respectively. Furthermore, we evaluate the control performance of the trained transformers, showing their ability to perform policy improvement on unseen MDPs.



(a) SARSA: learned $\mathbf{P}$. (b) SARSA: learned $\mathbf{V}$.



(c) Actor-Critic: learned $\mathbf{P}$. (d) Actor-Critic: learned $\mathbf{V}$.

Figure 2: Final learned $\mathbf{P}$ and $\mathbf{V}$ matrices after training, for the SARSA transformer (top row) and the Actor-Critic transformer (bottom row). In both settings, the non-zero structure matches the theoretical constructions $(\mathbf{P}^\star, \mathbf{V}^\star)$ in (5) and (15).

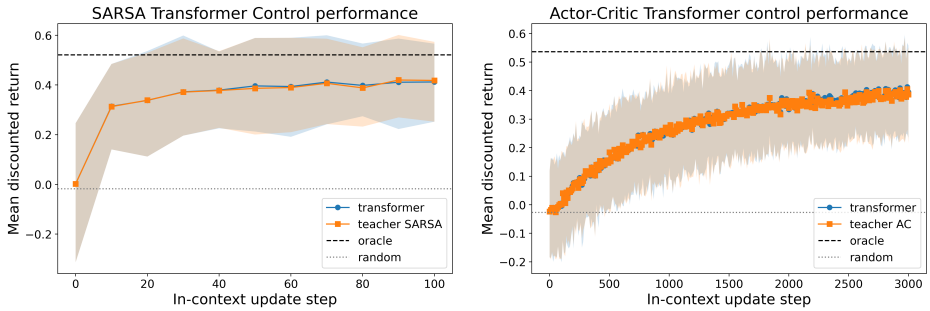
4.1 Training

Common Training Setup. Both experiments share the same training MDP family. Each MDP has $n_S = 9$ states and $n_A = 4$ actions, with discount $\gamma = 0.5$. For each MDP, the initial-state distribution P_0 and transition kernel $P(\cdot \mid s, a)$ are drawn independently from $\mathrm{Dirichlet}(1, \ldots, 1)$ (uniform over the probability simplex on n_S states), and the rewards $r(a, s') \sim \mathrm{Unif}(-1, 1)$ are sampled i.i.d. per (a, s'). Trajectories of window length $n = 20$ are rolled out with horizon $T = 1000$. In both experiments, the in-context learner is a one-layer transformer with corresponding embedding dimension D. Following Proposition 3.1, the inert blocks $\mathbf{P}_{11}, \mathbf{P}_{21}, \mathbf{V}_{11}, \mathbf{V}_{12}, \mathbf{P}_{22}, \mathbf{V}_{22}$ are initialized to zero, while the effective sub-blocks $(\mathbf{P}_{12}, \mathbf{V}_{21})$ are initialized with Xavier-normal entries (gain 0.1). Training is performed over $K = 5{,}000$ independently sampled MDPs using Adam optimizer [Kingma and Ba, 2015] with initial learning rate $\eta = 10^{-3}$ and exponential decay by a factor of 0.99 every 10 MDPs.

SARSA experiment specifications. For each MDP, the state–action feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ ($d = n_S n_A = 36$) and the initial weight $\mathbf{w}^{(0)} \in \mathbb{R}^d$ have entries drawn i.i.d. from $\mathrm{Unif}(-1, 1)$. The behavior policy is ϵ-greedy ($\epsilon = 0.1$) with $Q(s, a) = \boldsymbol{\phi}(s, a)^\top \mathbf{w}$, and the SARSA target $\mathbf{w}_{\mathrm{SARSA}}$ is computed from the trajectory window with step size $\alpha = 0.2$. The transformer uses embedding dimension $D = 3d + 2 = 110$.

Actor-Critic experiment specifications. For each MDP, the policy feature $\boldsymbol{\phi}_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ ($m = 36$), value feature $\boldsymbol{\phi}_V : \mathcal{S} \to \mathbb{R}^d$ ($d = 9$), and initial weights $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^m$, $\mathbf{w}^{(0)} \in \mathbb{R}^d$ are drawn i.i.d. from $\mathrm{Unif}(-1, 1)$. The behavior policy follows $\pi(a \mid s) \propto \exp(\boldsymbol{\phi}_\pi(s, a)^\top \boldsymbol{\lambda})$, with $\epsilon = 0.1$ – random exploration. At each step, the teacher applies a one-step critic update to $\mathbf{w}$ (step size $\beta = 0.8$) and an actor update to $\boldsymbol{\lambda}$ (step size $\alpha = 0.2$). The transformer uses embedding dimension $D = 3d + 2m + 2 = 101$.

Results. In both experiments, the training loss decays approximately exponentially to zero (see Appendix K.1). Figure 2 shows the learned $\mathbf{P}$ and $\mathbf{V}$ matrices: the upper-right block of $\mathbf{P}$ and the relevant rows of the lower-left block of $\mathbf{V}$ recover the structure of $(\mathbf{P}^\star, \mathbf{V}^\star)$ in (5) and (15) (up to scaling), while other blocks remain near zero. This confirms that the ICRL training algorithms 1 and 2 drive the parameters toward the optimal manifold.



(a) SARSA transformer.

(b) Actor-Critic transformer.

Figure 3: Closed-loop control performance on 100 held-out random MDPs versus in-context update step t. Curves show the per-MDP return averaged across MDPs (shaded: 25–75% interquartile band). The transformer curve closely tracks the teacher curve in both settings, indicating that the trained model approximately implements the analytical update rule at deployment.

4.2 Evaluation

Beyond parameter-level fidelity, we evaluate whether the trained transformer, when deployed as an in-context update rule on unseen MDPs, induces high-return policies. We randomly sample $K = 100$ test MDPs with corresponding features, disjoint from the training set. For each MDP, we run closed-loop in-context updates: at each step, the prompt $\mathbf{H}^{(t)}$ is fed to the transformer to produce $\mathbf{w}^{(t+1)}$ (or $(\boldsymbol{\lambda}^{(t+1)}, \mathbf{w}^{(t+1)})$ in the actor–critic setting), which defines the behavior policy used to generate the next trajectory window. Every 10 update steps, we freeze the iterate and estimate the discounted return via Monte Carlo using 32 trajectories of length 50, with initial state $s_0 \sim P_0^{(k)}$.

We compare against analytical teacher updates (SARSA (2) and actor–critic (3)) executed in the same closed loop, the oracle greedy policy under $Q^\star$ obtained by exact value iteration on $(P^{(k)}, R^{(k)})$, and a uniform random policy. Figure 3 reports the average return across MDPs. In both settings, the transformer closely tracks the teacher: performance improves rapidly from near-random initialization and approaches the oracle as t increases. Crucially, this consistency persists across long in-context horizons, indicating that the transformer faithfully implements the teacher update rule rather than merely mimicking short-horizon behavior.

Notably, the remaining gap to the oracle reflects limitations of the linear function approximation rather than imperfect imitation, since the same gap is also observed for the analytical teacher. These results show that convergence to the optimal parameter manifold (Theorem 3.2) translates into strong policy-level performance, with the trained transformer accurately reproducing the analytical update rules at deployment.

4.3 Additional Experiments

Additional details and experiments can be found in Appendix K. We summarize some of the findings below.

- The cosine similarity between the learned effective blocks $(\mathbf{P}_{12}, \bar{\mathbf{V}}_{21})$ and the theoretical constructions increases monotonically with the training budget and reaches 1.000 at $K = 5{,}000$ in both settings (Appendix K.2).
- The predicted block structure also emerges when training on a non-tabular, continuous-state MountainCar task family (Appendix K.3).
- For nonlinear attention, a two-head ReLU-attention transformer matches the teacher, whereas single-head ReLU and softmax variants fall short (Appendix K.4).
- The trained linear-attention transformer also generalizes beyond the training distribution: deployed on an unseen structured cliff-walk task, the same frozen SARSA transformer matches the analytical teacher to two decimals and finds the shortest safe path around the cliff (Appendix K.5).
- We ablate the actor-critic prompt by replacing the pre-computed score-function terms with raw policy features; the transformer no longer recovers the exact actor update (Appendix K.6).

5 Conclusion

In this work, we demonstrate that transformers can provably implement ICRL algorithms that perform policy improvement. Through explicit constructions, we show that a single linear-attention block can realize batch SARSA and actor-critic updates. Furthermore, we provide the first convergence guarantee in the ICRL literature, proving that under suitable richness conditions on the training MDPs, gradient-flow training to mimic the teacher update converges locally to the optimal parameter manifold. Our empirical results validate these theoretical findings, showing that trained transformers recover the predicted parameter structures and achieve strong control performance on unseen MDPs. These results help bridge the gap between the theoretical understanding and practical implementation of ICRL, suggesting that the transformer architecture naturally favors learning and implementing reinforcement-learning algorithmic structures. Overall, this work advances our understanding of how transformers perform RL algorithms in context, with implications for both AI safety and the design of more capable learning systems.

5.1 Limitations

Our work has several limitations. First, our convergence guarantee is local and requires initialization near the optimal manifold; understanding the global optimization landscape is an important direction for future work. Second, our theoretical results focus on linear attention; extending the convergence analysis to nonlinear attention mechanisms such as ReLU and softmax remains open. Finally, our analysis focuses on deterministic gradient flow in a simplified setting, leaving extensions to stochastic optimization and more complex transformer architectures for future study.

Acknowledgments and Disclosure of Funding

This work was supported in part by National Science Foundation under grants CCF-2232907 and ECCS-2448268. The authors declare no competing interests.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of Neural Information Processing Systems*, Virtual, December 2020. URL https://arxiv.org/abs/2005.14165.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In *Proceedings of Neural Information Processing Systems*, New Orleans, LA, USA, December 2022. URL https://arxiv.org/abs/2208.01066.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning, 2024. URL https://arxiv.org/abs/2301.00234.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Neural Information Processing Systems*, Long Beach, CA, USA, December 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. URL https://arxiv.org/abs/2303.08774.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2212.07677.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LziniAXEI9.

Arvind Mahankali, Tatsunori B. Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.03576.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=liMSqUuVg9.

Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context. In *Journal of Machine Learning Research*, 2024. URL https://arxiv.org/abs/2306.09927.

Haodong Liang, Krishna Balasubramanian, and Lifeng Lai. Transformers handle endogeneity in in-context linear regression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QfhU3ZC2g1.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.08566.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2502.07978.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dCYBAGQXLo.

Ethan Brooks, Logan Walls, Richard L. Lewis, and Satinder Singh. Large language models can implement policy iteration. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2210.03821.

Sharath Chandra Raparthy, Eric Hambro, Robert Kirk, Mikael Henaff, and Roberta Raileanu. Learning to solve new sequential decision-making tasks with in-context learning. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=JB3H6LS1PY.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pj06mxCXPl.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=a7APmM4B9d.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1273–1286. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/099fe6b0b444c23836c4a5d07346082b-Paper.pdf.

Jake Grigsby, Linxi Fan, and Yuke Zhu. AMAGO: Scalable in-context reinforcement learning for adaptive agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=M6XWoEdmwf.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=OWPzhVqIux.

Zixuan Xie, Xinyu Liu, Claire Chen, Shuze Daniel Liu, Rohan Chandra, and Shangtong Zhang. Beyond linear attention: Softmax transformers implement in-context reinforcement learning, 2026. URL https://arxiv.org/abs/2605.07333.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers, 2023. URL https://arxiv.org/abs/2310.05249.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality, 2024. URL https://arxiv.org/abs/2402.19442.

Yuandong Tian, Yiping Wang, Beidi Chen, and Simon Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.16380.

Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jNM4imlHZv.

A Additional Related Works

In this appendix, we discuss additional works that are related to our paper.

In-context learning theory for supervised tasks. Garg et al. [2022] initiate this discussion on simple function classes, showing empirically that a transformer trained from scratch can in-context learn linear regression, two-layer neural networks, and decision trees. A subsequent line of mechanistic work then asks *how* such behavior arises. von Oswald et al. [2023] and Akyürek et al. [2023] argue that transformers implement gradient descent or closed-form least squares on the in-context examples, and Ahn et al. [2023] and Mahankali et al. [2024] prove that, on linear regression, the optimal one-layer linear self-attention block is exactly one step of preconditioned gradient descent. Bai et al. [2023] broaden the picture by showing that transformers can implement a wide family of statistical estimators – including ridge regression, Lasso, and gradient descent on generalized linear models – and can even perform algorithm selection in context. Liang et al. [2025] further extend this line to in-context linear regression with endogenous covariates, constructing a multi-layer transformer that implements two-stage least squares and showing that it consistently estimates the structural parameter under standard instrumental-variable assumptions.

In-context reinforcement learning. The view of RL as in-context sequence modeling was popularized by the Decision Transformer [Chen et al., 2021] and the Trajectory Transformer [Janner et al., 2021], which cast policy improvement as autoregressive prediction over offline trajectories. Building on this paradigm, Laskin et al. [2023] introduce *algorithm distillation*, in which a causal transformer trained on RL learning histories reproduces the underlying learning algorithm at inference time, and Lee et al. [2023] show that supervised pretraining over offline trajectories suffices to elicit ICRL on new tasks. Subsequent empirical work demonstrates that large language models can in-context execute policy iteration on tabular MDPs [Brooks et al., 2023]; that pretraining on sufficiently diverse trajectories enables transformers to generalize to unseen sequential decision-making tasks purely from their context [Raparthy et al., 2023]; that scalable long-context architectures such as AMAGO learn adaptive policies in context across challenging environments [Grigsby et al., 2024]; and that LLMs can perform nontrivial exploration in bandit and small MDP problems even without external scaffolding [Krishnamurthy et al., 2024]. On the theoretical side, Lin et al. [2024] prove that supervised pretraining of transformers on offline trajectories yields near-optimal posterior-sampling policies, establishing sample-complexity guarantees for ICRL pretraining. The work most closely related to ours is Wang et al. [2025], which shows that a linear-attention block admits an invariant set under the expected gradient update of a value-prediction loss, thereby implementing policy evaluation via TD methods. Our work extends this line in two directions: from policy evaluation to policy improvement (SARSA and actor-critic), and from invariance to convergence of gradient training. In concurrent work, Xie et al. [2026] move beyond the linear-attention simplification and study in-context policy evaluation under standard softmax attention.

Training Dynamics. A complementary line of work studies the optimization landscape and training dynamics of attention itself, asking whether the in-context algorithms identified by mechanistic constructions are actually recoverable by gradient-based learning. Zhang et al. [2024] prove that gradient flow on a single linear self-attention block trained on random linear-regression tasks converges to the global minimum and recovers the preconditioned-gradient-descent solution predicted by Ahn et al. [2023], Mahankali et al. [2024]. Huang et al. [2023] extend such convergence guarantees to softmax attention under orthogonal feature assumptions, and Chen et al. [2024] analyze multi-head softmax attention in the in-context regression setting, identifying an emergent task-specialization phenomenon en route to the global minimizer. From a different angle, Tian et al. [2023] dissect token-level dynamics of one-layer attention during language pretraining, while Nichani et al. [2024] show that transformers can recover hidden causal structure via gradient descent on synthetic tasks. These results are confined to supervised regression-style objectives. Our work pushes the training-dynamics analysis into ICRL, where the loss is non-convex and the optima form a one-dimensional scaling manifold rather than an isolated point.

B Training Algorithm

B.1 Actor-Critic Transformer

Algorithm 2 Actor Critic Transformer Training

Input: Transformer parameter $\boldsymbol{\theta}^{(0)} = (\boldsymbol{P}^{(0)}, \boldsymbol{V}^{(0)})$, trajectory window length n, number of frames per MDP T, number of MDPs K, ϵ-greedy parameter ϵ, learning rate η, actor step size α, and critic step size β
for $k = 0, \ldots, K-1$ **do**
  Sample a MDP $\mathcal{M}^{(k)} = (\mathcal{S}, \mathcal{A}, P^{(k)}, R^{(k)}, \gamma)$, and generate corresponding feature map for policy $\boldsymbol{\phi}_\pi^{(k)} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$, and feature map for state value function $\boldsymbol{\phi}_V^{(k)} : \mathcal{S} \to \mathbb{R}^d$.Initialize parameters $\boldsymbol{\lambda}^{(0)}, \mathbf{w}^{(0)}$, and state $s_0^{(0)}$ randomly
  for $t = 0, \ldots, T-1$ **do**
    Calculate policy $\pi_{\boldsymbol{\lambda}^{(t)}}^{(k)}(a|s) = \frac{\exp\{\boldsymbol{\lambda}^{(t)\top} \boldsymbol{\phi}_\pi^{(k)}(s,a)\}}{\sum_b \exp\{\boldsymbol{\lambda}^{(t)\top} \boldsymbol{\phi}_\pi^{(k)}(s,b)\}}, \forall s \in \mathcal{S}, a \in \mathcal{A}$
    Estimate state value function $v_{\mathbf{w}^{(t)}}^{(k)}(s) = \mathbf{w}^{(t)\top} \boldsymbol{\phi}_V^{(k)}(s), \quad \forall s \in \mathcal{S}$
    Compute policy gradient $\mathbf{g}_{\boldsymbol{\lambda}^{(t)}}^{(k)}(s,a) = \boldsymbol{\phi}_\pi^{(k)}(s,a) - \sum_b \boldsymbol{\phi}_\pi^{(k)}(s,b) \pi_{\boldsymbol{\lambda}^{(t)}}^{(k)}(b|s), \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
    Sample n steps $(s_0^{(t)}, a_0^{(t)}, r_1^{(t)}, s_1^{(t)}, a_1^{(t)}, \ldots, r_n^{(t)}, s_n^{(t)}, a_n^{(t)})$ from the MDP $\mathcal{M}^{(k)}$, according to policy $\pi_{\boldsymbol{\lambda}^{(t)}}^{(k)}$ with ϵ-greedy exploration
    Formulate $\mathbf{H}^{(t)}$ as in (14) with the sampled trajectory and current parameters $\boldsymbol{\lambda}^{(t)}, \mathbf{w}^{(t)}$
    Let $\begin{bmatrix} \boldsymbol{\lambda}^{(t+1)} \\ \mathbf{w}^{(t+1)} \end{bmatrix} = \mathrm{TF}_{\boldsymbol{\theta}^{(t)}}(\mathbf{H}^{(t)}) := \begin{bmatrix} \boldsymbol{\lambda}^{(t)} \\ \mathbf{w}^{(t)} \end{bmatrix} + \frac{1}{n} \left[\mathbf{V}^{(t)} \mathbf{H}^{(t)} \mathbf{H}^{(t)\top} \mathbf{P}^{(t)} \mathbf{H}^{(t)} \right]_{-(d+m):,-1}$
    Compute $\mathbf{w}_{\mathrm{AC}}^{(t+1)}$ and $\boldsymbol{\lambda}_{\mathrm{AC}}^{(t+1)}$ according to (3)
    Compute loss $\hat{L}^{(t)} = \frac{1}{2} \|\mathbf{w}^{(t+1)} - \mathbf{w}_{\mathrm{AC}}^{(t+1)}\|^2 + \frac{1}{2} \|\boldsymbol{\lambda}^{(t+1)} - \boldsymbol{\lambda}_{\mathrm{AC}}^{(t+1)}\|^2$
    Update $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \hat{L}^{(t)}$
    Update $s_0^{(t+1)} \leftarrow s_n^{(t)}$, $\boldsymbol{\lambda}^{(t+1)} \leftarrow \boldsymbol{\lambda}_{\mathrm{AC}}^{(t+1)}$, $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}_{\mathrm{AC}}^{(t+1)}$
  end for
  Update $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}^{(T)}$
end for
Return Learned transformer parameter $\boldsymbol{\theta}$

Algorithm 2 extends Algorithm 1 to jointly learn both the value and policy parameters. At each iteration, trajectories are sampled from the MDP, and the corresponding actor-critic updates are computed as teacher signals for both the critic parameter $\mathbf{w}$ and the actor parameter $\boldsymbol{\lambda}$. The transformer parameters are then updated via gradient descent to match these targets. The empirical loss combines the squared errors for both parameter updates, encouraging the transformer to approximate the full actor-critic learning rule.

C Supporting Lemmas

Lemma C.1 (Transformer output decomposition)**.** Consider a transformer block with parameters $\boldsymbol{\theta} = \{(\mathbf{P}, \mathbf{V})\} \subseteq \mathbb{R}^{D\times D}$. We decompose the parameters into blocks as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix},$$

where $\mathbf{P}_{11}, \mathbf{V}_{11} \in \mathbb{R}^{(2d+1)\times(2d+1)}$, $\mathbf{P}_{12}, \mathbf{V}_{12} \in \mathbb{R}^{(2d+1)\times(d+1)}$, $\mathbf{P}_{21}, \mathbf{V}_{21} \in \mathbb{R}^{(d+1)\times(2d+1)}$, and $\mathbf{P}_{22}, \mathbf{V}_{22} \in \mathbb{R}^{(d+1)\times(d+1)}$. Let the output projection function be $f_{\text{read}}(\cdot) = \cdot_{-d:,-1}$, which extracts the last d elements of the last column. Denote $\tilde{\mathbf{w}} := [1 \quad \mathbf{w}]^\top$. Given formulated input prompt

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_0 & \cdots & \mathbf{x}_{n-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{w}} \end{bmatrix} \in \mathbb{R}^{D\times(n+1)},$$

the transformer output can then be written as:

$$\mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}) = \mathbf{w} + \bar{\mathbf{V}}_{21}\hat{\boldsymbol{\Sigma}}\mathbf{P}_{12}\tilde{\mathbf{w}} + \frac{1}{n}\bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top\mathbf{P}_{22}\tilde{\mathbf{w}}, \tag{16}$$

where $\hat{\boldsymbol{\Sigma}} := \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{x}_i\mathbf{x}_i^\top$, and $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}$ denote the last d rows of $\mathbf{V}_{21}$ and $\mathbf{V}_{22}$, respectively.

Lemma C.2 (Local Polyak–Łojasiewicz inequality)**.** Suppose Assumption 1 holds, and let the optimal manifold $\mathfrak{M}_I$, the constants $B_{\tilde{w}}, B_\Sigma, C_Q, m_0, M_0$, and the tube $\mathcal{T}_r(I)$ be as defined in Theorem 3.2. Then $\mathcal{L}$ attains its minimum value $\mathcal{L}^\star = 0$ on $\mathfrak{M}_I$, and for every $0 < r < \frac{\sqrt{m_0}}{3C_Q}$ the population loss satisfies

$$\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\|^2 \geq \mu_r\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big), \qquad \forall\, \boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I), \tag{17}$$

where

$$\mu_r := \frac{\big(m_0 - 3C_Q\sqrt{m_0}\, r\big)^2}{\big(\sqrt{M_0} + C_Q r\big)^2}.$$

D Proof of Supporting Lemmas

D.1 Proof of Lemma C.1

Proof. From (4), the input matrix is

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_0 & \cdots & \mathbf{x}_{n-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{\mathbf{w}} \end{bmatrix} \in \mathbb{R}^{D\times(n+1)}.$$

By the definition of the linear self-attention block in (1),

$$\mathbf{H}_{\text{out}} = \mathbf{H} + \frac{1}{n}(\mathbf{V}\mathbf{H})(\mathbf{H}^\top\mathbf{P}\mathbf{H}) \in \mathbb{R}^{D\times(n+1)},$$

and the readout extracts the last d entries of the last column of $\mathbf{H}_{\text{out}}$. Denote the last column of the input by

$$\mathbf{h}_{n+1} := \begin{bmatrix} \mathbf{0}_{2d+1} \\ \tilde{\mathbf{w}} \end{bmatrix} \in \mathbb{R}^D.$$

Then we have

$$\mathbf{H}_{\text{out}}[:, n+1] = \mathbf{h}_{n+1} + \frac{1}{n}(\mathbf{V}\mathbf{H})(\mathbf{H}^\top\mathbf{P}\mathbf{h}_{n+1}). \tag{18}$$

Using the block partition (8),

$$\mathbf{P}\mathbf{h}_{n+1} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{12}\tilde{\mathbf{w}} \\ \mathbf{P}_{22}\tilde{\mathbf{w}} \end{bmatrix}.$$

Hence

$$\begin{aligned}\big(\mathbf{H}^\top \mathbf{P}\mathbf{h}_{n+1}\big)_{i+1} &= \begin{bmatrix}\mathbf{x}_i^\top & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{12}\tilde{\mathbf{w}}\\ \mathbf{P}_{22}\tilde{\mathbf{w}}\end{bmatrix} = \mathbf{x}_i^\top \mathbf{P}_{12}\tilde{\mathbf{w}}, \quad i=0,\dots,n-1,\\ \big(\mathbf{H}^\top \mathbf{P}\mathbf{h}_{n+1}\big)_{n+1} &= \begin{bmatrix}\mathbf{0} & \tilde{\mathbf{w}}^\top\end{bmatrix}\begin{bmatrix}\mathbf{P}_{12}\tilde{\mathbf{w}}\\ \mathbf{P}_{22}\tilde{\mathbf{w}}\end{bmatrix} = \tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}.\end{aligned} \tag{19}$$

Similarly,

$$\begin{aligned}(\mathbf{V}\mathbf{H})[:, i+1] &= \mathbf{V}\begin{bmatrix}\mathbf{x}_i\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{V}_{11}\mathbf{x}_i\\ \mathbf{V}_{21}\mathbf{x}_i\end{bmatrix}, \quad i=0,\dots,n-1,\\ (\mathbf{V}\mathbf{H})[:, n+1] &= \mathbf{V}\begin{bmatrix}\mathbf{0}\\ \tilde{\mathbf{w}}\end{bmatrix} = \begin{bmatrix}\mathbf{V}_{12}\tilde{\mathbf{w}}\\ \mathbf{V}_{22}\tilde{\mathbf{w}}\end{bmatrix}.\end{aligned} \tag{20}$$

Expanding the matrix-vector product as a sum over columns and combining (19) and (20), we have

$$\begin{aligned}(\mathbf{V}\mathbf{H})\big(\mathbf{H}^\top \mathbf{P}\mathbf{h}_{n+1}\big) &= \sum_{i=0}^{n-1}\begin{bmatrix}\mathbf{V}_{11}\mathbf{x}_i\\ \mathbf{V}_{21}\mathbf{x}_i\end{bmatrix}\mathbf{x}_i^\top \mathbf{P}_{12}\tilde{\mathbf{w}} + \begin{bmatrix}\mathbf{V}_{12}\tilde{\mathbf{w}}\\ \mathbf{V}_{22}\tilde{\mathbf{w}}\end{bmatrix}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}\\ &= \begin{bmatrix}\mathbf{V}_{11}\big(\sum_{i=0}^{n-1}\mathbf{x}_i\mathbf{x}_i^\top\big)\mathbf{P}_{12}\tilde{\mathbf{w}} + \mathbf{V}_{12}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}\\ \mathbf{V}_{21}\big(\sum_{i=0}^{n-1}\mathbf{x}_i\mathbf{x}_i^\top\big)\mathbf{P}_{12}\tilde{\mathbf{w}} + \mathbf{V}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}\end{bmatrix}.\end{aligned}$$

Thus, rescaling by $\frac{1}{n}$, the lower $(d+1)$-dimensional block of (18) reads

$$\tilde{\mathbf{w}} + \mathbf{V}_{21}\hat{\mathbf{\Sigma}}\mathbf{P}_{12}\tilde{\mathbf{w}} + \frac{1}{n}\mathbf{V}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}.$$

Since $\tilde{\mathbf{w}} = [1, \mathbf{w}^\top]^\top$, the last d entries of $\tilde{\mathbf{w}}$ equal $\mathbf{w}$. The readout $f_{\text{read}}(\cdot) = \cdot_{-d:,-1}$ extracts the last d rows of the lower block, which selects the last d rows of $\mathbf{V}_{21}$ and $\mathbf{V}_{22}$, namely $\bar{\mathbf{V}}_{21}$ and $\bar{\mathbf{V}}_{22}$. Therefore,

$$\mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}) = \mathbf{w} + \bar{\mathbf{V}}_{21}\hat{\mathbf{\Sigma}}\mathbf{P}_{12}\tilde{\mathbf{w}} + \frac{1}{n}\bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}},$$

which completes the proof. □

D.2 Proof of Lemma C.2

Proof. For any effective parameter $\boldsymbol{\theta}_{\text{eff}} = (\mathbf{P}_{12}, \bar{\mathbf{V}}_{21})$, define the auxiliary predictor

$$f_{\boldsymbol{\theta}_{\text{eff}}}(z) := \bar{\mathbf{V}}_{21}\hat{\mathbf{\Sigma}}(z)\mathbf{P}_{12}\tilde{\mathbf{w}} \in \mathbb{R}^d.$$

By Theorem 3.1, $(\mathbf{P}_{12}^\star, \bar{\mathbf{V}}_{21}^\star)$ is a global minimizer of $\mathcal{L}$ with $\mathcal{L}^\star = 0$, and the scaling identity $f_{(c\mathbf{P}_{12}^\star, c^{-1}\bar{\mathbf{V}}_{21}^\star)} = f_{(\mathbf{P}_{12}^\star, \bar{\mathbf{V}}_{21}^\star)}$ for every $c > 0$ implies that every point of $\mathfrak{M}_I$ is a global minimizer. Consequently,

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star = \tfrac{1}{2}\mathbb{E}_z\big\|f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z)\big\|^2, \qquad \forall c \in I. \tag{21}$$

Throughout the proof, we denote $\Delta = (\mathbf{U}, \mathbf{W}) \in N_c\mathfrak{M}_I$ with $\|\Delta\| \le r$, where $\|\Delta\|^2 := \|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2$, and set $\boldsymbol{\theta}_{\text{eff}} := \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta$. Write $\|\cdot\|_{L^2} := \sqrt{\mathbb{E}_z\|\cdot\|^2}$.

Step 1: Decomposition of the prediction error. By the definition of $f_{\boldsymbol{\theta}_{\text{eff}}}$ together with $\boldsymbol{\theta}_{\text{eff}} = (c\mathbf{P}_{12}^\star + \mathbf{U},\ c^{-1}\bar{\mathbf{V}}_{21}^\star + \mathbf{W})$,

$$\begin{aligned}f_{\boldsymbol{\theta}_{\text{eff}}}(z) &= (c^{-1}\bar{\mathbf{V}}_{21}^\star + \mathbf{W})\hat{\mathbf{\Sigma}}(z)(c\mathbf{P}_{12}^\star + \mathbf{U})\tilde{\mathbf{w}},\\ f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) &= c^{-1}\bar{\mathbf{V}}_{21}^\star\hat{\mathbf{\Sigma}}(z)\, c\mathbf{P}_{12}^\star\tilde{\mathbf{w}} = \bar{\mathbf{V}}_{21}^\star\hat{\mathbf{\Sigma}}(z)\mathbf{P}_{12}^\star\tilde{\mathbf{w}}.\end{aligned}$$

Expanding the product and cancelling the matched leading term,

$$f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) = c^{-1}\bar{\mathbf{V}}_{21}^\star\hat{\mathbf{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}} + c\mathbf{W}\hat{\mathbf{\Sigma}}(z)\mathbf{P}_{12}^\star\tilde{\mathbf{w}} + \mathbf{W}\hat{\mathbf{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}}.$$

We now identify the first two terms with the trajectory statistics $\mathbf{R}(z)$ and $\mathbf{b}(z)$. Since $\bar{\mathbf{V}}_{21}^\star = [\alpha\mathbf{I}_d\ \mathbf{0}_{d\times d}\ \mathbf{0}_{d\times 1}]$, we have $\bar{\mathbf{V}}_{21}^\star\mathbf{x}_i = \alpha\boldsymbol{\phi}_i$, so

$$\bar{\mathbf{V}}_{21}^\star\hat{\mathbf{\Sigma}}(z) = \frac{1}{n}\sum_{i=0}^{n-1}(\bar{\mathbf{V}}_{21}^\star\mathbf{x}_i)\mathbf{x}_i^\top = \frac{\alpha}{n}\sum_{i=0}^{n-1}\boldsymbol{\phi}_i\mathbf{x}_i^\top = \alpha\,\mathbf{R}(z).$$

Similarly, $\mathbf{P}_{12}^{\star}\tilde{\mathbf{w}} = [-\mathbf{w};\, \mathbf{w};\, 1]$, hence $\mathbf{x}_i^\top \mathbf{P}_{12}^{\star}\tilde{\mathbf{w}} = -\boldsymbol{\phi}_i^\top \mathbf{w} + \gamma(\boldsymbol{\phi}_i^+)^\top \mathbf{w} + r_{i+1} = \delta_i$, so

$$\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}^{\star}\tilde{\mathbf{w}} = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}_i(\mathbf{x}_i^\top \mathbf{P}_{12}^{\star}\tilde{\mathbf{w}}) = \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}_i\delta_i = \mathbf{b}(z).$$

Substituting these identities,

$$\begin{aligned} f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) &= c^{-1}\alpha\, \mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}} + c\, \mathbf{W}\mathbf{b}(z) + \mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}} \\ &=: \mathbf{X}(z) + \mathbf{Y}(z), \end{aligned} \tag{22}$$

where

$$\mathbf{X}(z) := c^{-1}\alpha\, \mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}} + c\, \mathbf{W}\mathbf{b}(z), \qquad \mathbf{Y}(z) := \mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}}.$$

Step 2: Lower bound on $\|\mathbf{X}\|_{L^2}$. We first lower-bound the two summands of $\mathbf{X}(z)$ separately. Expanding the squared norm and conditioning on $\tilde{\mathbf{w}}$,

$$\begin{aligned} \mathbb{E}_z\|\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 &= \mathbb{E}_z\big[\tilde{\mathbf{w}}^\top \mathbf{U}^\top \mathbf{R}(z)^\top \mathbf{R}(z)\, \mathbf{U}\tilde{\mathbf{w}}\big] \\ &= \mathbb{E}_z\big[\tilde{\mathbf{w}}^\top \mathbf{U}^\top \mathbb{E}[\mathbf{R}(z)^\top \mathbf{R}(z) \mid \tilde{\mathbf{w}}]\, \mathbf{U}\tilde{\mathbf{w}}\big] \\ &\geq \kappa_R\, \mathbb{E}_z\|\mathbf{U}\tilde{\mathbf{w}}\|^2 \\ &= \kappa_R\, \mathrm{tr}\big(\mathbf{U}\, \mathbb{E}_z[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top]\, \mathbf{U}^\top\big) \\ &\geq \kappa_R\kappa_{\tilde{w}}\|\mathbf{U}\|_F^2, \end{aligned} \tag{23}$$

where the third line uses Assumption 1 (3), and the last inequality uses Assumption 1 (2). Similarly,

$$\begin{aligned} \mathbb{E}_z\|\mathbf{W}\mathbf{b}(z)\|^2 &= \mathbb{E}_z\big[\mathbf{b}(z)^\top \mathbf{W}^\top \mathbf{W}\mathbf{b}(z)\big] \\ &= \mathbb{E}_z\big[\mathrm{tr}\big(\mathbf{W}\mathbf{b}(z)\mathbf{b}(z)^\top \mathbf{W}^\top\big)\big] \\ &= \mathrm{tr}\big(\mathbf{W}\, \mathbb{E}_z[\mathbf{b}(z)\mathbf{b}(z)^\top]\, \mathbf{W}^\top\big) \\ &\geq \kappa_q\, \mathrm{tr}(\mathbf{W}\mathbf{W}^\top) \\ &= \kappa_q\|\mathbf{W}\|_F^2, \end{aligned} \tag{24}$$

where the fourth line uses Assumption 1 (4). Set $A := \mathbb{E}_z\|c^{-1}\alpha\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2$ and $B := \mathbb{E}_z\|c\mathbf{W}\mathbf{b}(z)\|^2$. Assumption 1 (5) lower-bounds the cross term as

$$2\mathbb{E}_z\langle c^{-1}\alpha\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}},\, c\mathbf{W}\mathbf{b}(z)\rangle \geq -2\rho\sqrt{AB},$$

so expanding $\|\mathbf{X}(z)\|^2$ and applying the cross-term bound,

$$\begin{aligned} \mathbb{E}_z\|\mathbf{X}(z)\|^2 &= A + B + 2\mathbb{E}_z\big\langle c^{-1}\alpha\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}},\, c\mathbf{W}\mathbf{b}(z)\big\rangle \\ &\geq A + B - 2\rho\sqrt{AB} \\ &= (1-\rho)(A+B) + \rho(\sqrt{A} - \sqrt{B})^2 \\ &\geq (1-\rho)(A+B), \end{aligned} \tag{25}$$

The two summands A, B are bounded below using the bounds (23) (24) and $|c| \in [c_-, c_+]$:

$$\begin{aligned} A &= c^{-2}\alpha^2\, \mathbb{E}_z\|\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 \geq c^{-2}\alpha^2\kappa_R\kappa_{\tilde{w}}\|\mathbf{U}\|_F^2 \geq c_+^{-2}\alpha^2\kappa_R\kappa_{\tilde{w}}\|\mathbf{U}\|_F^2, \\ B &= c^2\, \mathbb{E}_z\|\mathbf{W}\mathbf{b}(z)\|^2 \geq c^2\kappa_q\|\mathbf{W}\|_F^2 \geq c_-^2\kappa_q\|\mathbf{W}\|_F^2. \end{aligned}$$

Plugging into (25), with $\|\Delta\|^2 = \|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2$,

$$\begin{aligned} \mathbb{E}_z\|\mathbf{X}(z)\|^2 &\geq (1-\rho)\big(c_+^{-2}\alpha^2\kappa_R\kappa_{\tilde{w}}\|\mathbf{U}\|_F^2 + c_-^2\kappa_q\|\mathbf{W}\|_F^2\big) \\ &\geq (1-\rho)\min\big\{c_+^{-2}\alpha^2\kappa_R\kappa_{\tilde{w}},\, c_-^2\kappa_q\big\}\big(\|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2\big) \\ &= m_0\|\Delta\|^2. \end{aligned}$$

Hence

$$\|\mathbf{X}\|_{L^2} \geq \sqrt{m_0}\, \|\Delta\|. \tag{26}$$

Step 3: Upper bounds on $\|\mathbf{X}\|_{L^2}$ and $\|\mathbf{Y}\|_{L^2}$. By Assumption 1 (1),

$$\|\mathbf{x}_i\|^2 = \|\boldsymbol{\phi}_i\|^2 + \gamma^2\|\boldsymbol{\phi}_i^+\|^2 + r_{i+1}^2 \le 2B_\phi^2 + B_r^2 = B_\Sigma,$$

hence, applying the triangle inequality to $\hat{\boldsymbol{\Sigma}}(z) = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{x}_i\mathbf{x}_i^\top$,

$$\|\hat{\boldsymbol{\Sigma}}(z)\|_{\mathrm{op}} \le \tfrac{1}{n}\sum_{i=0}^{n-1}\|\mathbf{x}_i\mathbf{x}_i^\top\|_{\mathrm{op}} = \tfrac{1}{n}\sum_{i=0}^{n-1}\|\mathbf{x}_i\|^2 \le B_\Sigma.$$

Writing $\boldsymbol{\phi}_i = \mathbf{S}\mathbf{x}_i$ with $\mathbf{S} = [\mathbf{I}_d,\, \mathbf{0}_{d\times d},\, \mathbf{0}_{d\times 1}] \in \mathbb{R}^{d\times(2d+1)}$, we have

$$\mathbf{R}(z) = \tfrac{1}{n}\sum_i \mathbf{S}\mathbf{x}_i\mathbf{x}_i^\top = \mathbf{S}\hat{\boldsymbol{\Sigma}}(z),$$

and $\mathbf{S}\mathbf{S}^\top = \mathbf{I}_d$ gives $\|\mathbf{S}\|_{\mathrm{op}} = 1$. Submultiplicativity then yields

$$\begin{aligned}\|\mathbf{R}(z)\|_{\mathrm{op}} &= \|\mathbf{S}\hat{\boldsymbol{\Sigma}}(z)\|_{\mathrm{op}}\\ &\le \|\mathbf{S}\|_{\mathrm{op}}\,\|\hat{\boldsymbol{\Sigma}}(z)\|_{\mathrm{op}}\\ &\le B_\Sigma.\end{aligned}$$

Also $\|\tilde{\mathbf{w}}\| \le B_{\tilde{w}}$. Since $\mathbf{P}_{12}^{\star\top}\mathbf{P}_{12}^\star = \mathrm{diag}(1,\, 2\mathbf{I}_d)$, $\|\mathbf{P}_{12}^\star\|_{\mathrm{op}} = \sqrt{2}$, hence

$$\begin{aligned}\|\mathbf{b}(z)\| &\le \|\mathbf{P}_{12}^\star\|_{\mathrm{op}}\|\hat{\boldsymbol{\Sigma}}(z)\|_{\mathrm{op}}\|\tilde{\mathbf{w}}\|\\ &\le \sqrt{2}\,B_\Sigma B_{\tilde{w}}.\end{aligned}$$

By the triangle inequality and submultiplicativity,

$$\begin{aligned}\|\mathbf{X}\|_{L^2} &\le \big\|c^{-1}\alpha\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\big\|_{L^2} + \big\|c\,\mathbf{W}\mathbf{b}(z)\big\|_{L^2}\\ &\le |c|^{-1}\alpha\,\|\mathbf{R}(z)\|_{\mathrm{op}}\|\mathbf{U}\|_{\mathrm{op}}\|\tilde{\mathbf{w}}\| + |c|\,\|\mathbf{W}\|_{\mathrm{op}}\|\mathbf{b}(z)\|\\ &\le c_-^{-1}\alpha B_\Sigma B_{\tilde{w}}\|\mathbf{U}\|_F + \sqrt{2}\,c_+ B_\Sigma B_{\tilde{w}}\|\mathbf{W}\|_F.\end{aligned}$$

Applying Cauchy–Schwarz to the right-hand side,

$$\|\mathbf{X}\|_{L^2} \le B_\Sigma B_{\tilde{w}}\sqrt{c_-^{-2}\alpha^2 + 2c_+^2}\,\sqrt{\|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2} \tag{27}$$

$$= \sqrt{M_0}\,\|\Delta\|. \tag{28}$$

For the quadratic term,

$$\begin{aligned}\|\mathbf{Y}(z)\| &= \|\mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}}\|\\ &\le \|\mathbf{W}\|_{\mathrm{op}}\|\hat{\boldsymbol{\Sigma}}(z)\|_{\mathrm{op}}\|\mathbf{U}\|_{\mathrm{op}}\|\tilde{\mathbf{w}}\|\\ &\le B_\Sigma B_{\tilde{w}}\|\mathbf{U}\|_F\|\mathbf{W}\|_F\\ &\le B_\Sigma B_{\tilde{w}}\big(\tfrac{1}{2}\|\mathbf{U}\|_F^2 + \tfrac{1}{2}\|\mathbf{W}\|_F^2\big)\\ &= C_Q\|\Delta\|^2.\end{aligned}$$

Thus

$$\|\mathbf{Y}\|_{L^2} \le C_Q\|\Delta\|^2. \tag{29}$$

Step 4: Excess-loss upper bound. By (21), (22), and (27)–(29),

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_{\mathrm{eff}}) - \mathcal{L}^\star &= \tfrac{1}{2}\|\mathbf{X}+\mathbf{Y}\|_{L^2}^2\\ &\le \tfrac{1}{2}\big(\sqrt{M_0}\,\|\Delta\| + C_Q\|\Delta\|^2\big)^2\\ &\le \tfrac{1}{2}\big(\sqrt{M_0} + C_Q r\big)^2\|\Delta\|^2,\end{aligned} \tag{30}$$

where the last line uses $\|\Delta\| \le r$.

Step 5: Directional derivative lower bound. Define

$$g(t) := \mathcal{L}(\bar{\boldsymbol{\theta}}_{\text{eff}}(c) + t\Delta).$$

By the chain rule,

$$g'(1) = \langle \nabla \mathcal{L}(\boldsymbol{\theta}_{\text{eff}}), \Delta \rangle,$$

so Cauchy–Schwarz gives

$$\|\nabla \mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\| \, \|\Delta\| \geq g'(1). \tag{31}$$

Thus, to establish the PL inequality, it suffices to lower-bound $g'(1)$ with respect to $\|\Delta\|^2$. We now compute $g'(1)$. Replacing $\Delta = (\mathbf{U}, \mathbf{W})$ by $t\Delta = (t\mathbf{U},\, t\mathbf{W})$ in the decomposition (22),

$$\begin{aligned} f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)+t\Delta}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) &= c^{-1}\alpha\, \mathbf{R}(z)(t\mathbf{U})\tilde{\mathbf{w}} + c\,(t\mathbf{W})\mathbf{b}(z) + (t\mathbf{W})\hat{\boldsymbol{\Sigma}}(z)(t\mathbf{U})\tilde{\mathbf{w}} \\ &= t\big[c^{-1}\alpha\, \mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}} + c\,\mathbf{W}\mathbf{b}(z)\big] + t^2\, \mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}} \\ &= t\mathbf{X}(z) + t^2\mathbf{Y}(z), \end{aligned}$$

hence

$$g(t) = \tfrac{1}{2}\mathbb{E}_z\|t\mathbf{X}(z) + t^2\mathbf{Y}(z)\|^2.$$

Differentiating and evaluating at $t = 1$,

$$\begin{aligned} g'(1) &= \|\mathbf{X}\|_{L^2}^2 + 3\mathbb{E}_z\langle \mathbf{X}(z), \mathbf{Y}(z)\rangle + 2\|\mathbf{Y}\|_{L^2}^2 \\ &\geq \|\mathbf{X}\|_{L^2}^2 - 3\|\mathbf{X}\|_{L^2}\|\mathbf{Y}\|_{L^2}. \end{aligned}$$

Set $u := \|\mathbf{X}\|_{L^2}$ and $v := \|\mathbf{Y}\|_{L^2}$. The hypothesis $r < \sqrt{m_0}/(3C_Q)$ together with (26) and (29) gives

$$u \geq \sqrt{m_0}\,\|\Delta\| > 3C_Q\|\Delta\|^2 \geq 3v,$$

so $u \mapsto u^2 - 3uv$ is increasing at u. Substituting the lower bound on $\mathbf{u}$ (26) and upper bound on $\mathbf{v}$ (29),

$$\begin{aligned} g'(1) &\geq u^2 - 3uv \\ &\geq m_0\|\Delta\|^2 - 3C_Q\sqrt{m_0}\,\|\Delta\|^3 \\ &\geq m_0\|\Delta\|^2 - 3C_Q\sqrt{m_0}r\,\|\Delta\|^2 \\ &= \big(m_0 - 3C_Q\sqrt{m_0}\,r\big)\|\Delta\|^2. \end{aligned} \tag{32}$$

Step 6: PL inequality. Combining (31) with (32),

$$\|\nabla \mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\| \, \|\Delta\| \geq g'(1) \geq \big(m_0 - 3C_Q\sqrt{m_0}\,r\big)\|\Delta\|^2.$$

Squaring (and using (30) to convert $\|\Delta\|^2$ into the excess loss),

$$\tfrac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\|^2 \geq \frac{\big(m_0 - 3C_Q\sqrt{m_0}\,r\big)^2}{\big(\sqrt{M_0} + C_Q r\big)^2}\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big) = \mu_r\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big),$$

which is the claimed PL inequality on $\mathcal{T}_r(I)$. □

E Proof of Theorem 3.1

Proof. We construct $\boldsymbol{\theta}^\star = (\mathbf{P}^\star, \mathbf{V}^\star)$ explicitly in four steps: (i) reduce the readout to four blocks of $\boldsymbol{\theta}$; (ii) cancel the cubic-in-$\mathbf{w}$ contribution; (iii) match the affine contribution to the SARSA target; (iv) assemble $\boldsymbol{\theta}^\star$ and verify its scaling invariance.

Step 1: Reduction to $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}, \mathbf{P}_{12}, \mathbf{P}_{22}$. Partition $\mathbf{V}, \mathbf{P}$ as in (8), conformal with $D = (2d+1) + (d+1)$, and write $\tilde{\mathbf{w}} = [1; \mathbf{w}] \in \mathbb{R}^{d+1}$. Let $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}$ denote the last d rows of $\mathbf{V}_{21}, \mathbf{V}_{22}$, respectively. By Lemma C.1, the transformer update increment simplifies to

$$\Delta \mathbf{w}_{\mathrm{TF}}(\boldsymbol{\theta}) \;:=\; \mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}) - \mathbf{w} \;=\; \bar{\mathbf{V}}_{21}\hat{\boldsymbol{\Sigma}}\mathbf{P}_{12}\tilde{\mathbf{w}} + \tfrac{1}{n}\bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^{\top}\mathbf{P}_{22}\tilde{\mathbf{w}}, \tag{33}$$

where $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{(2d+1)\times(2d+1)}$ is the empirical second-moment matrix of the trajectory features, with the block decomposition

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \frac{\sum_{i=0}^{n-1} \boldsymbol{\phi}(s_i,a_i)\boldsymbol{\phi}(s_i,a_i)^{\top}}{n} & \frac{\sum_{i=0}^{n-1} \gamma\boldsymbol{\phi}(s_i,a_i)\boldsymbol{\phi}(s_{i+1},a_{i+1})^{\top}}{n} & \frac{\sum_{i=0}^{n-1} r_{i+1}\boldsymbol{\phi}(s_i,a_i)}{n} \\ \frac{\sum_{i=0}^{n-1} \gamma\boldsymbol{\phi}(s_{i+1},a_{i+1})\boldsymbol{\phi}(s_i,a_i)^{\top}}{n} & \frac{\sum_{i=0}^{n-1} \gamma^2\boldsymbol{\phi}(s_{i+1},a_{i+1})\boldsymbol{\phi}(s_{i+1},a_{i+1})^{\top}}{n} & \frac{\sum_{i=0}^{n-1} \gamma r_{i+1}\boldsymbol{\phi}(s_{i+1},a_{i+1})}{n} \\ \frac{\sum_{i=0}^{n-1} r_{i+1}\boldsymbol{\phi}(s_i,a_i)^{\top}}{n} & \frac{\sum_{i=0}^{n-1} \gamma r_{i+1}\boldsymbol{\phi}(s_{i+1},a_{i+1})^{\top}}{n} & \frac{\sum_{i=0}^{n-1} r_{i+1}^2}{n} \end{bmatrix}$$

$$:= \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{\phi\phi} & \gamma\hat{\boldsymbol{\Sigma}}_{\phi\phi_+} & \hat{\boldsymbol{\Sigma}}_{\phi r} \\ \gamma\hat{\boldsymbol{\Sigma}}_{\phi_+\phi} & \gamma^2\hat{\boldsymbol{\Sigma}}_{\phi_+\phi_+} & \gamma\hat{\boldsymbol{\Sigma}}_{\phi_+ r} \\ \hat{\boldsymbol{\Sigma}}_{r\phi} & \gamma\hat{\boldsymbol{\Sigma}}_{r\phi_+} & \hat{\boldsymbol{\Sigma}}_{rr} \end{bmatrix}.$$

In particular, the blocks $\mathbf{V}_{11}, \mathbf{V}_{12}, \mathbf{P}_{11}, \mathbf{P}_{21}$ and the first rows of $\mathbf{V}_{21}, \mathbf{V}_{22}$ do not enter the readout, so they are unconstrained.

Step 2: Cancel the cubic term. The batch SARSA update increment (2) can be written compactly as

$$\Delta \mathbf{w}_{\mathrm{SARSA}} = \alpha\big[\hat{\boldsymbol{\Sigma}}_{\phi r} + \gamma\hat{\boldsymbol{\Sigma}}_{\phi\phi_+}\mathbf{w} - \hat{\boldsymbol{\Sigma}}_{\phi\phi}\mathbf{w}\big], \tag{34}$$

which is affine in $\mathbf{w}$. The first term in (33) is also affine via $\tilde{\mathbf{w}} = [1; \mathbf{w}]$, while the second is cubic in $\mathbf{w}$. Enforcing $\Delta \mathbf{w}_{\mathrm{TF}} \equiv \Delta \mathbf{w}_{\mathrm{SARSA}}$ for every $\mathbf{w}$ therefore requires the cubic term to vanish, for which we set

$$\bar{\mathbf{V}}_{22}^{\star} = \mathbf{0}, \quad \mathbf{P}_{22}^{\star} = \mathbf{0}.$$

Step 3: Match the affine term. It remains to choose $\bar{\mathbf{V}}_{21}^{\star}, \mathbf{P}_{12}^{\star}$ so that

$$\bar{\mathbf{V}}_{21}^{\star}\hat{\boldsymbol{\Sigma}}\mathbf{P}_{12}^{\star}\tilde{\mathbf{w}} = \alpha\big[\hat{\boldsymbol{\Sigma}}_{\phi r} + \gamma\hat{\boldsymbol{\Sigma}}_{\phi\phi_+}\mathbf{w} - \hat{\boldsymbol{\Sigma}}_{\phi\phi}\mathbf{w}\big].$$

Take $\bar{\mathbf{V}}_{21}^{\star}$ to extract the first d rows of $\hat{\boldsymbol{\Sigma}}$ with prefactor α:

$$\bar{\mathbf{V}}_{21}^{\star} = \big[\alpha\mathbf{I}_d,\ \mathbf{0}_{d\times d},\ \mathbf{0}_{d\times 1}\big] \quad \Rightarrow \quad \bar{\mathbf{V}}_{21}^{\star}\hat{\boldsymbol{\Sigma}} = \alpha\big[\hat{\boldsymbol{\Sigma}}_{\phi\phi},\ \gamma\hat{\boldsymbol{\Sigma}}_{\phi\phi_+},\ \hat{\boldsymbol{\Sigma}}_{\phi r}\big].$$

The matching condition then reduces to $\mathbf{P}_{12}^{\star}\tilde{\mathbf{w}} = [-\mathbf{w};\ \mathbf{w};\ 1]$, which holds for every $\mathbf{w} \in \mathbb{R}^d$ when

$$\mathbf{P}_{12}^{\star} = \begin{bmatrix} \mathbf{0}_{d\times 1} & -\mathbf{I}_d \\ \mathbf{0}_{d\times 1} & \mathbf{I}_d \\ 1 & \mathbf{0}_{1\times d} \end{bmatrix}.$$

Step 4: Assembly and scaling. Leave the unconstrained blocks identified in Step 1 as arbitrary entries (denoted $*$). This yields the structure $(\mathbf{P}^{\star}, \mathbf{V}^{\star})$ in (5), and by construction

$$\mathsf{TF}_{\boldsymbol{\theta}^{\star}}(\mathbf{H}) = \mathbf{w} + \Delta \mathbf{w}_{\mathrm{SARSA}}.$$

Finally, both terms in (33) are bilinear in $(\mathbf{V}, \mathbf{P})$, so the rescaled pair $(c\mathbf{P}^{\star}, c^{-1}\mathbf{V}^{\star})$ produces the same readout for every $c \neq 0$, completing the construction. □

F Proof of Proposition 3.1

Proof. Let $\boldsymbol{\delta}(\boldsymbol{\theta}, z) := \mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}(z)) - \mathbf{w}_{\mathrm{SARSA}}(z)$ denote the per-sample residual, so that

$$\mathcal{L}(\boldsymbol{\theta}) = \tfrac{1}{2}\mathbb{E}_z\|\boldsymbol{\delta}(\boldsymbol{\theta}, z)\|^2.$$

By Lemma C.1,

$$\boldsymbol{\delta}(\boldsymbol{\theta}, z) = \bar{\mathbf{V}}_{21}\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}\tilde{\mathbf{w}}(z) + \tfrac{1}{n}\bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}(z)\tilde{\mathbf{w}}(z)^{\top}\mathbf{P}_{22}\tilde{\mathbf{w}}(z) + \big(\mathbf{w}(z) - \mathbf{w}_{\mathrm{SARSA}}(z)\big), \tag{35}$$

so $\mathcal{L}$ depends on $(\mathbf{V}, \mathbf{P})$ only through the four sub-blocks $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}, \mathbf{P}_{12}, \mathbf{P}_{22}$.

Part (i). Since $\mathbf{V}_{11}, \mathbf{V}_{12}, \mathbf{P}_{11}, \mathbf{P}_{21}$ do not appear in (35),

$$\nabla_{\mathbf{V}_{11}}\mathcal{L} = \nabla_{\mathbf{V}_{12}}\mathcal{L} = \nabla_{\mathbf{P}_{11}}\mathcal{L} = \nabla_{\mathbf{P}_{21}}\mathcal{L} = \mathbf{0},$$

so each is stationary under the gradient flow (7). For $\mathbf{V}_{21}$, the loss factors through $\bar{\mathbf{V}}_{21} = \mathbf{S}\mathbf{V}_{21}$ with $\mathbf{S} = [\mathbf{0}_{d\times 1}, \mathbf{I}_d]$, and the chain rule yields

$$\nabla_{\mathbf{V}_{21}}\mathcal{L} = \mathbf{S}^\top \nabla_{\bar{\mathbf{V}}_{21}}\mathcal{L}.$$

Since $\mathbf{S}^\top = [\mathbf{0}_{1\times d}; \mathbf{I}_d]$ has zero first row, so does $\nabla_{\mathbf{V}_{21}}\mathcal{L}$, and the first row of $\mathbf{V}_{21}$ remains unchanged. The same argument applies to $\mathbf{V}_{22}$.

Part (ii). A direct computation from (35) gives

$$\nabla_{\bar{\mathbf{V}}_{22}}\mathcal{L}(\boldsymbol{\theta}) = \tfrac{1}{n}\,\mathbb{E}_z\left[\boldsymbol{\delta}(\boldsymbol{\theta}, z)\big(\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top \mathbf{P}_{22}\tilde{\mathbf{w}}\big)^\top\right], \tag{36}$$

$$\nabla_{\mathbf{P}_{22}}\mathcal{L}(\boldsymbol{\theta}) = \tfrac{1}{n}\,\mathbb{E}_z\left[\big(\boldsymbol{\delta}(\boldsymbol{\theta}, z)^\top \bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}\big)\,\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top\right]. \tag{37}$$

Both (36) and (37) vanish whenever $\mathbf{P}_{22} = \mathbf{0}$ and $\bar{\mathbf{V}}_{22} = \mathbf{0}$. Hence the gradient flow (7) admits a trajectory in which $\mathbf{P}_{22}^{(t)} \equiv \mathbf{0}$ and $\bar{\mathbf{V}}_{22}^{(t)} \equiv \mathbf{0}$, with the remaining blocks evolving freely. □

G Proof of Theorem 3.2

Proof. We recall the local PL inequality (17) from Lemma C.2: for every $\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I)$,

$$\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\|^2 \geq \mu_r\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big).$$

We keep the $\mathcal{L}^\star$ term explicit for clarity, but it can be dropped since $\mathcal{L}^\star = 0$.

Step 1: Bounding the parameter distance to the optimum manifold. Fix any $\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I)$ and write its normal decomposition as

$$\boldsymbol{\theta}_{\text{eff}} = \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta, \quad \Delta = (\mathbf{U}, \mathbf{W}) \in N_c\mathfrak{M}_I, \quad \|\Delta\| < r.$$

As in the proof of Lemma C.2, the residual decomposes as

$$f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) = \mathbf{X}(z) + \mathbf{Y}(z),$$

where, by (22),

$$\mathbf{X}(z) := c^{-1}\alpha\,\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}} + c\,\mathbf{W}\mathbf{b}(z),$$
$$\mathbf{Y}(z) := \mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}}.$$

By (26),

$$\|\mathbf{X}\|_{L^2} \geq \sqrt{m_0}\,\|\Delta\|,$$

and by (29),

$$\|\mathbf{Y}\|_{L^2} \leq C_Q\,\|\Delta\|^2.$$

Therefore, applying the reverse triangle inequality,

$$\begin{aligned}\sqrt{2\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big)} &= \|\mathbf{X} + \mathbf{Y}\|_{L^2} \\ &\geq \|\mathbf{X}\|_{L^2} - \|\mathbf{Y}\|_{L^2} \\ &\geq \big(\sqrt{m_0} - C_Q\|\Delta\|\big)\|\Delta\| \\ &\geq \big(\sqrt{m_0} - C_Q r\big)\|\Delta\|,\end{aligned}$$

where the last inequality uses $\|\Delta\| < r$. Squaring both sides,

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star \geq \frac{1}{2}\big(\sqrt{m_0} - C_Q r\big)^2\|\Delta\|^2 = \lambda_r\,\|\Delta\|^2, \tag{38}$$

where we set

$$\lambda_r := \tfrac{1}{2}\big(\sqrt{m_0} - C_Q r\big)^2.$$

By definition of the tube (11), every point $\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I)$ admits a unique normal decomposition

$$\boldsymbol{\theta}_{\text{eff}} = \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta, \qquad \Delta \in N_c\mathfrak{M}_I,$$

so the normal displacement equals the distance to the manifold patch:

$$\|\Delta\| = \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I).$$

Combining this identity with (38), for all $\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I)$,

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star \geq \lambda_r \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I)^2,$$

and hence

$$\operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I) \leq \frac{1}{\sqrt{\lambda_r}} \sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star}. \tag{39}$$

Step 2: Local upper bound on the gradient norm. Define the prediction error

$$\begin{aligned}\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z) &:= f_{\mathbf{P}_{12},\bar{\mathbf{V}}_{21}}(z) - f_{\mathbf{P}_{12}^\star,\bar{\mathbf{V}}_{21}^\star}(z) \\ &= \bar{\mathbf{V}}_{21}\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}\tilde{\mathbf{w}} - \bar{\mathbf{V}}_{21}^\star\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}^\star\tilde{\mathbf{w}}.\end{aligned}$$

Then

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star = \frac{1}{2}\,\mathbb{E}_z\|\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\|^2,$$

so that

$$\mathbb{E}_z\|\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\|^2 = 2\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star\big).$$

Differentiating the bilinear predictor with respect to $\bar{\mathbf{V}}_{21}$ and $\mathbf{P}_{12}$ gives

$$\begin{aligned}\nabla_{\bar{\mathbf{V}}_{21}}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) &= \mathbb{E}_z\Big[\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\,\big(\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}\tilde{\mathbf{w}}\big)^\top\Big], \\ \nabla_{\mathbf{P}_{12}}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) &= \mathbb{E}_z\Big[\hat{\boldsymbol{\Sigma}}(z)^\top\,\bar{\mathbf{V}}_{21}^\top\,\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\,\tilde{\mathbf{w}}^\top\Big].\end{aligned}$$

For every $\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I)$, the triangle inequality together with $\|\Delta\| < r$ yields

$$\begin{aligned}\|\mathbf{P}_{12}\|_F &\leq c_+\|\mathbf{P}_{12}^\star\|_F + r \\ &= c_+\sqrt{2d+1} + r \\ &=: P_{\max,r},\end{aligned}$$

and

$$\begin{aligned}\|\bar{\mathbf{V}}_{21}\|_F &\leq c_-^{-1}\|\bar{\mathbf{V}}_{21}^\star\|_F + r \\ &= c_-^{-1}\alpha\sqrt{d} + r \\ &=: V_{\max,r}.\end{aligned}$$

Moreover, by Assumption 1 (1),

$$\|\hat{\boldsymbol{\Sigma}}(z)\|_{\text{op}} \leq B_\Sigma,$$

and

$$\|\tilde{\mathbf{w}}\| \leq B_{\tilde{w}}.$$

Therefore, by Cauchy–Schwarz and submultiplicativity,

$$\begin{aligned}\|\nabla_{\bar{\mathbf{V}}_{21}}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\|_F &\leq B_\Sigma B_{\tilde{w}}\,P_{\max,r}\,\big(\mathbb{E}_z\|\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\|^2\big)^{1/2} \\ &= \sqrt{2}\,B_\Sigma B_{\tilde{w}}\,P_{\max,r}\,\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star},\end{aligned}$$

and similarly,

$$\|\nabla_{\mathbf{P}_{12}}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\|_F \leq \sqrt{2}\,B_\Sigma B_{\tilde{w}}\,V_{\max,r}\,\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star}.$$

Combining the two estimates and recalling that

$$K_r = \sqrt{2}\,B_\Sigma B_{\tilde{w}}\sqrt{P_{\max,r}^2 + V_{\max,r}^2},$$

we obtain the gradient upper bound

$$\|\nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}})\| \leq K_r\,\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star}, \qquad \forall\,\boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I). \tag{40}$$

Step 3: Invariance of the tube. Let $\boldsymbol{\theta}_{\text{eff}}(t)$ denote the maximal solution of the gradient flow (7) starting from $\boldsymbol{\theta}_{\text{eff}}(0)$. Since $\nabla\mathcal{L}$ is locally Lipschitz, a unique local solution exists. Define the exit time

$$\tau := \inf\{t > 0 : \boldsymbol{\theta}_{\text{eff}}(t) \notin \mathcal{T}_r(I)\}.$$

We will show that $\tau = \infty$. For all $t < \tau$, the local PL inequality (17) holds, and along the gradient flow,

$$\begin{aligned}\frac{d}{dt}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) &= \langle \nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)), \dot{\boldsymbol{\theta}}_{\text{eff}}(t)\rangle \\ &= -\|\nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t))\|^2 \\ &\le -2\mu_r\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) - \mathcal{L}^\star\big).\end{aligned}$$

Since $\mathcal{L}^\star = 0$, Gronwall's inequality yields the loss decay bound

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) - \mathcal{L}^\star \le e^{-2\mu_r t}\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star\big), \qquad \forall\, t < \tau. \tag{41}$$

Combining (40) with (41), for every $t < \tau$,

$$\begin{aligned}\|\dot{\boldsymbol{\theta}}_{\text{eff}}(t)\| &= \|\nabla\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t))\| \\ &\le K_r\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) - \mathcal{L}^\star} \\ &\le K_r\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star}\, e^{-\mu_r t}.\end{aligned}$$

Integrating, for every $t < \tau$,

$$\begin{aligned}\|\boldsymbol{\theta}_{\text{eff}}(t) - \boldsymbol{\theta}_{\text{eff}}(0)\| &\le \int_0^t \|\dot{\boldsymbol{\theta}}_{\text{eff}}(s)\|\, ds \\ &\le K_r\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star}\int_0^t e^{-\mu_r s}\, ds \\ &\le \frac{K_r}{\mu_r}\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star}.\end{aligned} \tag{42}$$

We next upper-bound the initial excess loss in terms of the distance to $\mathfrak{M}_{I_0}$. From (30), since $\|\Delta\| = \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I)$ and $\mathfrak{M}_{I_0} \subset \mathfrak{M}_I$,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star &\le \tfrac{1}{2}\big(\sqrt{M_0} + C_Q r\big)^2\|\Delta\|^2 \\ &= \tfrac{1}{2}\big(\sqrt{M_0} + C_Q r\big)^2 \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I)^2 \\ &\le \tfrac{1}{2}\big(\sqrt{M_0} + C_Q r\big)^2 \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_{I_0})^2.\end{aligned} \tag{43}$$

Combining (42) and (43), for every $t < \tau$,

$$\begin{aligned}\operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(t), \mathfrak{M}_{I_0}) &\le \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(0), \mathfrak{M}_{I_0}) + \|\boldsymbol{\theta}_{\text{eff}}(t) - \boldsymbol{\theta}_{\text{eff}}(0)\| \\ &\le \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(0), \mathfrak{M}_{I_0}) + \frac{K_r}{\mu_r}\sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star} \\ &\le \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(0), \mathfrak{M}_{I_0})\left(1 + \frac{K_r\big(\sqrt{M_0} + C_Q r\big)}{\sqrt{2}\,\mu_r}\right) \\ &< \eta_{I_0,r},\end{aligned}$$

where the last inequality follows from the initialization condition (13). By the definition of $\eta_{I_0,r}$ in (12), this implies

$$\boldsymbol{\theta}_{\text{eff}}(t) \in \mathcal{T}_r(I), \qquad \forall\, t < \tau.$$

Hence the solution cannot exit $\mathcal{T}_r(I)$ at any finite time, so $\tau = \infty$.

Step 4: Convergence to the optimum manifold. Since the trajectory is confined to the bounded neighborhood

$$\big\{\boldsymbol{\theta}_{\text{eff}} \in \boldsymbol{\Theta}_{\text{eff}} : \operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_{I_0}) \le \eta_{I_0,r}\big\},$$

no finite-time blow-up occurs and the gradient flow is well defined for all $t \ge 0$. The exponential loss decay (41) therefore holds globally:

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) - \mathcal{L}^\star \le e^{-2\mu_r t}\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star\big), \qquad \forall\, t \ge 0. \tag{44}$$

Combining (39) with (44),

$$\begin{aligned}\operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(t), \mathfrak{M}_I)^2 &\le \frac{1}{\lambda_r}\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(t)) - \mathcal{L}^\star\big) \\ &\le \frac{1}{\lambda_r}\, e^{-2\mu_r t}\big(\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star\big).\end{aligned} \tag{45}$$

It remains to show that the parameters themselves converge to a point on $\mathfrak{M}_I$. By the same argument as in (42), for any $0 \le t_1 < t_2$,

$$\begin{aligned}\|\boldsymbol{\theta}_{\text{eff}}(t_2) - \boldsymbol{\theta}_{\text{eff}}(t_1)\| &\le \int_{t_1}^{t_2} \|\dot{\boldsymbol{\theta}}_{\text{eff}}(s)\|\, ds \\ &\le K_r \sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star} \int_{t_1}^{t_2} e^{-\mu_r s}\, ds \\ &\le \frac{K_r}{\mu_r} \sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star}\, e^{-\mu_r t_1}.\end{aligned} \tag{46}$$

Hence $\boldsymbol{\theta}_{\text{eff}}(t)$ is a Cauchy curve as $t \to \infty$. By completeness of $\boldsymbol{\Theta}_{\text{eff}}$, there exists $\boldsymbol{\theta}_{\text{eff}}(\infty) \in \boldsymbol{\Theta}_{\text{eff}}$ such that

$$\lim_{t\to\infty} \boldsymbol{\theta}_{\text{eff}}(t) = \boldsymbol{\theta}_{\text{eff}}(\infty).$$

Letting $t_2 \to \infty$ in (46),

$$\|\boldsymbol{\theta}_{\text{eff}}(t) - \boldsymbol{\theta}_{\text{eff}}(\infty)\| \le \frac{K_r}{\mu_r} \sqrt{\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(0)) - \mathcal{L}^\star}\, e^{-\mu_r t}.$$

By continuity of $\mathcal{L}$ together with (44),

$$\mathcal{L}(\boldsymbol{\theta}_{\text{eff}}(\infty)) = \mathcal{L}^\star.$$

Moreover, by (45),

$$\operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}(t), \mathfrak{M}_I) \to 0 \qquad \text{as } t \to \infty,$$

and since $\mathfrak{M}_I$ is closed, we conclude that

$$\boldsymbol{\theta}_{\text{eff}}(\infty) \in \mathfrak{M}_I.$$

This completes the proof. □

H Proof of Theorem 3.3

Proof. The argument follows the same four-step structure as the proof of Theorem 3.1; only the dimensions and the affine target change.

Step 1: Reduction. The actor-critic prompt (14) has the same block-diagonal form as the SARSA prompt (4), except the trajectory columns $\mathbf{h}_{i+1} := [\mathbf{x}_i; \mathbf{0}]$, $i = 0, \dots, n-1$ occupy the upper $(2d + m + 1)$-block, while the parameter column $\mathbf{h}_{n+1} = [\mathbf{0}_{2d+m+1}; \tilde{\mathbf{w}}]$ occupies the lower $(d + m + 1)$-block, with

$$\mathbf{x}_i = \begin{bmatrix} \boldsymbol{\phi}_V(s_i) \\ \gamma \boldsymbol{\phi}_V(s_{i+1}) \\ r_{i+1} \\ \gamma^i \mathbf{g}_{\boldsymbol{\lambda}}(s_i, a_i) \end{bmatrix} \in \mathbb{R}^{2d+m+1}, \text{ and } \tilde{\mathbf{w}} = \begin{bmatrix} 1 \\ \boldsymbol{\lambda} \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{d+m+1}.$$

Partitioning $\mathbf{V}, \mathbf{P}$ conformally with $D = (2d+m+1)+(d+m+1)$ as in (8), and letting $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}$ now denote the *last* $d+m$ *rows* of $\mathbf{V}_{21}, \mathbf{V}_{22}$, the same block computation as in Lemma C.1 yields

$$\begin{bmatrix}\Delta\boldsymbol{\lambda}_{\mathsf{TF}}(\boldsymbol{\theta})\\ \Delta\mathbf{w}_{\mathsf{TF}}(\boldsymbol{\theta})\end{bmatrix} := \mathsf{TF}_{\boldsymbol{\theta}}(\mathbf{H}) - \begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{w}\end{bmatrix} \;=\; \bar{\mathbf{V}}_{21}\hat{\boldsymbol{\Sigma}}\mathbf{P}_{12}\tilde{\mathbf{w}} + \tfrac{1}{n}\bar{\mathbf{V}}_{22}\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top\mathbf{P}_{22}\tilde{\mathbf{w}}, \tag{47}$$

where $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{(2d+m+1)\times(2d+m+1)}$ is the empirical second-moment matrix of the trajectory features:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=0}^{n-1}\mathbf{x}_i\mathbf{x}_i^\top := \begin{bmatrix}\hat{\boldsymbol{\Sigma}}_{\phi_V\phi_V} & \gamma\hat{\boldsymbol{\Sigma}}_{\phi_V\phi_{V+}} & \hat{\boldsymbol{\Sigma}}_{\phi_V r} & \hat{\boldsymbol{\Sigma}}_{\phi_V g}\\ \gamma\hat{\boldsymbol{\Sigma}}_{\phi_{V+}\phi_V} & \gamma^2\hat{\boldsymbol{\Sigma}}_{\phi_{V+}\phi_{V+}} & \gamma\hat{\boldsymbol{\Sigma}}_{\phi_{V+}r} & \gamma\hat{\boldsymbol{\Sigma}}_{\phi_{V+}g}\\ \hat{\boldsymbol{\Sigma}}_{r\phi_V} & \gamma\hat{\boldsymbol{\Sigma}}_{r\phi_{V+}} & \hat{\boldsymbol{\Sigma}}_{rr} & \hat{\boldsymbol{\Sigma}}_{rg}\\ \hat{\boldsymbol{\Sigma}}_{g\phi_V} & \gamma\hat{\boldsymbol{\Sigma}}_{g\phi_{V+}} & \hat{\boldsymbol{\Sigma}}_{gr} & \hat{\boldsymbol{\Sigma}}_{gg}\end{bmatrix}.$$

The blocks $\mathbf{V}_{11}, \mathbf{V}_{12}, \mathbf{P}_{11}, \mathbf{P}_{21}$ and the first row of $\mathbf{V}_{21}, \mathbf{V}_{22}$ are unconstrained.

Step 2: Cancel the cubic term. The actor-critic update (3) can be written as

$$\begin{bmatrix}\Delta\boldsymbol{\lambda}_{\text{AC}}\\ \Delta\mathbf{w}_{\text{AC}}\end{bmatrix} = \begin{bmatrix}\alpha\big[\hat{\boldsymbol{\Sigma}}_{gr} + \gamma\hat{\boldsymbol{\Sigma}}_{g\phi_{V+}}\mathbf{w} - \hat{\boldsymbol{\Sigma}}_{g\phi_V}\mathbf{w}\big]\\ \beta\big[\hat{\boldsymbol{\Sigma}}_{\phi_V r} + \gamma\hat{\boldsymbol{\Sigma}}_{\phi_V\phi_{V+}}\mathbf{w} - \hat{\boldsymbol{\Sigma}}_{\phi_V\phi_V}\mathbf{w}\big]\end{bmatrix}, \tag{48}$$

which is affine in $(\boldsymbol{\lambda}, \mathbf{w})$. Setting $\bar{\mathbf{V}}^\star_{22} = \mathbf{P}^\star_{22} = \mathbf{0}$ removes the cubic term in (47).

Step 3: Match the affine term. We require $\bar{\mathbf{V}}^\star_{21}\hat{\boldsymbol{\Sigma}}\mathbf{P}^\star_{12}\tilde{\mathbf{w}} = \big[\Delta\boldsymbol{\lambda}_{\text{AC}};\, \Delta\mathbf{w}_{\text{AC}}\big]$ for every $(\boldsymbol{\lambda}, \mathbf{w})$. Take

$$\bar{\mathbf{V}}^\star_{21} = \begin{bmatrix}\mathbf{0}_{m\times d} & \mathbf{0}_{m\times d} & \mathbf{0}_{m\times 1} & \alpha\mathbf{I}_m\\ \beta\mathbf{I}_d & \mathbf{0}_{d\times d} & \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m}\end{bmatrix},$$

which extracts the bottom m rows (prefactor by α), and the top d rows (prefactor by β) of $\hat{\boldsymbol{\Sigma}}$:

$$\bar{\mathbf{V}}^\star_{21}\hat{\boldsymbol{\Sigma}} = \begin{bmatrix}\alpha\hat{\boldsymbol{\Sigma}}_{g\phi_V} & \alpha\gamma\hat{\boldsymbol{\Sigma}}_{g\phi_{V+}} & \alpha\hat{\boldsymbol{\Sigma}}_{gr} & \alpha\hat{\boldsymbol{\Sigma}}_{gg}\\ \beta\hat{\boldsymbol{\Sigma}}_{\phi_V\phi_V} & \beta\gamma\hat{\boldsymbol{\Sigma}}_{\phi_V\phi_{V+}} & \beta\hat{\boldsymbol{\Sigma}}_{\phi_V r} & \beta\hat{\boldsymbol{\Sigma}}_{\phi_V g}\end{bmatrix}.$$

The matching condition then reduces to $\mathbf{P}^\star_{12}\tilde{\mathbf{w}} = [-\mathbf{w};\, \mathbf{w};\, 1;\, \mathbf{0}_m]$. With $\tilde{\mathbf{w}} = [1; \boldsymbol{\lambda}; \mathbf{w}]$, this holds for every $(\boldsymbol{\lambda}, \mathbf{w})$ when

$$\mathbf{P}^\star_{12} = \begin{bmatrix}\mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & -\mathbf{I}_d\\ \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & \mathbf{I}_d\\ 1 & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times d}\\ \mathbf{0}_{m\times 1} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times d}\end{bmatrix}.$$

Step 4: Assembly and scaling. Leaving the unconstrained blocks as $*$, we recover $(\mathbf{P}^\star, \mathbf{V}^\star)$ as displayed in (15). Bilinearity of (47) in $(\mathbf{V}, \mathbf{P})$ gives the same scaling invariance under $(\mathbf{P}^\star, \mathbf{V}^\star) \mapsto (c\mathbf{P}^\star, c^{-1}\mathbf{V}^\star)$ for any $c \neq 0$, completing the construction. □

I Convergence Analysis for the Actor-Critic Transformer

In this appendix, we extend the training-dynamics analysis of Section 3.2 to the actor-critic transformer and establish a local convergence guarantee (Theorem I.1) analogous to Theorem 3.2. With a slight abuse of notation, the quantities $\mathbf{x}_i$, $\tilde{\mathbf{w}}$, δ_i, $\hat{\boldsymbol{\Sigma}}(z)$, $\mathbf{R}(z)$, $\mathbf{b}(z)$, as well as the constants $B_\phi, B_r, B_{\tilde{w}}, \kappa_{\tilde{w}}, \kappa_R, \kappa_q, \rho, B_\Sigma, C_Q, m_0, M_0, K_r, \lambda_r, \mu_r$, are redefined below in their actor-critic form; within this appendix they supersede the SARSA quantities of the same name.

I.1 Setup and invariant parameter groups

A training sample is now $z = (\mathcal{M}, \boldsymbol{\phi}_\pi, \boldsymbol{\phi}_V, \boldsymbol{\lambda}, \mathbf{w}, \tau)$, where $\mathcal{M}$ is the training MDP, $\boldsymbol{\phi}_\pi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^m$ and $\boldsymbol{\phi}_V : \mathcal{S} \to \mathbb{R}^d$ are the policy and value feature maps, $(\boldsymbol{\lambda}, \mathbf{w})$ are the current actor and critic parameters defining the behavior policy $\pi_{\boldsymbol{\lambda}}$, and $\tau = (s_0, a_0, r_1, \ldots, s_n, a_n)$ is the resulting n-step trajectory. For $i = 0, \ldots, n-1$, define

$$\boldsymbol{\phi}_{V,i} := \boldsymbol{\phi}_V(s_i), \qquad \mathbf{g}_i := \mathbf{g}_{\boldsymbol{\lambda}}(s_i, a_i), \qquad \delta_i := r_{i+1} + \gamma\mathbf{w}^\top\boldsymbol{\phi}_{V,i+1} - \mathbf{w}^\top\boldsymbol{\phi}_{V,i},$$

$$\mathbf{x}_i := \begin{bmatrix}\boldsymbol{\phi}_{V,i}\\ \gamma\boldsymbol{\phi}_{V,i+1}\\ r_{i+1}\\ \gamma^i\mathbf{g}_i\end{bmatrix} \in \mathbb{R}^{2d+m+1}, \qquad \boldsymbol{\psi}_i := \begin{bmatrix}\gamma^i\mathbf{g}_i\\ \boldsymbol{\phi}_{V,i}\end{bmatrix} \in \mathbb{R}^{m+d}, \qquad \tilde{\mathbf{w}} := \begin{bmatrix}1\\ \boldsymbol{\lambda}\\ \mathbf{w}\end{bmatrix} \in \mathbb{R}^{d+m+1},$$

together with the trajectory statistics and the step-size matrix

$$\hat{\mathbf{\Sigma}}(z) := \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}_i \mathbf{x}_i^\top, \qquad \mathbf{R}(z) := \frac{1}{n}\sum_{i=0}^{n-1} \boldsymbol{\psi}_i \mathbf{x}_i^\top, \qquad \mathbf{b}(z) := \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{x}_i\, \delta_i, \qquad \mathbf{D} := \begin{bmatrix} \alpha \mathbf{I}_m & \mathbf{0} \\ \mathbf{0} & \beta \mathbf{I}_d \end{bmatrix}.$$

Mimicking the teacher update (3) as in Algorithm 2, the population loss is

$$\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}) := \mathbb{E}_z \left[\frac{1}{2} \left\| \mathsf{TF}_{\boldsymbol{\theta}}\big(\mathbf{H}(z)\big) - \begin{bmatrix} \boldsymbol{\lambda}_{\text{AC}}(z) \\ \mathbf{w}_{\text{AC}}(z) \end{bmatrix} \right\|^2 \right], \tag{49}$$

where $\mathbf{H}(z)$ is the prompt (14) and $(\boldsymbol{\lambda}_{\text{AC}}(z), \mathbf{w}_{\text{AC}}(z))$ is the batch actor-critic target (3). We study the corresponding gradient flow

$$\frac{d\boldsymbol{\theta}(t)}{dt} = -\nabla \mathcal{L}_{\text{AC}}(\boldsymbol{\theta}(t)). \tag{50}$$

The invariance structure of Proposition 3.1 carries over unchanged.

Proposition I.1 (Invariant parameter groups, actor-critic)**.** *Partition $\mathbf{P}, \mathbf{V}$ into sub-blocks as in* (8)*, conformally with $D = (2d+m+1)+(d+m+1)$, and let $\bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22}$ denote the last $d+m$ rows of $\mathbf{V}_{21}, \mathbf{V}_{22}$. Under the gradient flow* (50)*, the following parameters are invariant:*

(i) The blocks $\mathbf{P}_{11}, \mathbf{P}_{21}, \mathbf{V}_{11}, \mathbf{V}_{12}$ and the first rows of $\mathbf{V}_{21}, \mathbf{V}_{22}$ do not affect the transformer output and remain unchanged throughout training.

(ii) If $\mathbf{P}_{22}^{(0)} = \mathbf{0}$ and $\bar{\mathbf{V}}_{22}^{(0)} = \mathbf{0}$, then $\mathbf{P}_{22}^{(t)} = \mathbf{0}$ and $\bar{\mathbf{V}}_{22}^{(t)} = \mathbf{0}$ for all $t \geq 0$.

Proof. By the readout decomposition (47), the per-sample residual

$$\boldsymbol{\delta}(\boldsymbol{\theta}, z) = \bar{\mathbf{V}}_{21} \hat{\mathbf{\Sigma}}(z) \mathbf{P}_{12} \tilde{\mathbf{w}} + \tfrac{1}{n} \bar{\mathbf{V}}_{22} \tilde{\mathbf{w}} \tilde{\mathbf{w}}^\top \mathbf{P}_{22} \tilde{\mathbf{w}} + \begin{bmatrix} \boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{AC}}(z) \\ \mathbf{w} - \mathbf{w}_{\text{AC}}(z) \end{bmatrix}$$

has exactly the same bilinear-plus-cubic dependence on the sub-blocks $(\mathbf{P}_{12}, \mathbf{P}_{22}, \bar{\mathbf{V}}_{21}, \bar{\mathbf{V}}_{22})$ as (35), so the proof of Proposition 3.1 applies verbatim. □

As in Section 3.2, we henceforth restrict to initializations with $\mathbf{P}_{22}^{(0)} = \mathbf{0}$ and $\bar{\mathbf{V}}_{22}^{(0)} = \mathbf{0}$, which reduces the training dynamics to the effective parameter $\boldsymbol{\theta}_{\text{eff}} := (\mathbf{P}_{12}, \bar{\mathbf{V}}_{21}) \in \boldsymbol{\Theta}_{\text{eff}} := \mathbb{R}^{(2d+m+1)\times(d+m+1)} \times \mathbb{R}^{(d+m)\times(2d+m+1)}$, with the associated prediction map

$$f_{\boldsymbol{\theta}_{\text{eff}}}(z) := \bar{\mathbf{V}}_{21} \hat{\mathbf{\Sigma}}(z) \mathbf{P}_{12} \tilde{\mathbf{w}} \in \mathbb{R}^{d+m}.$$

I.2 Optimal parameter manifold

From Step 3 of the proof of Theorem 3.3, the optimal effective sub-blocks are

$$\mathbf{P}_{12}^\star = \begin{bmatrix} \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & -\mathbf{I}_d \\ \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} & \mathbf{I}_d \\ 1 & \mathbf{0}_{1\times m} & \mathbf{0}_{1\times d} \\ \mathbf{0}_{m\times 1} & \mathbf{0}_{m\times m} & \mathbf{0}_{m\times d} \end{bmatrix}, \qquad \bar{\mathbf{V}}_{21}^\star = \begin{bmatrix} \mathbf{0}_{m\times d} & \mathbf{0}_{m\times d} & \mathbf{0}_{m\times 1} & \alpha \mathbf{I}_m \\ \beta \mathbf{I}_d & \mathbf{0}_{d\times d} & \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} \end{bmatrix},$$

which satisfy the two key identities

$$\bar{\mathbf{V}}_{21}^\star \hat{\mathbf{\Sigma}}(z) = \mathbf{D}\mathbf{R}(z), \qquad \hat{\mathbf{\Sigma}}(z) \mathbf{P}_{12}^\star \tilde{\mathbf{w}} = \mathbf{b}(z), \tag{51}$$

since $\bar{\mathbf{V}}_{21}^\star \mathbf{x}_i = \mathbf{D}\boldsymbol{\psi}_i$ and $\mathbf{x}_i^\top \mathbf{P}_{12}^\star \tilde{\mathbf{w}} = -\boldsymbol{\phi}_{V,i}^\top \mathbf{w} + \gamma \boldsymbol{\phi}_{V,i+1}^\top \mathbf{w} + r_{i+1} = \delta_i$. Consequently,

$$f_{(\mathbf{P}_{12}^\star, \bar{\mathbf{V}}_{21}^\star)}(z) = \mathbf{D} \cdot \frac{1}{n}\sum_{i=0}^{n-1} \boldsymbol{\psi}_i \delta_i = \begin{bmatrix} \boldsymbol{\lambda}_{\text{AC}}(z) - \boldsymbol{\lambda} \\ \mathbf{w}_{\text{AC}}(z) - \mathbf{w} \end{bmatrix},$$

so $\mathcal{L}_{\text{AC}}$ attains its minimum value $\mathcal{L}_{\text{AC}}^\star = 0$ at $(\mathbf{P}_{12}^\star, \bar{\mathbf{V}}_{21}^\star)$ and, by bilinearity of the predictor, at every rescaling $(c\mathbf{P}_{12}^\star, c^{-1}\bar{\mathbf{V}}_{21}^\star)$ with $c \neq 0$. As in Section 3.2, we restrict without loss of generality to $c > 0$ and a compact interval $I = [c_-, c_+] \subset (0, \infty)$, and define the optimal parameter manifold and its normal spaces

$$\begin{aligned} \mathfrak{M}_I &:= \big\{ \bar{\boldsymbol{\theta}}_{\text{eff}}(c) := (c\mathbf{P}_{12}^\star,\, c^{-1}\bar{\mathbf{V}}_{21}^\star) \,:\, c \in I \big\}, \\ N_c\mathfrak{M}_I &:= \big\{ (\mathbf{U}, \mathbf{W}) \in \boldsymbol{\Theta}_{\text{eff}} : \langle \mathbf{U}, \mathbf{P}_{12}^\star \rangle_F - c^{-2} \langle \mathbf{W}, \bar{\mathbf{V}}_{21}^\star \rangle_F = 0 \big\}. \end{aligned} \tag{52}$$

I.3 Richness assumptions and main result

Assumption 2 (Richness of training MDPs, actor-critic)**.** Let Π denote the class of behavior policies used to generate training trajectories. There exist positive constants $B_\phi, B_g, B_r, B_{\tilde{w}}, \kappa_{\tilde{w}}, \kappa_R, \kappa_q$ and $\rho \in [0, 1)$ such that, for every $\pi \in \Pi$:

(1) **Boundedness.** Almost surely, $\|\boldsymbol{\phi}_V(s)\| \le B_\phi$, $\|\mathbf{g}_{\boldsymbol{\lambda}}(s,a)\| \le B_g$, $|r| \le B_r$, and $\|\tilde{\mathbf{w}}\| \le B_{\tilde{w}}$.
(2) **Parameter excitation.** $\mathbb{E}[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top] \succeq \kappa_{\tilde{w}} \mathbf{I}_{d+m+1}$.
(3) **Bellman-regressor excitation.** $\mathbb{E}[\mathbf{R}(z)^\top \mathbf{R}(z) \mid \tilde{\mathbf{w}}] \succeq \kappa_R \mathbf{I}_{2d+m+1}$ for every $\tilde{\mathbf{w}}$.
(4) **TD-target excitation.** $\mathbb{E}[\mathbf{b}(z)\mathbf{b}(z)^\top] \succeq \kappa_q \mathbf{I}_{2d+m+1}$.
(5) **No destructive cancellation.** For every $c \in I$ and $(\mathbf{U}, \mathbf{W}) \in N_c \mathfrak{M}_I$,

$$\mathbb{E}_z \langle \mathbf{D}\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}},\, \mathbf{W}\mathbf{b}(z) \rangle \ge -\rho \sqrt{\mathbb{E}_z \|\mathbf{D}\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 \cdot \mathbb{E}_z \|\mathbf{W}\mathbf{b}(z)\|^2}.$$

As in Assumption 1, conditions (1)–(2) can be enforced by construction; in particular, the score function is automatically bounded, $\|\mathbf{g}_{\boldsymbol{\lambda}}(s,a)\| \le 2 \max_{s,b} \|\boldsymbol{\phi}_\pi(s,b)\|$, so condition (1) holds whenever the policy features are normalized. Conditions (3)–(5) are structural excitation conditions on the trajectory distribution, analogous to those in the SARSA setting.

Define the constants

$$\alpha_\vee := \max\{\alpha, \beta\}, \qquad \alpha_\wedge := \min\{\alpha, \beta\}, \qquad B_\Sigma := 2B_\phi^2 + B_r^2 + B_g^2, \qquad C_Q := \frac{1}{2} B_\Sigma B_{\tilde{w}},$$
$$m_0 := (1-\rho) \min\{c_+^{-2} \alpha_\wedge^2 \kappa_R \kappa_{\tilde{w}},\, c_-^2 \kappa_q\}, \qquad M_0 := B_\Sigma^2 B_{\tilde{w}}^2 \big(c_-^{-2} \alpha_\vee^2 + 2c_+^2\big),$$

and, for $0 < r < \frac{\sqrt{m_0}}{3C_Q}$, the tube

$$\mathcal{T}_r(I) := \big\{ \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta : c \in I,\, \Delta \in N_c \mathfrak{M}_I,\, \|\Delta\| < r \big\}. \tag{53}$$

We first state the actor-critic counterpart of the local PL inequality (Lemma C.2).

Lemma I.1 (Local Polyak–Łojasiewicz inequality, actor-critic)**.** Suppose Assumption 2 holds. Then $\mathcal{L}_{\text{AC}}$ attains its minimum value $\mathcal{L}_{\text{AC}}^\star = 0$ on $\mathfrak{M}_I$, and for every $0 < r < \frac{\sqrt{m_0}}{3C_Q}$,

$$\frac{1}{2} \|\nabla \mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}})\|^2 \ge \mu_r \big(\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}_{\text{AC}}^\star\big), \qquad \forall\, \boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I), \tag{54}$$

where $\mu_r := \frac{(m_0 - 3C_Q\sqrt{m_0}\, r)^2}{\big(\sqrt{M_0} + C_Q r\big)^2}$.

We can now state the main result of this appendix.

Theorem I.1 (Local convergence to the optimal parameter manifold, actor-critic)**.** *Suppose Assumption 2 holds, and let $0 < r < \frac{\sqrt{m_0}}{3C_Q}$. Fix a compact subinterval $I_0 \subset \operatorname{int}(I)$ and let*

$$\eta_{I_0,r} := \operatorname{dist}\big(\mathfrak{M}_{I_0}, \boldsymbol{\Theta}_{\mathit{eff}} \setminus \mathcal{T}_r(I)\big). \tag{55}$$

Suppose the initial parameter $\boldsymbol{\theta}_{\mathit{eff}}(0)$ satisfies

$$\operatorname{dist}(\boldsymbol{\theta}_{\mathit{eff}}(0), \mathfrak{M}_{I_0}) < \min\left(r,\; \frac{\eta_{I_0,r}}{1 + \frac{K_r\big(\sqrt{M_0} + C_Q r\big)}{\sqrt{2}\mu_r}} \right), \tag{56}$$

where

$$K_r := \sqrt{2} B_\Sigma B_{\tilde{w}} \Big(\big(c_+ \sqrt{2d+1} + r\big)^2 + \big(c_-^{-1} \sqrt{\alpha^2 m + \beta^2 d} + r\big)^2 \Big)^{1/2}.$$

Consider the gradient flow (50) *restricted to the effective parameters, with initial point $\boldsymbol{\theta}_{\mathit{eff}}(0)$. Then the solution exists for all $t \ge 0$, remains in $\mathcal{T}_r(I)$, and satisfies*

$$\mathcal{L}_{\mathit{AC}}(\boldsymbol{\theta}_{\mathit{eff}}(t)) \le e^{-2\mu_r t}\, \mathcal{L}_{\mathit{AC}}(\boldsymbol{\theta}_{\mathit{eff}}(0)),$$

where μ_r is as in Lemma I.1. Furthermore, $\boldsymbol{\theta}_{\mathit{eff}}(t)$ converges to a point $\boldsymbol{\theta}_{\mathit{eff}}(\infty) \in \mathfrak{M}_I$, and

$$\|\boldsymbol{\theta}_{\mathit{eff}}(t) - \boldsymbol{\theta}_{\mathit{eff}}(\infty)\| \le \frac{K_r}{\mu_r} \sqrt{\mathcal{L}_{\mathit{AC}}(\boldsymbol{\theta}_{\mathit{eff}}(0))}\, e^{-\mu_r t}.$$

I.4 Proof of Lemma I.1

Proof. Fix $c \in I$ and $\Delta = (\mathbf{U}, \mathbf{W}) \in N_c\mathfrak{M}_I$ with $\|\Delta\| \le r$, where $\|\Delta\|^2 := \|\mathbf{U}\|_F^2 + \|\mathbf{W}\|_F^2$, and set $\boldsymbol{\theta}_{\text{eff}} := \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta$. Write $\|\cdot\|_{L^2} := \sqrt{\mathbb{E}_z\|\cdot\|^2}$. Since every point of $\mathfrak{M}_I$ is a global minimizer of $\mathcal{L}_{\text{AC}}$ with $\mathcal{L}_{\text{AC}}^\star = 0$,

$$\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}_{\text{AC}}^\star = \tfrac{1}{2}\mathbb{E}_z\big\|f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z)\big\|^2. \tag{57}$$

Step 1: Decomposition of the prediction error. Expanding $f_{\boldsymbol{\theta}_{\text{eff}}}(z) = (c^{-1}\bar{\mathbf{V}}_{21}^\star + \mathbf{W})\hat{\boldsymbol{\Sigma}}(z)(c\mathbf{P}_{12}^\star + \mathbf{U})\tilde{\mathbf{w}}$, cancelling the matched leading term, and applying the identities (51),

$$f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) = \underbrace{c^{-1}\mathbf{D}\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}} + c\,\mathbf{W}\mathbf{b}(z)}_{=:\mathbf{X}(z)} + \underbrace{\mathbf{W}\hat{\boldsymbol{\Sigma}}(z)\mathbf{U}\tilde{\mathbf{w}}}_{=:\mathbf{Y}(z)}, \tag{58}$$

which is the exact analogue of (22), with the scalar step size α there replaced by the step-size matrix $\mathbf{D}$.

Step 2: Lower bound on $\|\mathbf{X}\|_{L^2}$. Since $\mathbf{D} \succeq \alpha_\wedge \mathbf{I}_{m+d}$, we have $\|\mathbf{D}\mathbf{v}\| \ge \alpha_\wedge\|\mathbf{v}\|$ for every $\mathbf{v} \in \mathbb{R}^{m+d}$. Conditioning on $\tilde{\mathbf{w}}$ and applying Assumption 2(3) and then Assumption 2(2), exactly as in (23),

$$\mathbb{E}_z\|\mathbf{D}\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 \ge \alpha_\wedge^2\,\mathbb{E}_z\|\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2 \ge \alpha_\wedge^2\kappa_R\kappa_{\tilde{w}}\|\mathbf{U}\|_F^2.$$

Similarly, Assumption 2(4) yields $\mathbb{E}_z\|\mathbf{W}\mathbf{b}(z)\|^2 \ge \kappa_q\|\mathbf{W}\|_F^2$, as in (24). Setting $A := \mathbb{E}_z\|c^{-1}\mathbf{D}\mathbf{R}(z)\mathbf{U}\tilde{\mathbf{w}}\|^2$ and $B := \mathbb{E}_z\|c\,\mathbf{W}\mathbf{b}(z)\|^2$, Assumption 2(5) lower-bounds the cross term of $\mathbb{E}_z\|\mathbf{X}(z)\|^2$ by $-2\rho\sqrt{AB}$, and the same algebra as in (25), together with $c \in [c_-, c_+]$, gives

$$\mathbb{E}_z\|\mathbf{X}(z)\|^2 \ge (1-\rho)(A+B) \ge (1-\rho)\min\big\{c_+^{-2}\alpha_\wedge^2\kappa_R\kappa_{\tilde{w}},\, c_-^2\kappa_q\big\}\|\Delta\|^2 = m_0\|\Delta\|^2, \tag{59}$$

i.e., $\|\mathbf{X}\|_{L^2} \ge \sqrt{m_0}\,\|\Delta\|$.

Step 3: Upper bounds on $\|\mathbf{X}\|_{L^2}$ and $\|\mathbf{Y}\|_{L^2}$. By Assumption 2(1) and $\gamma \in [0,1)$,

$$\|\mathbf{x}_i\|^2 = \|\boldsymbol{\phi}_{V,i}\|^2 + \gamma^2\|\boldsymbol{\phi}_{V,i+1}\|^2 + r_{i+1}^2 + \gamma^{2i}\|\mathbf{g}_i\|^2 \le 2B_\phi^2 + B_r^2 + B_g^2 = B_\Sigma,$$

hence $\|\hat{\boldsymbol{\Sigma}}(z)\|_{\text{op}} \le B_\Sigma$ by the triangle inequality. Since $\boldsymbol{\psi}_i = \mathbf{S}\mathbf{x}_i$ with the selection matrix

$$\mathbf{S} := \begin{bmatrix} \mathbf{0}_{m\times d} & \mathbf{0}_{m\times d} & \mathbf{0}_{m\times 1} & \mathbf{I}_m \\ \mathbf{I}_d & \mathbf{0}_{d\times d} & \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times m} \end{bmatrix} \in \mathbb{R}^{(m+d)\times(2d+m+1)}, \qquad \mathbf{S}\mathbf{S}^\top = \mathbf{I}_{m+d},$$

we have $\mathbf{R}(z) = \mathbf{S}\hat{\boldsymbol{\Sigma}}(z)$ and thus $\|\mathbf{R}(z)\|_{\text{op}} \le B_\Sigma$. Moreover, $\|\mathbf{D}\|_{\text{op}} = \alpha_\vee$, and $\mathbf{P}_{12}^{\star\top}\mathbf{P}_{12}^\star = \text{diag}(1, \mathbf{0}_{m\times m}, 2\mathbf{I}_d)$ gives $\|\mathbf{P}_{12}^\star\|_{\text{op}} = \sqrt{2}$, whence $\|\mathbf{b}(z)\| \le \|\mathbf{P}_{12}^\star\|_{\text{op}}\|\hat{\boldsymbol{\Sigma}}(z)\|_{\text{op}}\|\tilde{\mathbf{w}}\| \le \sqrt{2}B_\Sigma B_{\tilde{w}}$. Therefore, following the derivations of (27) and (29),

$$\|\mathbf{X}\|_{L^2} \le c_-^{-1}\alpha_\vee B_\Sigma B_{\tilde{w}}\|\mathbf{U}\|_F + \sqrt{2}\,c_+ B_\Sigma B_{\tilde{w}}\|\mathbf{W}\|_F \le \sqrt{M_0}\,\|\Delta\|, \tag{60}$$

$$\|\mathbf{Y}\|_{L^2} \le B_\Sigma B_{\tilde{w}}\|\mathbf{U}\|_F\|\mathbf{W}\|_F \le C_Q\|\Delta\|^2. \tag{61}$$

Step 4: Conclusion. By bilinearity of the predictor, $f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)+t\Delta}(z) - f_{\bar{\boldsymbol{\theta}}_{\text{eff}}(c)}(z) = t\mathbf{X}(z) + t^2\mathbf{Y}(z)$ for all $t \in \mathbb{R}$. Steps 4–6 of the proof of Lemma C.2 use only this scaling structure, the identity (57), and the three bounds (59)–(61); repeating them verbatim yields the excess-loss upper bound

$$\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}_{\text{AC}}^\star \le \tfrac{1}{2}\big(\sqrt{M_0} + C_Q r\big)^2\|\Delta\|^2, \tag{62}$$

the directional-derivative lower bound $\langle\nabla\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}), \Delta\rangle \ge \big(m_0 - 3C_Q\sqrt{m_0}\,r\big)\|\Delta\|^2$, and finally the PL inequality (54) with the stated μ_r. □

I.5 Proof of Theorem I.1

Proof. The proof follows the four steps of the proof of Theorem 3.2 in Appendix G; we verify that each ingredient holds with the actor-critic quantities.

Step 1: Distance bound. For $\boldsymbol{\theta}_{\text{eff}} = \bar{\boldsymbol{\theta}}_{\text{eff}}(c) + \Delta \in \mathcal{T}_r(I)$, the reverse triangle inequality applied to (58), together with (59) and (61), gives

$$\sqrt{2\big(\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star_{\text{AC}}\big)} = \|\mathbf{X} + \mathbf{Y}\|_{L^2} \geq \|\mathbf{X}\|_{L^2} - \|\mathbf{Y}\|_{L^2} \geq \big(\sqrt{m_0} - C_Q r\big)\|\Delta\|,$$

so, with $\lambda_r := \frac{1}{2}\big(\sqrt{m_0} - C_Q r\big)^2$ and using that the normal displacement equals the distance to the manifold,

$$\operatorname{dist}(\boldsymbol{\theta}_{\text{eff}}, \mathfrak{M}_I) = \|\Delta\| \leq \frac{1}{\sqrt{\lambda_r}} \sqrt{\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star_{\text{AC}}}. \tag{63}$$

Step 2: Gradient upper bound. Let $\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z) := f_{\boldsymbol{\theta}_{\text{eff}}}(z) - f_{(\mathbf{P}^\star_{12}, \bar{\mathbf{V}}^\star_{21})}(z)$, so that $\mathbb{E}_z\|\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\|^2 = 2\big(\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star_{\text{AC}}\big)$. Differentiating the bilinear predictor,

$$\nabla_{\bar{\mathbf{V}}_{21}} \mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) = \mathbb{E}_z\big[\mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\big(\hat{\boldsymbol{\Sigma}}(z)\mathbf{P}_{12}\tilde{\mathbf{w}}\big)^\top\big], \qquad \nabla_{\mathbf{P}_{12}} \mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) = \mathbb{E}_z\big[\hat{\boldsymbol{\Sigma}}(z)^\top \bar{\mathbf{V}}_{21}^\top \mathbf{e}_{\boldsymbol{\theta}_{\text{eff}}}(z)\tilde{\mathbf{w}}^\top\big].$$

On $\mathcal{T}_r(I)$, using $\|\mathbf{P}^\star_{12}\|_F = \sqrt{2d+1}$ and $\|\bar{\mathbf{V}}^\star_{21}\|_F = \sqrt{\alpha^2 m + \beta^2 d}$, the triangle inequality gives

$$\|\mathbf{P}_{12}\|_F \leq c_+\sqrt{2d+1} + r =: P_{\max,r}, \qquad \|\bar{\mathbf{V}}_{21}\|_F \leq c_-^{-1}\sqrt{\alpha^2 m + \beta^2 d} + r =: V_{\max,r},$$

and Cauchy–Schwarz together with $\|\hat{\boldsymbol{\Sigma}}(z)\|_{\text{op}} \leq B_\Sigma$ and $\|\tilde{\mathbf{w}}\| \leq B_{\tilde{w}}$ yields, exactly as in the derivation of (40),

$$\|\nabla \mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}})\| \leq K_r \sqrt{\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}) - \mathcal{L}^\star_{\text{AC}}}, \qquad \forall\, \boldsymbol{\theta}_{\text{eff}} \in \mathcal{T}_r(I), \tag{64}$$

with $K_r = \sqrt{2} B_\Sigma B_{\tilde{w}} \sqrt{P^2_{\max,r} + V^2_{\max,r}}$ as defined in the theorem statement.

Steps 3–4: Tube invariance and convergence. Steps 3 and 4 of the proof of Theorem 3.2 rely only on the following ingredients: the local PL inequality (here Lemma I.1), the gradient upper bound (here (64)), the excess-loss upper bound (here (62)), the distance bound (here (63)), the definitions of the buffer distance and the initialization condition (here (55) and (56)), and local Lipschitzness of $\nabla \mathcal{L}_{\text{AC}}$ (immediate, since $\mathcal{L}_{\text{AC}}$ is a polynomial in $\boldsymbol{\theta}_{\text{eff}}$). Repeating those steps verbatim with $(\mathcal{L}, m_0, M_0, C_Q, K_r, \lambda_r, \mu_r)$ replaced by their actor-critic counterparts shows that the gradient flow solution exists for all $t \geq 0$ and remains in $\mathcal{T}_r(I)$, that the loss decays as $\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}(t)) \leq e^{-2\mu_r t}\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}(0))$, and that $\boldsymbol{\theta}_{\text{eff}}(t)$ converges to a point $\boldsymbol{\theta}_{\text{eff}}(\infty) \in \mathfrak{M}_I$ with $\|\boldsymbol{\theta}_{\text{eff}}(t) - \boldsymbol{\theta}_{\text{eff}}(\infty)\| \leq \frac{K_r}{\mu_r}\sqrt{\mathcal{L}_{\text{AC}}(\boldsymbol{\theta}_{\text{eff}}(0))}\, e^{-\mu_r t}$. This completes the proof. □

J Discussion and Numerical Verification of Assumption 1

In this section, we discuss the intuitions of Assumption 1 and provide numerical evidence that it holds in our experiments.

J.1 The diagnostic conditions

Our goal is to establish whether gradient training converges to an exact implementation, rather than merely whether such an implementation exists. This requires the training distribution to provide enough information in the parameter directions relevant to the target update, and Assumption 1 formalizes the requirement as excitation conditions on that distribution.

These conditions are not additional structure imposed on the problem: every training distribution has well-defined values of $\kappa_{\tilde{w}}, \kappa_R, \kappa_q \geq 0$ and $\rho \in [0, 1]$, so the substantive richness requirement for our convergence result is only the non-degeneracy condition

$$\kappa_{\tilde{w}}, \kappa_R, \kappa_q > 0, \qquad \rho < 1.$$

What the condition asks of the training tasks. Each quantity corresponds to a concrete requirement on the diversity of the training tasks:

- $\kappa_{\tilde{w}} > 0$ requires the behavior parameter to vary in all directions across tasks, which can be ensured by drawing a fresh behavior parameter for each task from a full-dimensional distribution. Initializing every prompt at the same point, e.g. $\mathbf{w}^{(0)} = \mathbf{0}$, reduces this diversity.
- $\kappa_R > 0$ requires sufficient variation in both the feature maps across tasks and the visited state–action pairs, the latter of which could be supported by using $\epsilon > 0$ exploration.
- $\kappa_q > 0$ suggests that the training trajectories should not be too long, because near the task's SARSA fixed point the mean update $\mathbb{E}[\boldsymbol{\phi}\delta]$ vanishes and late windows become largely uninformative. Thus, for a fixed data budget, κ_q favors sampling more tasks with fewer windows per task.
- $\rho < 1$ requires the two update channels not to be perfectly aligned, which is supported by jointly varying rewards, transitions, and features across tasks.

Together, these conditions ensure that the training distribution is sufficiently rich for the transformer to learn the underlying update rule rather than a fixed distribution-specific pattern.

J.2 Numerical check

The conditions are checkable in practice from the trajectory statistics $(\tilde{\mathbf{w}}, \mathbf{R}(z), \mathbf{b}(z))$ computed from the prompts generated during training. All the required quantities are determined by second moments of these statistics, so the numerical check reduces to eigenvalue and constrained canonical-correlation computations.

The expectations in Assumption 1 are over the prompts encountered during training. This distribution mixes over the random task *and* the position of the context window along the teacher trajectory, so the behavior parameter is not fixed but evolves between consecutive windows through the teacher update. We sample directly from this distribution using Monte Carlo for the random-MDP family in our experiments. For each of $N = 10{,}000$ samples, we draw a fresh MDP, feature map, and initial parameter $\mathbf{w}^{(0)}$ exactly as in Section 4.1; draw a window index $t \sim \mathrm{Unif}\{0, \dots, T/n - 2\}$ (the final window is excluded because the training loop does not train on it); roll the teacher forward for t windows, applying the teacher update (2), $\mathbf{w}^{(j+1)} = \mathbf{w}^{(j)} + \frac{\alpha}{n} \sum_{i=0}^{n-1} \boldsymbol{\phi}_i \delta_i$, after each; and record $\tilde{\mathbf{w}}_j$, $\mathbf{R}_j := \mathbf{R}(z_j)$, and $\mathbf{b}_j := \mathbf{b}(z_j)$ from the selected window.

J.2.1 The excitation constants

The estimates of $\kappa_{\tilde{w}}$ and κ_q are the smallest eigenvalues of the empirical second moments,

$$\hat{\kappa}_{\tilde{w}} := \lambda_{\min}\left(\frac{1}{N}\sum_{j=1}^{N} \tilde{\mathbf{w}}_j \tilde{\mathbf{w}}_j^\top\right), \qquad \hat{\kappa}_q := \lambda_{\min}\left(\frac{1}{N}\sum_{j=1}^{N} \mathbf{b}_j \mathbf{b}_j^\top\right).$$

Condition (3) is different: it quantifies over every behavior parameter $\mathbf{w}$, which no finite computation can exhaust. We approximate the quantifier by the sweep $\mathcal{W}_{\text{sweep}} := \{\mathbf{0}, \pm\mathbf{u}_1, \dots, \pm\mathbf{u}_8, 5\mathbf{u}_1\}$, where $\mathbf{u}_1, \dots, \mathbf{u}_8$ are independent random unit directions. For each fixed $\mathbf{w} \in \mathcal{W}_{\text{sweep}}$ we draw $M = 800$ fresh MDPs (14,400 in total) and set

$$\hat{\kappa}_R := \min_{\mathbf{w} \in \mathcal{W}_{\text{sweep}}} \lambda_{\min}\left(\frac{1}{M}\sum_{j=1}^{M} \mathbf{R}_j^\top \mathbf{R}_j\right).$$

For the SARSA task distribution in Section 4.1, the smallest empirical eigenvalue over $\mathcal{W}_{\text{sweep}}$ occurs at $\mathbf{w} = \mathbf{0}$.

J.2.2 The no-cancellation constant

Condition (5) bounds the correlation between the two channels $\mathbf{a}_j(\mathbf{U}) := \mathbf{R}_j \mathbf{U} \tilde{\mathbf{w}}_j \in \mathbb{R}^d$ and $\mathbf{q}_j(\mathbf{W}) := \mathbf{W}\mathbf{b}_j \in \mathbb{R}^d$ through which a perturbation $(\mathbf{U}, \mathbf{W}) \in \boldsymbol{\Theta}_{\text{eff}}$ of a manifold point $(c\mathbf{P}_{12}^\star, c^{-1}\bar{\mathbf{V}}_{21}^\star)$ enters the transformer output. Their empirical energies and cross term are

$$A(\mathbf{U}) := \frac{1}{N}\sum_j \|\mathbf{a}_j\|^2, \qquad B(\mathbf{W}) := \frac{1}{N}\sum_j \|\mathbf{q}_j\|^2, \qquad C(\mathbf{U}, \mathbf{W}) := \frac{1}{N}\sum_j \langle \mathbf{a}_j, \mathbf{q}_j \rangle.$$

The empirical analogue of ρ is the largest normalized negative cross term,

$$\hat{\rho}_{\mathrm{emp}} := \sup_{c>0} \sup_{\substack{(\mathbf{U},\mathbf{W})\in\mathcal{N}_c \\ A(\mathbf{U})B(\mathbf{W})>0}} \frac{-C(\mathbf{U},\mathbf{W})}{\sqrt{A(\mathbf{U})B(\mathbf{W})}}. \tag{65}$$

We reduce the joint supremum in (65) to a one-dimensional search over c by expressing it in terms of a constrained eigenvalue for each c.

Since $\mathbf{a}_j$ and $\mathbf{q}_j$ are linear in the perturbations, A, B, and C are quadratic forms in the vectorizations $\mathbf{u} := \mathrm{rvec}(\mathbf{U})$ and $\mathbf{v} := \mathrm{rvec}(\mathbf{W})$, where rvec stacks the rows of a matrix into a column vector. Indeed, the identity $\mathrm{rvec}(\mathbf{AXB}) = (\mathbf{A}\otimes\mathbf{B}^\top)\,\mathrm{rvec}(\mathbf{X})$ gives $\mathbf{a}_j = (\mathbf{R}_j\otimes\tilde{\mathbf{w}}_j^\top)\mathbf{u}$ and $\mathbf{q}_j = (\mathbf{I}_d\otimes\mathbf{b}_j^\top)\mathbf{v}$, hence

$$A(\mathbf{U}) = \mathbf{u}^\top\mathbf{M}_A\mathbf{u}, \qquad B(\mathbf{W}) = \mathbf{v}^\top\mathbf{M}_B\mathbf{v}, \qquad C(\mathbf{U},\mathbf{W}) = \mathbf{u}^\top\mathbf{M}_C\mathbf{v},$$

with moment matrices accumulated in a single pass over the samples:

$$\mathbf{M}_A := \frac{1}{N}\sum_{j=1}^{N}(\mathbf{R}_j^\top\mathbf{R}_j)\otimes(\tilde{\mathbf{w}}_j\tilde{\mathbf{w}}_j^\top), \qquad \mathbf{M}_B := \mathbf{I}_d\otimes\frac{1}{N}\sum_{j=1}^{N}\mathbf{b}_j\mathbf{b}_j^\top,$$

$$\mathbf{M}_C := \frac{1}{N}\sum_{j=1}^{N}\mathbf{R}_j^\top\otimes(\tilde{\mathbf{w}}_j\mathbf{b}_j^\top).$$

Let $\mathbf{M}_A = \mathbf{L}_A\mathbf{L}_A^\top$ and $\mathbf{M}_B = \mathbf{L}_B\mathbf{L}_B^\top$ be Cholesky factorizations, and pass to the whitened coordinates

$$\boldsymbol{\xi} := \mathbf{L}_A^\top\mathbf{u}, \qquad \boldsymbol{\eta} := \mathbf{L}_B^\top\mathbf{v}, \qquad \mathbf{K} := \mathbf{L}_A^{-1}\mathbf{M}_C\mathbf{L}_B^{-\top}, \tag{66}$$

in which $A = \|\boldsymbol{\xi}\|^2$, $B = \|\boldsymbol{\eta}\|^2$, and $C = \boldsymbol{\xi}^\top\mathbf{K}\boldsymbol{\eta}$. Without the normal-space constraint, the inner supremum in (65) equals the largest singular value $\sigma_1(\mathbf{K})$. Along the scaling tangent $(\mathbf{U},\mathbf{W})\propto(\mathbf{P}_{12}^\star, -c^{-2}\bar{\mathbf{V}}_{21}^\star)$, the two channels satisfy $\mathbf{q}_j = -\alpha c^{-2}\mathbf{a}_j$ for every sample j. Their normalized negative correlation therefore attains its upper bound of one, giving $\sigma_1(\mathbf{K}) = 1$. The normal-space constraint excludes this tangent direction, allowing the diagnostic to test whether comparable cancellation occurs along any remaining perturbation direction.

In the whitened coordinates (66), the constraint is a single hyperplane. Let $\mathbf{n}_U := \mathbf{L}_A^{-1}\,\mathrm{rvec}(\mathbf{P}_{12}^\star)$ and $\mathbf{n}_W := \mathbf{L}_B^{-1}\,\mathrm{rvec}(\bar{\mathbf{V}}_{21}^\star)$, and define

$$\mathbf{n}(c) := \begin{bmatrix} c\,\mathbf{n}_U \\ -c^{-1}\mathbf{n}_W \end{bmatrix}, \qquad \mathbf{H} := \begin{bmatrix} \mathbf{0} & \mathbf{K} \\ \mathbf{K}^\top & \mathbf{0} \end{bmatrix}. \tag{67}$$

For $AB > 0$, normalize the two blocks by setting

$$\mathbf{z} := \begin{bmatrix} \dfrac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} \\ \dfrac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|} \end{bmatrix}. \tag{68}$$

Absorbing the normalization factors into the scale $c > 0$, the constraint in (10) becomes

$$\mathbf{n}(c)^\top\mathbf{z} = 0. \tag{69}$$

The unit block norms in (68) and the definition of $\mathbf{H}$ in (67) give

$$\frac{\mathbf{z}^\top\mathbf{H}\mathbf{z}}{\|\mathbf{z}\|^2} = \frac{\boldsymbol{\xi}^\top\mathbf{K}\boldsymbol{\eta}}{\|\boldsymbol{\xi}\|\,\|\boldsymbol{\eta}\|} = \frac{C}{\sqrt{AB}}. \tag{70}$$

For a general nonzero vector $\mathbf{x}$ satisfying the same hyperplane constraint (69), partition it into two blocks and set

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \qquad A := \|\mathbf{x}_1\|^2, \qquad B := \|\mathbf{x}_2\|^2, \qquad C := \mathbf{x}_1^\top\mathbf{K}\mathbf{x}_2.$$

Thus $\|\mathbf{x}\|^2 = A + B$ and $\mathbf{x}^\top \mathbf{H}\mathbf{x} = 2C$. When $C < 0$, the inequality $A + B \geq 2\sqrt{AB}$ gives

$$-\frac{\mathbf{x}^\top \mathbf{H}\mathbf{x}}{\|\mathbf{x}\|^2} = \frac{-2C}{A+B} \leq \frac{-C}{\sqrt{AB}} \leq \hat{\rho}_{\mathrm{emp}}. \tag{71}$$

Vectors with $C \geq 0$ give nonpositive values. Conversely, (70) shows that the normalized construction (68) realizes every objective value in (65) as a negative Rayleigh quotient at some $c > 0$. Together with (71), this proves that the joint suprema agree, giving

$$\hat{\rho}_{\mathrm{emp}} = \sup_{c>0} \big[-\mu(c)\big], \qquad \mu(c) := \min_{\substack{\mathbf{x}\neq\mathbf{0} \\ \mathbf{n}(c)^\top \mathbf{x}=0}} \frac{\mathbf{x}^\top \mathbf{H}\mathbf{x}}{\|\mathbf{x}\|^2}. \tag{72}$$

For fixed c, the constrained minimum $\mu(c)$ in (72) is available exactly. The singular value decomposition $\mathbf{K} = \sum_i \sigma_i \boldsymbol{\ell}_i \mathbf{r}_i^\top$ gives the nonzero eigenpairs of $\mathbf{H}$ in (67) as

$$\left(\pm\sigma_i,\ \frac{1}{\sqrt{2}} \begin{bmatrix} \boldsymbol{\ell}_i \\ \pm \mathbf{r}_i \end{bmatrix}\right), \qquad \sigma_i > 0.$$

Let $(\lambda_j, \mathbf{e}_j)$ enumerate an orthonormal eigenbasis of $\mathbf{H}$, including the zero eigenspace, with $\lambda_1 \leq \lambda_2 \leq \cdots$. The squared projection of the unit normal onto $\mathbf{e}_j$ is

$$\omega_j(c) := \left(\mathbf{e}_j^\top \frac{\mathbf{n}(c)}{\|\mathbf{n}(c)\|}\right)^2 = \frac{\big(\mathbf{e}_j^\top \mathbf{n}(c)\big)^2}{\|\mathbf{n}(c)\|^2}. \tag{73}$$

For fixed c, the Lagrangian for the constrained minimization in (72) is

$$\mathcal{L}_c(\mathbf{x}, \nu) := \frac{\mathbf{x}^\top \mathbf{H}\mathbf{x}}{\|\mathbf{x}\|^2} - \nu\, \mathbf{n}(c)^\top \mathbf{x}.$$

At a minimizer, the first-order condition $\nabla_{\mathbf{x}} \mathcal{L}_c = \mathbf{0}$ gives

$$\nabla_{\mathbf{x}} \left(\frac{\mathbf{x}^\top \mathbf{H}\mathbf{x}}{\|\mathbf{x}\|^2}\right) = \frac{2}{\|\mathbf{x}\|^2}(\mathbf{H} - \mu(c)\mathbf{I})\mathbf{x} = \nu\, \mathbf{n}(c). \tag{74}$$

Thus the gradient is parallel to $\mathbf{n}(c)$. Writing $\beta := \frac{\nu\|\mathbf{x}\|^2}{2}$, (74) becomes $(\mathbf{H} - \mu(c)\mathbf{I})\mathbf{x} = \beta \mathbf{n}(c)$. When $\mu(c) \neq \lambda_j$ for every j, we have $\beta \neq 0$ because $\mathbf{x} \neq \mathbf{0}$. Projecting onto $\mathbf{e}_j$ gives

$$\mathbf{e}_j^\top \mathbf{x} = \beta \frac{\mathbf{e}_j^\top \mathbf{n}(c)}{\lambda_j - \mu(c)}.$$

Substituting this expression into the hyperplane constraint in (72) yields

$$0 = \mathbf{n}(c)^\top \mathbf{x} = \sum_j \big(\mathbf{e}_j^\top \mathbf{n}(c)\big)\big(\mathbf{e}_j^\top \mathbf{x}\big) = \beta \sum_j \frac{\big(\mathbf{e}_j^\top \mathbf{n}(c)\big)^2}{\lambda_j - \mu(c)}.$$

Dividing by $\beta\|\mathbf{n}(c)\|^2$ and using the weights in (73) gives the secular equation

$$\sum_j \frac{\omega_j(c)}{\lambda_j - \mu(c)} = 0. \tag{75}$$

Eigenvalue interlacing places $\mu(c)$ in $[\lambda_1, \lambda_2]$, providing the interval for bracketed root finding in (75). If $\mu(c)$ coincides with an eigenvalue, the undivided stationarity condition (74) applies. In particular, $\omega_1(c) = 0$ in (73) implies that $\mathbf{e}_1$ satisfies the hyperplane constraint and $\mu(c) = \lambda_1$.

It remains to evaluate the one-dimensional supremum over c in (72). Only the direction of $\mathbf{n}(c)$ matters. We evaluate $-\mu(c)$ on a 97-point grid uniform in $\log c \in [-12, 12]$ and refine the five best grid neighborhoods by bounded one-dimensional optimization. We also evaluate the two limiting cases, obtained from (67) as

$$\lim_{c\to\infty} \frac{\mathbf{n}(c)}{\|\mathbf{n}(c)\|} = \frac{1}{\|\mathbf{n}_U\|} \begin{bmatrix} \mathbf{n}_U \\ \mathbf{0} \end{bmatrix}, \qquad \lim_{c\to 0^+} \frac{\mathbf{n}(c)}{\|\mathbf{n}(c)\|} = \frac{1}{\|\mathbf{n}_W\|} \begin{bmatrix} \mathbf{0} \\ -\mathbf{n}_W \end{bmatrix}.$$

Let $\mathcal{C}_{\mathrm{num}}$ collect all points evaluated by this search, including the two limiting cases. We define

$$\hat{\rho}_{\mathrm{num}} := \max_{c \in \mathcal{C}_{\mathrm{num}}} \big[-\mu(c)\big] \approx \sup_{c>0} \big[-\mu(c)\big] = \hat{\rho}_{\mathrm{emp}}.$$

We report $\hat{\rho}_{\mathrm{num}}$ as our numerical estimate of $\hat{\rho}_{\mathrm{emp}}$.

J.2.3 Results

With $N = 10{,}000$ mixture samples and the $14{,}400$-MDP sweep for κ_R, we obtain

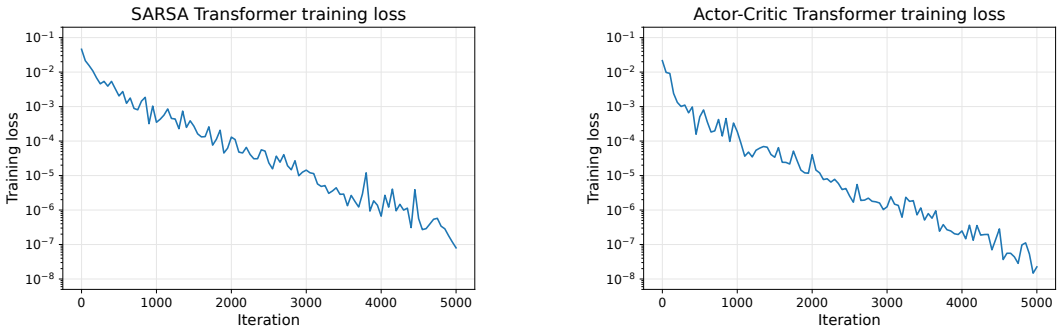
$$\hat{\kappa}_{\tilde{w}} = 0.154, \qquad \hat{\kappa}_R = 0.043, \qquad \hat{\kappa}_q = 3.9 \times 10^{-3}, \qquad \hat{\rho}_{\text{num}} = 0.885.$$

All three excitation estimates are positive, and $\hat{\rho}_{\text{num}} = 0.885 < 1$.

K Additional Experiments and Details

K.1 Training loss curves

We plot the training loss $\hat{\mathcal{L}}(\boldsymbol{\theta}^{(t)})$ over training iterations for both the SARSA transformer and the Actor-Critic transformer in our experiments presented in Section 4. The results are shown in Figure 4.



(a) SARSA transformer.

(b) Actor-Critic transformer.

Figure 4: Training loss $\hat{\mathcal{L}}(\boldsymbol{\theta}^{(t)})$ over training iterations for the SARSA transformer (left) and the Actor-Critic transformer (right). The vertical axes are logarithmic. In both settings, the loss decays exponentially, consistent with the local convergence guarantee in Theorem 3.2.

K.2 Cosine similarity to the theoretical constructions and effect of the training budget

We quantify the visual similarity reported in Figure 2 between the learned parameters and the theoretical constructions in (5) and (15). Because the constructions are recovered only up to the scaling symmetry $(\mathbf{P}, \mathbf{V}) \mapsto (c\mathbf{P}, c^{-1}\mathbf{V})$, cosine similarity is the natural metric: for a learned block $\mathbf{A}$ with analytical counterpart $\mathbf{A}^\star$, we report

$$\cos(\mathbf{A}) := \frac{\langle \mathbf{A}, \mathbf{A}^\star \rangle_F}{\|\mathbf{A}\|_F \, \|\mathbf{A}^\star\|_F},$$

which is invariant to the scale c, and satisfies $|\cos(\mathbf{A})| = 1$ if and only if the learned block is an exact member of the theoretical family. We evaluate it on the effective sub-blocks $(\mathbf{P}_{12}, \bar{\mathbf{V}}_{21})$.

Table 1 reports the cosine similarities of checkpoints trained with $K = 100$, 500, $1{,}000$, and $5{,}000$ tasks, for both the SARSA and the actor-critic transformers of Section 4. The similarity increases monotonically with the training budget and, at $K = 5{,}000$, the constructions of (5) and (15) are recovered exactly up to the scalar symmetry.

Table 1: Cosine similarity between the learned effective blocks and the theoretical constructions in (5) (SARSA) and (15) (actor-critic), as a function of the number of training tasks K.

	SARSA		Actor-critic	
Training tasks K	$\cos(\mathbf{P}_{12})$	$\cos(\bar{\mathbf{V}}_{21})$	$\cos(\mathbf{P}_{12})$	$\cos(\bar{\mathbf{V}}_{21})$
100	0.411	0.856	0.571	0.934
500	0.980	0.964	0.973	0.988
1,000	0.997	0.993	0.993	0.995
5,000	1.000	1.000	1.000	1.000

Parameter-level convergence translates into control performance at deployment. Table 2 reports the final mean discounted return of the same checkpoints and of the analytical teacher on 100 held-out MDPs, following the closed-loop evaluation protocol of Section 4.2 (SARSA: after 100 in-context updates; actor-critic: after 3,000 in-context updates). The transformer–teacher gap closes monotonically as the training budget grows.

Table 2: Final mean discounted return on 100 held-out MDPs as a function of the number of training tasks K.

	SARSA		Actor-critic	
Training tasks K	transformer	teacher	transformer	teacher
100	0.30	0.42	0.08	0.39
500	0.37	0.42	0.18	0.39
1,000	0.40	0.42	0.32	0.39
5,000	0.41	0.42	0.39	0.39

K.3 Non-tabular MDPs: continuous-state MountainCar

Our theoretical setting operates on linear function approximation with an arbitrary feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, and all results (Theorems 3.1, 3.2, and 3.3) are stated at this level of generality. Moreover, the convergence guarantee depends on the training distribution only through the richness conditions of Assumptions 1 and 2, which do not distinguish tabular from non-tabular environments. To verify empirically that the recovered block structure generalizes beyond the tabular case, we repeat the SARSA experiment on a parameterized family of MountainCar tasks with continuous states.

Setup. The two-dimensional continuous state $s = (x, \dot{x})$ (position and velocity) is represented by a 12-dimensional Gaussian radial basis feature vector $\boldsymbol{\psi}_a(s) \in \mathbb{R}^{12}$ for each action, with the RBF centers and bandwidth independently sampled for each task and action. The corresponding state–action feature is

$$\boldsymbol{\phi}(s, a) = \mathbf{e}_a \otimes \boldsymbol{\psi}_a(s) \in \mathbb{R}^{36},$$

where $\mathbf{e}_a \in \mathbb{R}^3$ is the one-hot encoding of action $a \in \{0, 1, 2\}$ (accelerate left, coast, and accelerate right, respectively). The feature dimension $d = 36$ and transformer input dimension $D = 3d + 2 = 110$ are identical to those in our tabular SARSA experiments. The transition follows the MountainCar dynamics

$$\dot{x}_{t+1} = \mathrm{clip}(\dot{x}_t + (a_t - 1)F - G\cos(3x_t), -0.07, 0.07)\,, \qquad x_{t+1} = \mathrm{clip}(x_t + \dot{x}_{t+1}, -1.2, 0.6)\,,$$

with the velocity reset to zero at the left boundary. For each task, the engine-force coefficient and gravity coefficient are independently sampled as $F \sim \mathrm{Unif}(0.0015, 0.003)$ and $G \sim \mathrm{Unif}(0.0015, 0.003)$ and are then held fixed within the task. Each task also independently resamples the goal position, initial-state distribution, step cost, goal reward, and RBF basis. Upon reaching the goal, the car receives the goal reward and is reset to a newly sampled initial state, producing a continuing trajectory. We train the same one-layer looped linear-attention transformer using Algorithm 1 on the MountainCar task family.

Task distribution. For each task, we draw the goal position $x_{\mathrm{goal}} \sim \mathrm{Unif}(0.4, 0.55)$, the reward for a non-goal transition $r_{\mathrm{step}} \sim \mathrm{Unif}(-0.1, -0.02)$, and the goal reward $r_{\mathrm{goal}} \sim \mathrm{Unif}(1, 2)$. To define the initial-state distribution, we draw a center $c_x \sim \mathrm{Unif}(-0.9, x_{\mathrm{goal}} - 0.1)$ and a half-width $h_x \sim \mathrm{Unif}(0.1, 0.5)$, then sample

$$\begin{aligned} x_0 &\sim \mathrm{Unif}(\max\{c_x - h_x, -1.2\}, \min\{c_x + h_x, x_{\mathrm{goal}} - 0.02\})\,, \\ \dot{x}_0 &\sim \mathrm{Unif}(-0.02, 0.02). \end{aligned}$$

These task parameters are held fixed throughout the trajectory. A transition with $x_{t+1} \geq x_{\mathrm{goal}}$ receives r_{goal} instead of r_{step}, followed by a reset drawn from the same initial-state distribution; reaching the goal does not end the training task.

Feature construction. We normalize the state to $\mathbf{z}(s) = ((x+1.2)/1.8, (\dot{x}+0.07)/0.14) \in [0,1]^2$. For every task and action a, we independently draw $\sigma_a \sim \mathrm{Unif}(0.15, 0.35)$ and perturb each center of a 4×3 grid by independent coordinate-wise noise $\boldsymbol{\xi}_{a,jk} \sim \mathrm{Unif}(-0.08, 0.08)^2$:

$$\mathbf{c}_{a,jk} = \left(\frac{j+1/2}{4}, \frac{k+1/2}{3}\right) + \boldsymbol{\xi}_{a,jk}, \qquad j = 0, \ldots, 3, \quad k = 0, \ldots, 2,$$

$$[\mathbf{q}_a(s)]_{jk} = \exp\left(-\frac{\|\mathbf{z}(s) - \mathbf{c}_{a,jk}\|_2^2}{2\sigma_a^2}\right), \qquad \boldsymbol{\psi}_a(s) = \frac{\mathbf{q}_a(s)}{\|\mathbf{q}_a(s)\|_2}.$$

The 12 entries are stacked in a fixed order. The basis is fixed within a task and resampled independently for each action and each new task.

Training details. We use discount $\gamma = 0.99$, SARSA step size $\alpha = 0.2$, exploration $\epsilon = 0.1$, window length $n = 20$, and a horizon of 1,000 environment transitions per task. At each task's start, the entries of $\mathbf{w}^{(0)} \in \mathbb{R}^{36}$ are sampled independently from $\mathrm{Unif}(-1, 1)$. The teacher parameter is updated after each window and determines the behavior policy for the next window. The final window, which reaches the horizon, is not used for training, giving 49 optimizer updates per task. The transformer has one layer and one linear-attention head, with the block initialization described in Section 4.1. We use Adam with initial learning rate 10^{-3} and exponential decay by a factor of 0.99 every 10 MDPs. Training runs for $K = 10{,}000$ tasks.

Results. After training, all trainable blocks outside the effective parameter $(\mathbf{P}_{12}, \bar{\mathbf{V}}_{21})$ are exactly zero, consistent with Proposition 3.1, so we report in Table 3 the cosine similarity between each learned block and its analytical SARSA counterpart in (5), at training checkpoints of $K = 100$, 500, 1,000, 5,000, and 10,000 tasks. Figure 5 shows the full learned matrices $\mathbf{P}$ and $\mathbf{V}$, and Figure 6 shows the training loss. The learned parameter matrices recover the predicted block structure in this non-tabular, continuous-state setting as well.

Table 3: Cosine similarity between the learned effective blocks and the analytical SARSA construction (5) on the MountainCar task family, as a function of the number of training tasks K.

Training tasks K	$\cos(\mathbf{P}_{12})$	$\cos(\bar{\mathbf{V}}_{21})$
100	0.064	0.050
500	0.334	0.301
1,000	0.549	0.508
5,000	0.897	0.901
10,000	1.000	0.997

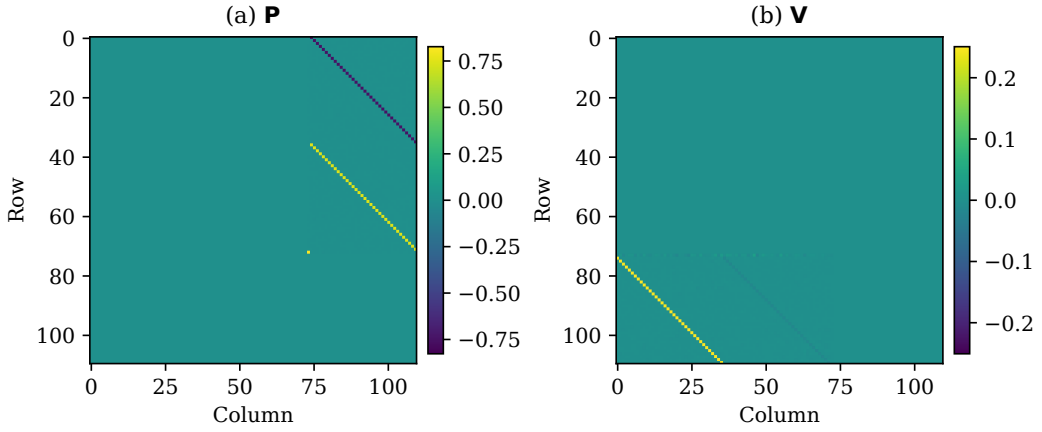


Figure 5: Full parameter matrices $\mathbf{P}$ (left) and $\mathbf{V}$ (right) learned after $K = 10{,}000$ MountainCar training tasks.

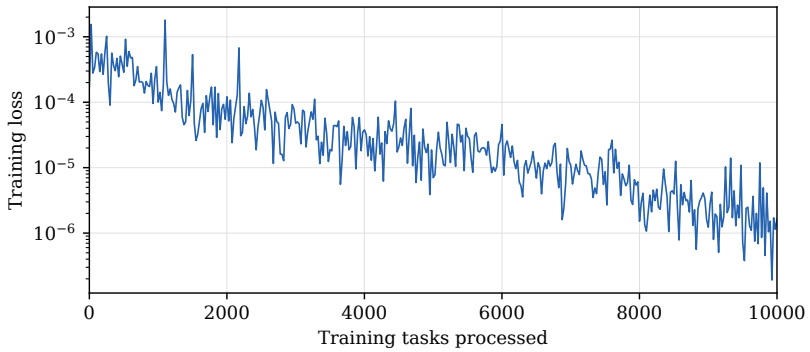


Figure 6: Training loss on the MountainCar task family over $K = 10{,}000$ tasks. The vertical axis is logarithmic.

K.4 Multi-layer, multi-head, and nonlinear attention variants

Our construction extends directly to multi-head and multi-layer linear attention: the outputs of a multi-head layer are summed, so zeroing the value maps of the extra heads recovers the single-head implementation exactly, and each attention block acts residually via (1), so setting $\mathbf{V} = \mathbf{0}$ in the remaining layers of an L-layer stack reduces them to the identity. Nonlinear activations change the picture: under ReLU attention, *two* heads recover linear attention exactly, since $\mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x$, whereas a softmax head cannot represent the update exactly and can only approximate a linear head using additional heads.

To test whether these representational considerations are predictive of trainability, we repeat the main SARSA experiment from Section 4.1 with nonlinear-attention variants: single- and two-head ReLU and softmax attention, obtained by applying the corresponding activation to the attention scores $\mathbf{H}^\top \mathbf{P} \mathbf{H}$ in (1) (with the outputs of multiple heads summed), keeping all other settings fixed. Table 4 reports the mean discounted return on 100 held-out MDPs after 100 closed-loop in-context updates, following the evaluation protocol of Section 4.2, for models trained with $K = 100$, 500, 1,000, and 5,000 tasks, alongside the analytical SARSA teacher for reference.

Table 4: Mean discounted return on 100 held-out MDPs after 100 closed-loop in-context updates, for linear-, ReLU-, and softmax-attention transformers trained with K tasks.

	linear	ReLU		softmax	
Training tasks K	1 head	1 head	2 heads	1 head	2 heads
100	0.290	0.112	0.055	0.076	0.104
500	0.387	0.191	0.200	0.125	0.114
1,000	0.370	0.210	0.273	0.121	0.139
5,000	0.404	0.219	0.402	0.110	0.136
analytical SARSA teacher			0.403		

The linear-attention transformer closes the gap to the teacher as the training budget grows and matches it at $K = 5{,}000$. ReLU attention shows exactly the head-count dependence predicted by the reduction $\mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x$: a single ReLU head, which cannot realize this identity, saturates well below the teacher, while with two heads gradient training finds a ReLU implementation of the update and reaches the teacher after a slower start. Softmax attention, in contrast, plateaus early with either one or two heads. This is consistent with the representational discussion above: a softmax head cannot represent the SARSA update exactly, and approximating it requires more attention heads. These results also show that supplying the teacher update as the training target does not by itself guarantee that training will recover it: whether imitation succeeds depends on the architecture's ability to represent the update.

K.5 Out-of-distribution deployment: cliff walk

If the transformer has learned the update rule itself, rather than a pattern tied to the training distribution, then the same frozen weights should work on structured MDPs it has never seen, provided the feature dimension matches. To test this, we deploy the *same* trained SARSA transformer from Section 4 (trained on random 9-state MDPs, and never on any grid world) on a single fixed 3×3 cliff-walk task.

Setup. The four actions move up, down, left, or right; a move that would leave the grid leaves the state unchanged, and with slip probability 0.1 the executed action is replaced by a uniformly random one. The start is $(1, 1)$, the goal is $(3, 3)$, and the cliff is the center cell $(2, 2)$, so the agent must detour around it. Entering the cliff incurs a penalty of -1 and resets the agent to the start; reaching the goal pays $+1$ and also resets to the start; every other transition costs -0.125. As during training, the transformer never sees the grid coordinates: states and actions enter the prompt only through a random feature map $\boldsymbol{\phi}(s, a) \in \mathbb{R}^{36}$ with i.i.d. $\mathrm{Unif}(-1, 1)$ entries. Starting from $\mathbf{w}^{(0)} = \mathbf{0}$, each in-context update follows the closed-loop protocol of Section 4.2: the current parameter $\mathbf{w}^{(t)}$ defines an ϵ-greedy behavior policy ($\epsilon = 0.1$) that generates the next $n = 20$ environment steps, those steps form the prompt, and the transformer's output becomes $\mathbf{w}^{(t+1)}$. Test prompts are undiscounted ($\gamma = 1$), and the analytical SARSA teacher (step size $\alpha = 0.2$) is run in its own closed loop on the same random stream. After 100 in-context updates we freeze $\mathbf{w}^{(100)}$ and run its greedy ($\epsilon = 0$) policy from the start for 32 episodes, recording the number of steps taken to first reach the goal. For reference, "optimal" is the expected hitting time of the best policy under the same slip dynamics (computed by value iteration) and "random" is the uniform policy.

Table 5: Average number of steps to reach the goal on the 3×3 cliff-walk task, for the final greedy policy after 100 in-context updates.

Model	transformer	analytical SARSA	optimal	random
Steps to goal	4.56	4.56	4.50	52.84

Results. As reported in Table 5, the frozen transformer reaches the goal in every episode and matches the algorithm it implements to two decimals, with the residual 0.06 steps above the optimal 4.50 due to slip. Its final greedy policy is a shortest path around the cliff, $(1, 1) \to (2, 1) \to (3, 1) \to (3, 2) \to (3, 3)$. The pretrained model thus behaves like a learner rather than a memorized map: the same frozen weights, trained only on random MDPs, carry out in-context policy improvement on a structured task far outside the training distribution.

K.6 Prompt ablation for actor-critic

The actor-critic prompt (14) supplies the score-function terms $\gamma^i \mathbf{g}_{\boldsymbol{\lambda}}(s_i, a_i)$ pre-computed. To test a less engineered prompt, we repeat the main actor-critic experiment with an identical setup, except that the score-function row is replaced by the raw policy feature $\boldsymbol{\phi}_\pi(s_i, a_i)$ of the executed action; the training target remains the exact actor-critic update (3). Table 6 reports the mean discounted return on 100 held-out MDPs after $3{,}000$ closed-loop in-context updates, for models trained with K tasks.

Table 6: Mean discounted return on 100 held-out MDPs after $3{,}000$ closed-loop in-context updates, for the actor-critic transformer trained with the original prompt (14) and with the raw prompt in which the score-function row is replaced by $\boldsymbol{\phi}_\pi(s_i, a_i)$.

Training tasks K	original prompt	raw prompt ($\boldsymbol{\phi}_\pi$ row)	teacher
100	0.07	0.04	0.39
500	0.22	0.12	0.39
1,000	0.32	0.17	0.39
5,000	0.39	0.31	0.39
10,000	0.39	0.31	0.39
15,000	0.39	0.32	0.39

Training remains stable and the return improves monotonically with K, but while the original prompt recovers the teacher exactly by $K = 5{,}000$, the raw-prompt model saturates at ≈ 0.31 and the gap persists through $K = 15{,}000$. This is expected from the definition $\mathbf{g}_{\boldsymbol{\lambda}}(s, a) = \boldsymbol{\phi}_\pi(s, a) - \sum_b \boldsymbol{\phi}_\pi(s, b)\, \pi_{\boldsymbol{\lambda}}(b \mid s)$: the baseline term depends on the policy features of *all* actions available at each step, while the raw prompt contains only the executed action's feature, so without additional embedding dimensions or layers the model cannot infer the score function needed for the policy gradient. Consistently, as shown in Figure 7, the trained weights show that the model still implements the critic update exactly, while the actor readout is learned as a strongly shrunk surrogate, which keeps closed-loop deployment stable but limits policy improvement.

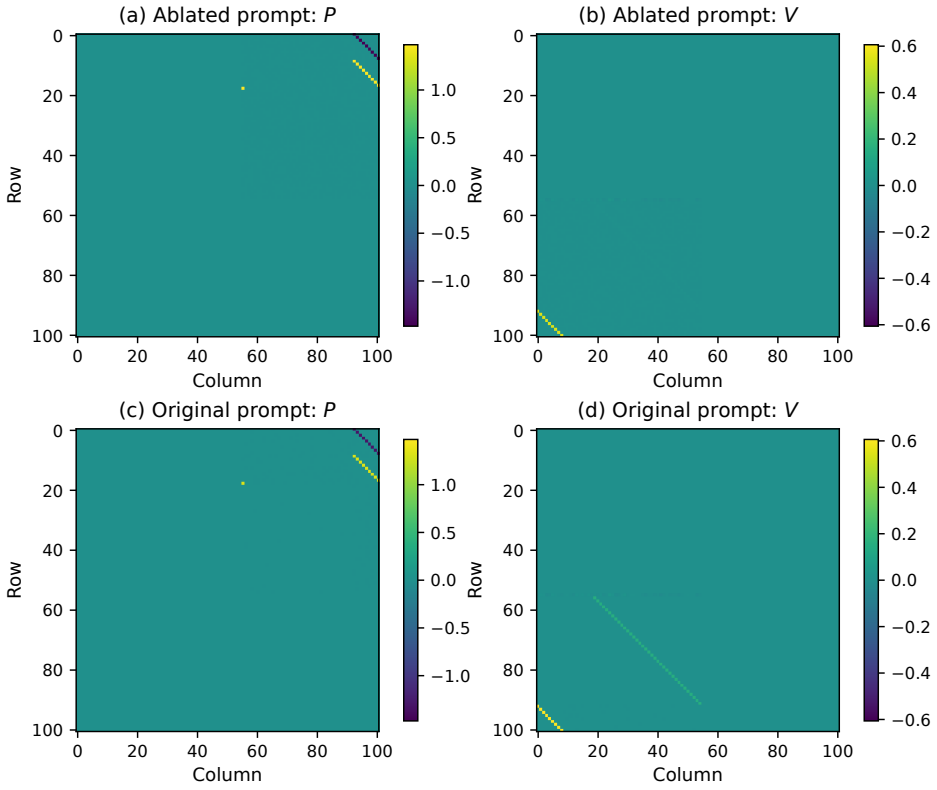


Figure 7: Final learned $\mathbf{P}$ (left) and $\mathbf{V}$ (right) after $K = 15{,}000$ training tasks, using the ablated prompt with raw policy features (top) and the original actor-critic prompt (bottom).

K.7 Runtime

The trainings and evaluations of the transformers used in our experiment were conducted on a MacBook Pro with the following specifications:

- Apple M5 chip
- 16 GB Unified memory

The total training time is approximately 28 minutes, and the total evaluation time is approximately 5 minutes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's main contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work and potential future directions in Conclusion part.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All the theorems and lemmas are provided with a complete proof, and all assumptions are clearly stated in the main paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the experimental setup, including the training data used, the evaluation metrics, and the implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data are provided in the supplemental material, and the detailed instructions for reproducing the results are in the README file.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The training and test details are provided in the Experiments section, with all necessary information to understand the results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports error bars in the figures, showing the 25–75% interquartile range across MDPs, which provides information about the variability of the results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides experimental details in appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper focuses on theoretical analysis and has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is only used for writing, editing, or formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.