Factorized Spectral Representations for Reinforcement Learning

Junyi Wu
University of Washington
Seattle, WA, USA
junyiwu@uw.edu

Dan Li
University of Washington
Seattle, WA, USA
dli27@uw.edu

Abstract

Learning a compact model of the world from interaction data is central to sample-efficient deep reinforcement learning. Spectral representation methods have become the leading paradigm for representation learning in continuous control by taking a matrix view of the transition kernel, with state-action pairs on one side and next states on the other, and learning a low-rank factorization through self-supervised contrastive objectives. We take this view one step further. The transition kernel is naturally a three-mode tensor over states, actions, and next states, and a CANDECOMP/PARAFAC (CP) decomposition gives one feature map per mode. We propose FaStR, which fits this decomposition with a noise contrastive objective, producing separate state, action, and next-state encoders that together form a single spectral representation. The factored form yields a smaller hypothesis class, and under CP realizability the sufficient-sample upper bound for representation learning improves by a factor that scales with the smaller of the state and action dimensions. Empirically, FaStR delivers its largest gains on high-dimensional locomotion tasks whose dynamics align with the factored structure, and the learned state encoder transfers intact across actuator shift while only the action encoder is retrained.

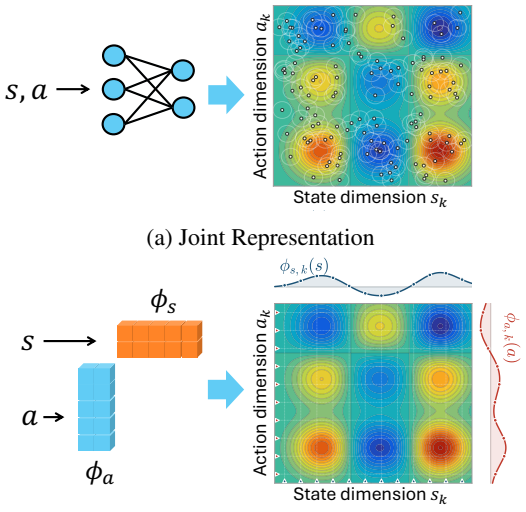
1 Introduction

Sample efficiency is a central concern in continuous-control RL. As state and action dimensions grow, the number of environment interactions required by model-free methods rises sharply, limiting real-world deployment [Fujimoto et al., 2018, Haarnoja et al., 2018]. One productive route is the framework of low-rank MDPs [Jin et al., 2020, Yang and Wang, 2020, Agarwal et al., 2020], which assumes the transition kernel admits a finite-rank factorization. The representation-learning problem then becomes one of finding a feature space in which the dynamics are linear. Spectral representation methods bring this low-rank view closer to practical deep RL [Gao et al., 2025]. They use transition samples from online interaction to learn transition-aware features, often without reward labels, and feed these features to policy and value learning. When such features are estimated more accurately from the same interaction budget, the critic receives a representation closer to the Bellman-linear structure earlier in training. The contrastive objective used to extract these features has been refined steadily across this line of work, and the resulting agents are now competitive on standard benchmarks.

RL representations preserve different kinds of information. Successor and forward–backward representations summarize long-term visitation and behavior [Dayan, 1993, Barreto et al., 2017, Touati and Ollivier, 2021]. Laplacian and proto-value methods describe the geometry of the transition graph [Mahadevan, 2005, Wu et al., 2019], while bisimulation, contrastive, and predictive methods learn state features from behavioral similarity or future information [Ferns et al., 2004, Zhang et al., 2021, Laskin et al., 2020, Stooke et al., 2021, Schwarzer et al., 2021]. Spectral RL instead represents the one-step transition itself, which makes how the state and the action enter the representation a design choice. In many control systems, states and actions play different roles. The state describes the

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

current configuration of the system, while the action describes the input driving the system. Existing spectral RL methods [Ren et al., 2023b, Zhang et al., 2022, Gao et al., 2025, Shribak et al., 2024, Nabati et al., 2026] implement low-rank structure by treating the state-action pair as a joint argument to the transition kernel. Separately, model-based methods learn explicit dynamics models for planning or imagination [Hafner et al., 2023, Hansen et al., 2024], while factored value-function methods place low-rank structure on the critic or the policy instead of the transition [Mahajan et al., 2021, Lobel et al., 2023, Rozada et al., 2024, 2025]. This work identifies a gap between these two directions: the lack of a factored low-rank representation for the transition dynamics themselves. The motivation stems from the observation that distinct actions often drive identical state-dependent motion patterns. In such cases, a joint representation must relearn shared state-side structure across actions, whereas a factored representation can reuse the same state-side modes while learning how actions modulate them.



(a) Joint Representation

(b) Factorized Representation

Figure 1: Comparison of representation structures. Factorization constrains the joint score landscape to the intersection of independent state and action profiles.

To this end, we introduce Factored Spectral Representations (FaStR), which treats the transition kernel as a three-way tensor and learns one encoder per mode (state, action, next-state) through a CANDECOMP/PARAFAC (CP) decomposition. Figure 1 illustrates the key difference. A joint encoder maps the concatenated pair (s, a) directly into a state-action representation, so the learned score can vary freely over the full joint plane (Figure 1a). In this view, local peaks in the score landscape may be tied to specific state-action regions. FaStR instead learns separate state and action profiles, and combines them through a Hadamard product (Figure 1b). Thus, peaks and variations in the joint score must arise from the interaction of reusable state-side and action-side factors, encouraging sharing across rows and columns of the state-action plane. This structured sharing acts as an implicit regularizer and shifts the statistical burden from covering isolated state-action regions to estimating reusable factors, allowing the frozen feature to become useful for Bellman backups with fewer transitions when the dynamics match the factorization. Specifically, our contributions are as follows:

1. **A factored view of spectral RL representations.** We formulate the transition kernel as a three-mode object over state, action, and next state, and instantiate this view as a representation with a separate encoder for each of these three modes. The factored representation keeps these modes explicit while remaining compatible with contrastive training.

2. **A smaller sufficient-sample upper bound under CP structure.** We prove that the factored hypothesis class has a covering number that decomposes additively over the state, action, and next-state factors. Under CP realizability and the corresponding regularity conditions, FaStR obtains a smaller sufficient-sample upper bound than the joint class, with an exponent gap of $\min(d_s, d_a)$; under approximate CP structure, the guarantee becomes a bias-estimation tradeoff.

3. **Empirical evaluation and modular transfer.** On DM Control Suite [Tassa et al., 2018] tasks across morphologies and action dimensions, we test when the factored representation outperforms a joint state-action encoder. FaStR obtains the highest mean return on all five tasks with $\dim \mathcal{A} \geq 21$ and matches or exceeds the joint encoder at lower action dimensions; Appendix E locates the boundary of this regime on seven further tasks. Under actuator shift, the state factor transfers intact and only the action factor is retrained.

2 Preliminaries

We consider a discounted MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with continuous state space $\mathcal{S}$, continuous action space $\mathcal{A}$, transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0, 1)$. The Q-function $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a]$ satisfies the Bellman equation

$$Q^\pi(s, a) = r(s, a) + \gamma \int_{\mathcal{S}} P(s'|s, a)\, V^\pi(s')\, ds', \tag{1}$$

where $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s, a)]$.

Linear MDPs and the linear-feature view. A common simplification is to assume the transition kernel admits a finite-rank factorization. The linear MDP assumption [Jin et al., 2020] states that there exist a feature map $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, a kernel map $\mu : \mathcal{S} \to \mathbb{R}^d$, and a reward parameter $\theta_r \in \mathbb{R}^d$ such that

$$P(s'|s, a) = \varphi(s, a)^\top \mu(s'), \qquad r(s, a) = \varphi(s, a)^\top \theta_r. \tag{2}$$

Under this assumption, the value function is linear in φ: for any policy π, $Q^\pi(s, a) = \varphi(s, a)^\top w^\pi$ for some $w^\pi \in \mathbb{R}^d$ that absorbs the integral against V^π. The complexity of value learning then depends on the feature dimension d rather than the size of the state-action space, which enables provably efficient algorithms [Jin et al., 2020, Uehara et al., 2022]. In practice φ is unknown and must be learned from interaction data.

Spectral RL: learning transition-ratio scores. Spectral representation methods learn φ and μ from transition samples using self-supervised contrastive objectives [Gutmann and Hyvärinen, 2010, van den Oord et al., 2018, Zhang et al., 2022, Gao et al., 2025]. A scoring function $h(s, a, s') = \varphi(s, a)^\top \mu(s')$ is trained to assign high values to true next-state samples $s' \sim P(\cdot|s, a)$ and low values to negatives drawn from a reference distribution. At the population level, such objectives fit a transition-ratio score: the logit measures how plausible a candidate next state is under the transition kernel relative to the negative distribution. Existing deep spectral methods parameterize this score bilinearly, using a fused state-action feature on one side and a next-state feature on the other. The trained φ can then be used as the linear-MDP feature for downstream value learning.

The contrastive template that FaStR will instantiate is the ranking-based perturbed noise contrastive estimation (RP-NCE) objective [Zhang et al., 2022, Gao et al., 2025], in which negatives are obtained by perturbing true next states across a sequence of noise levels and the model is trained to rank a true (perturbed) positive above $K - 1$ corrupted alternatives:

$$\ell_{\text{NCE}} = -\frac{1}{MN} \sum_{m=1}^{M} \sum_{i=1}^{N} \log \frac{\exp\big(\varphi_i^\top \mu(\tilde{s}'_{i,0}; \beta_m)\big)}{\sum_{j=0}^{K-1} \exp\big(\varphi_i^\top \mu(\tilde{s}'_{i,j}; \beta_m)\big)}, \tag{3}$$

where $\varphi_i = \varphi(s_i, a_i)$, $\{\beta_m\}_{m=1}^M$ are noise levels along a variance-preserving (VP) schedule [Song et al., 2021], which keeps the marginal variance fixed across levels, and $\tilde{s}'_{i,j}$ denotes the j-th candidate next state corrupted at level β_m, with $j = 0$ the true target and $j \geq 1$ negatives drawn from other transitions in the mini-batch (in-batch negatives). At the population level this ranking objective is consistent with maximum likelihood estimation as the number of negatives grows [Ma and Collins, 2018]. Thus RP-NCE is best viewed as an estimator for a chosen transition-ratio score class; FaStR keeps this estimator but changes the score class from a fused bilinear to a mode-wise trilinear form.

The full spectral RL pipeline. The contrastive loss is one component of the system. After training, the encoder φ is used as a frozen representation: a critic Q_θ takes $\varphi(s, a)$ as input and is trained by standard temporal-difference (TD) losses without back-propagating through φ, and a policy network π is trained on top of this frozen representation. This separation between representation learning (the contrastive loss) and value learning (TD on the frozen feature) is what makes the learned φ usable as the linear-MDP feature in Jin et al. [2020]. We work within this two-stage setup and locate our contribution in the structural form of the encoder φ.

3 Method

3.1 The FaStR Framework

A tensor view of the transition kernel. The standard linear MDP merges s and a into a single index and factorizes the resulting matrix $P \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}|}$ via an SVD-like decomposition. This treats the two arguments uniformly, although the state specifies the configuration of the system while the action specifies the input applied to it. We separate the two arguments by viewing the transition kernel as a third-way tensor $\mathcal{T} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|\times|\mathcal{S}|}$ indexed by s, a, and s'.[1]

A standard general decomposition of such a tensor is the Tucker form [Tucker, 1966, Kolda and Bader, 2009] $P(s'|s,a) = \sum_{i,j,k} G_{ijk}\, \phi_{s,i}(s)\, \phi_{a,j}(a)\, m_k(s')$ with a core $G \in \mathbb{R}^{d_s\times d_a\times d_m}$. The CP (CANDECOMP/PARAFAC) decomposition [Carroll and Chang, 1970, Harshman, 1970, Kolda and Bader, 2009] is the special case where G is superdiagonal, $G_{kkk} = 1$, eliminating the core entirely; lower-rank or block-diagonal cores interpolate between CP and dense Tucker. Among these, FaStR uses the CP form; the case for this choice is made below in the context of contrastive training.

Assumption 1 (CP-Factored MDP)**.** *There exist feature maps $\phi_s : \mathcal{S} \to \mathbb{R}^d$, $\phi_a : \mathcal{A} \to \mathbb{R}^d$, a kernel map $m : \mathcal{S} \to \mathbb{R}^d$, and a reward parameter $\theta_r \in \mathbb{R}^d$ such that for all (s, a, s'):*

$$P(s'|s,a) = \big(\phi_s(s) \odot \phi_a(a)\big)^\top m(s') = \sum_{k=1}^{d} \phi_{s,k}(s)\, \phi_{a,k}(a)\, m_k(s'), \tag{4}$$

$$r(s,a) = \big(\phi_s(s) \odot \phi_a(a)\big)^\top \theta_r. \tag{5}$$

We write $\psi(s,a) := \phi_s(s) \odot \phi_a(a) \in \mathbb{R}^d$ for the factored feature.

Each component k in Equation (4) is a rank-one term combining a state factor $\phi_{s,k}(s)$, an action factor $\phi_{a,k}(a)$, and a next-state factor $m_k(s')$, with the action factor scaling the coupling between the other two. CP fits well when actions scale shared state-dependent modes of motion, such as posture or balance in locomotion; it underfits when strong contact or limb-coupling effects entangle modes across actions. The CP form has scale invariances between ϕ_s, ϕ_a, and m, which are fixed by uniform norm bounds B_ϕ and $B_{m,\infty}$ throughout the analysis (Appendix C.1); the implementation uses LayerNorm to stabilize the scales.

Latent factorization versus coordinate-wise separability. The CP interaction in Assumption 1 is diagonal in the learned latent features, not in the physical state or action variables. Both ϕ_s and ϕ_a are deep networks: each latent factor can depend on all state (or action) variables, and a single physical action can contribute to many latent factors. The diagonal core removes only the terms in which latent state factor i meets latent action factor $j \neq i$. The assumption therefore requires that a low-rank CP model approximate the transition well in the learned latent space; it does not require independent physical actuators. Couplings that the encoders can reorganize are absorbed by them, and the remainder is the CP approximation error ϵ_{CP}. The diagonal form trades this approximation error for a smaller function class, and Proposition 3.2 states the tradeoff.

Factored representation and Q-linearity. FaStR learns a feature that is useful for control, not only for predicting the next state. Let $\psi(s,a) := \phi_s(s) \odot \phi_a(a)$. Under the CP-factored model, the transition and reward share this same feature: $P(s'|s,a) = \psi(s,a)^\top m(s')$ and $r(s,a) = \psi(s,a)^\top \theta_r$. Therefore, a Bellman backup does not leave the span of ψ: for any policy π, $Q^\pi(s,a)$ is derived as:

$$r(s,a) + \gamma \int_{\mathcal{S}} P(s'|s,a) V^\pi(s') ds' = \psi(s,a)^\top \left(\theta_r + \gamma \int_{\mathcal{S}} m(s') V^\pi(s') ds'\right) =: \psi(s,a)^\top w^\pi. \tag{6}$$

Thus the downstream linear critic is not an additional assumption: once the transition and reward admit the CP form, the corresponding Q-function is linear in the learned factored feature. A derivation with norm bounds is given in Appendix C.2.

[1] The tensor view is stated in the finite case for intuition; Assumption 1 below is stated directly through feature maps on general $\mathcal{S}$ and $\mathcal{A}$.

Learning a trilinear transition-ratio score. We learn the factors by viewing contrastive spectral learning as conditional ratio estimation. For an anchor (s_i, a_i), the observed next state is the positive candidate and perturbed in-batch next states are negatives. InfoNCE [van den Oord et al., 2018] turns this into a classification problem: among one true next state and several distractors, the model is trained to assign the highest energy to the true candidate. At the population optimum, differences between these energies estimate the log ratio between the transition law at the anchor and the negative candidate distribution. RP-NCE repeats this classification across noise levels β, giving a multi-scale ratio target for the next-state side. In standard spectral RL, this energy is bilinear: a fused state-action feature is matched with a β-conditioned next-state feature. FaStR instead chooses the CP trilinear energy

$$f_\Theta(s, a, s'; \beta) = \sum_{k=1}^{d} \phi_{s,\Theta,k}(s)\, \phi_{a,\Theta,k}(a)\, m_{\Theta,k}(s'; \beta) = \big(\phi_{s,\Theta}(s) \odot \phi_{a,\Theta}(a)\big)^\top m_\Theta(s'; \beta).$$

Each coordinate is a ratio channel: the state factor selects a state-dependent mode, the action factor gates it, and the next-state factor scores the candidate at noise level β. Substituting this energy into the RP-NCE template (3) gives

$$\ell_{\mathrm{FaStR}} = -\frac{1}{MN} \sum_{b=1}^{M} \sum_{i=1}^{N} \log \frac{\exp\big(f_\Theta(s_i, a_i, \tilde{s}'_{i,0}; \beta_b)\big)}{\sum_{j=0}^{K-1} \exp\big(f_\Theta(s_i, a_i, \tilde{s}'_{i,j}; \beta_b)\big)}. \tag{7}$$

The learned state-action feature is the anchor side of this energy, $\psi_\Theta(s, a) = \phi_{s,\Theta}(s) \odot \phi_{a,\Theta}(a)$. The downstream RFF critic then kernelizes the frozen ψ_Θ for value learning, while the mode-wise geometry of ψ_Θ is fixed by the transition-ratio energy. After this representation stage, we freeze ψ_Θ, train the actor and critic on top of it, and use the same downstream control pipeline as prior spectral RL; the structural change is in how transition data shape the state-action feature. Algorithm 1 gives the per-step update.

CP vs. Tucker cores. The CP form is the diagonal member of a broader trilinear transition-ratio score class, written as $h(s, a, s') = \phi_s(s)^\top M(s')\phi_a(a)$, with $M(s')$ generated by the next-state mode and restricted to be diagonal under CP. Other parameterizations correspond to less constrained interaction matrices: a dense $d_s \times d_a$ matrix gives full Tucker, and intermediate cases (low-rank $M(s') = U(s')V(s')^\top$, block-diagonal cores) interpolate between CP and dense Tucker. Such factorizations have been used in prior bilinear MDP methods trained with regression or maximum-likelihood objectives [Yang and Wang, 2020, Du et al., 2021, Lin et al., 2024], where the gradient on $M(s')$ is full-rank per anchor and large interaction matrices can be trained with enough data.

Under contrastive training the gradient structure is different. The chain rule on $h_{ij} = \phi_s(s_i)^\top M(s'_j)\phi_a(a_i)$ gives

$$\nabla_{\mathrm{vec}(M(s'_j))}\, \ell_{\mathrm{NCE}} \;\in\; \mathrm{span}\big\{\phi_a(a_i) \otimes \phi_s(s_i)\big\}, \tag{8}$$

a one-dimensional subspace per anchor. A mini-batch of N anchors therefore covers at most $\min(N, \dim(M))$ directions in the parameter space of $M(s')$ in a single update. This rank-one structure follows from the conditional bilinear form of the trilinear score and is the same for any parameterization of M: low-rank or block-diagonal cores reduce the parameter count below $d_s d_a$, but each anchor still contributes a rank-one direction projected onto the corresponding submanifold. For dense Tucker, $\dim(M) = d_s d_a$, so per-batch coverage requires $N \geq d_s d_a$, which scales quadratically in the per-mode dimension and exceeds typical batch sizes. The CP form has $\dim(M) = d$, so the linear requirement $N \geq d$ is met by standard mini-batch sizes.

Structured cores cannot satisfy encoder capacity, NCE validity ($N \geq d_s d_a$), and per-batch gradient coverage simultaneously at standard batch sizes; in the controlled ablation of Appendix D.1, no Tucker rank within this constraint matches CP under matched compute.

3.2 Provable Representation Efficiency

Section 3.1 chose CP among bilinear parameterizations on training-dynamics grounds. The matching statistical question is whether CP also makes the representation class easier to learn from a finite sample. We answer it in three steps. We first bound the size of the factored class relative to the joint

class (Theorem 3.1). We then convert this bound into a finite-sample bound on representation error (Proposition 3.2). Finally, we invert the bound to compare the sample sizes that suffice for each class to reach a target error (Proposition 3.3). The dimension dependence of the final result matches the empirical pattern in Section 4.

We work with two scoring classes. The factored class $\mathcal{G}_{\text{fac}}$ contains $g(s,a,s') = (\phi_s(s) \odot \phi_a(a))^\top m(s')$ with $(\phi_s, \phi_a, m) \in \mathcal{F}_s \times \mathcal{F}_a \times \mathcal{F}_m$; the joint class $\mathcal{G}_{\text{joint}}$ contains $g(s,a,s') = \varphi(s,a)^\top \mu(s')$ with $\varphi \in \mathcal{F}_{sa}$, $\mu \in \mathcal{F}_m$. Norm bounds and the convention for sup-ℓ_2 covering numbers are in Appendix C.1.

Step 1: bounding the size of the factored class. A monolithic encoder takes each (s,a) pair as a single input, and the size of its hypothesis class scales with the combined state-action dimension. The CP parameterization splits the score into three networks: a state factor, an action factor, and a kernel encoder, coupled through a diagonal trilinear interaction (the Hadamard form of Eq. 4). The size of the resulting class is then controlled by the three component networks separately rather than by their product. Covering numbers, a standard measure of the size of a function class, make this precise.

Theorem 3.1 (Covering-number decomposition)**.** *For every $\epsilon > 0$,*

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \;\le\; \log \mathcal{N}\Big(\mathcal{F}_s, \tfrac{\epsilon}{3B_{m,\infty}B_\phi}\Big) + \log \mathcal{N}\Big(\mathcal{F}_a, \tfrac{\epsilon}{3B_{m,\infty}B_\phi}\Big) + \log \mathcal{N}\Big(\mathcal{F}_m, \tfrac{\epsilon}{3B_\phi^2}\Big).$$

If $\mathcal{F}_{sa}$ contains every Hadamard product $\phi_s \odot \phi_a$ with $(\phi_s, \phi_a) \in \mathcal{F}_s \times \mathcal{F}_a$, then $\mathcal{G}_{\text{fac}} \subseteq \mathcal{G}_{\text{joint}}$, and the inclusion holds for both covering numbers and Rademacher complexities at every sample size.

Proof. See Appendix C.3. □

For Lipschitz encoders on compact domains, this gives an exponent $D_{\max} = \max(d_s, d_a)$ for the factored class versus $D_{\text{sum}} = d_s + d_a$ for the joint class, leaving a gap of $\min(d_s, d_a)$ in the rate exponent that is formalized as a comparison of sufficient sample sizes in Proposition 3.3. Once $\psi = \phi_s \odot \phi_a$ is frozen, TD on ψ satisfies the linear-MDP sample-complexity guarantees of Jin et al. [2020], so a smaller bound on the representation error at a given n becomes a smaller bound on the return gap in policy learning; Appendix C.9 reads the bound on Humanoid.

Step 2: from class size to representation error. A smaller class size matters only if it leads to a smaller generalization gap. The target we analyze is the L_2 representation objective of Ren et al. [2023b], the squared $L^2(\nu)$ distance from the true transition density,

$$L(g) := \mathbb{E}_\rho\left[\int_{\mathcal{S}} \big(P(s' \mid s,a) - g(s,a,s')\big)^2 d\nu(s')\right].$$

We analyze this objective rather than NCE itself because finite-sample guarantees are tractable for the L_2 surrogate at the population level, while the contrastive loss serves as its practical proxy at training time [Ren et al., 2023b].

Proposition 3.2 (Finite-sample generalization)**.** *Let $\mathcal{G} \in \{\mathcal{G}_{\text{fac}}, \mathcal{G}_{\text{joint}}\}$ with $|g| \le B_g$, and let $\hat{g}_n$ be an empirical minimizer of the centered objective on n i.i.d. transitions. With probability at least $1-\delta$,*

$$L(\hat{g}_n) \;\le\; \underbrace{L(g^*_{\mathcal{G}})}_{\textit{approximation}} \;+\; \underbrace{8\,\mathcal{R}_n(\mathcal{G}) + 4\,\mathcal{R}_n(\mathcal{H}_{\mathcal{G}})}_{\textit{estimation}} + O\left((B_g + B_g^2)\sqrt{\tfrac{\log(1/\delta)}{n}}\right),$$

where $\mathcal{H}_{\mathcal{G}} := \{(s,a) \mapsto \int g^2(s,a,u)\, d\nu(u) : g \in \mathcal{G}\}$ is the integrated quadratic class arising from expanding the squared loss. The quadratic class satisfies $\mathcal{N}(\mathcal{H}_{\mathcal{G}}, \epsilon) \le \mathcal{N}(\mathcal{G}, \epsilon/(2B_g))$, so its Rademacher complexity inherits the rate of $\mathcal{G}$ itself.

Proof. See Appendix C.6. □

The bound has two terms with opposite scaling. The approximation term $L(g^*_{\mathcal{G}})$ measures whether the class contains the true dynamics; under exact CP realizability and the inclusion of Theorem 3.1, both classes contain the truth and this term is zero for both. Under approximate CP, the factored class has an irreducible bias $\epsilon^2_{\text{CP}} := \inf_{g \in \mathcal{G}_{\text{fac}}} L(g)$ that the joint class does not have. The estimation term is smaller for the factored class: both Rademacher terms inherit the smaller exponent $D_{\max}$ from Theorem 3.1, so the factored certificate decays faster in n.

Step 3: from representation error to sample size. The previous two steps bound representation error at a fixed sample size. The inverse question is how many samples each class needs to reach the same target error.

Proposition 3.3 (Sufficient-sample-size separation)**.** *Assume exact CP realizability, the inclusion condition of Theorem 3.1, and L-Lipschitz encoder classes on compact domains with $D_{\max} \geq 3$. Proposition 3.2 then gives certificates of the form $L(\hat{g}_n^{\mathrm{fac}}) \leq C_{\mathrm{fac}}(\delta)\, n^{-1/D_{\max}}$ and $L(\hat{g}_n^{\mathrm{joint}}) \leq C_{\mathrm{joint}}(\delta)\, n^{-1/D_{\mathrm{sum}}}$, with $C_{\mathrm{fac}}, C_{\mathrm{joint}}$ depending polynomially on the norm bounds, Lipschitz constants, and feature dimension, and logarithmically on $1/\delta$. Let $n_{\mathrm{fac}}^{\mathrm{suff}}(\epsilon)$ and $n_{\mathrm{joint}}^{\mathrm{suff}}(\epsilon)$ be the smallest n at which each certificate guarantees error at most ϵ^2. Then*

$$\frac{n_{\mathrm{joint}}^{\mathrm{suff}}(\epsilon)}{n_{\mathrm{fac}}^{\mathrm{suff}}(\epsilon)} \;=\; \Theta\Big(\epsilon^{-2\min(d_s,d_a)}\Big) \quad \textit{as } \epsilon \to 0.$$

Proof. See Appendix C.7. □

Under CP realizability and the corresponding regularity conditions, FaStR therefore obtains a smaller sufficient-sample upper bound, with the exponent gap equal to $\min(d_s, d_a)$; Appendix C.7 details the scope of this upper-bound comparison. This gap is meaningful when the smaller of the state and action dimensions is nontrivial and the CP bias is small; otherwise the approximate-CP certificate $L(\hat{g}_n^{\mathrm{fac}}) \lesssim \epsilon_{\mathrm{CP}}^2 + C_{\mathrm{fac}}(\delta) n^{-1/D_{\max}}$ can be dominated by bias. We test these two conditions in Section 4 and Appendix E.1. Appendix C.8 further gives a conditional LSVI-UCB implication: if the learned feature satisfies approximate Bellman linearity, the factored structure can reduce the representation cost of obtaining features suitable for optimism-driven exploration.

4 Experiments

We compare FaStR against the monolithic encoder under an otherwise identical pipeline on locomotion benchmarks (Section 4.2), and test whether the separable state factor enables adaptation under actuator shift (Section 4.3).

4.1 Setup

We evaluate on DM Control Suite [Tassa et al., 2018] locomotion tasks that vary in action dimensionality and morphology, with $\dim(\mathcal{A}) \in \{6, 12, 21, 38\}$ and shared proprioceptive observations. To attribute performance differences to the encoder structure, FaStR and the monolithic baseline CTRL-SR [Gao et al., 2025] share the same TD3 backbone, hidden width, representation dimension, learning rates, noise schedule, and training budget, with no task-specific tuning. The only varying component is the representation: FaStR's factored encoders (ϕ_s, ϕ_a, m) versus CTRL-SR's joint encoder $\varphi([s; a])$. We align hidden width rather than parameter count, and Appendix D.2 repeats the comparison at three widths. Figure 2 compares FaStR against CTRL-SR. Table 1 adds Diff-SR [Shribak et al., 2024], a spectral baseline that replaces the contrastive objective with a diffusion next-state model on the same TD3 backbone and training hyperparameters; the model-free methods TD7 [Fujimoto et al., 2023] and SAC [Haarnoja et al., 2018]; and the model-based TD-MPC2 [Hansen et al., 2024] in its official 5M-parameter single-task configuration. TD7 uses its authors' recommended hyperparameters, and SAC(3-layer) uses the hidden width of the spectral methods. All methods share the environment-interaction budget (10^6 frames), action repeat, evaluation protocol, and random seeds, and all results are reported over 8 seeds. Full hyperparameters are in Appendix B.

Task selection. We evaluate 5 DMControl domains with action dimensions of at least 6, spanning action dimensions from 6 to 38. Within these domains, we include representative standing, walking, and running tasks. Under our computational budget, this provides broad coverage of action dimensions and morphologies without evaluating every reward variant. The task set was fixed before the experiments, and neither training stability nor performance influenced the selection. Seven additional official tasks with $\dim \mathcal{A} \leq 5$ are reported in Appendix E.2, and Table 5 lists every evaluated task.

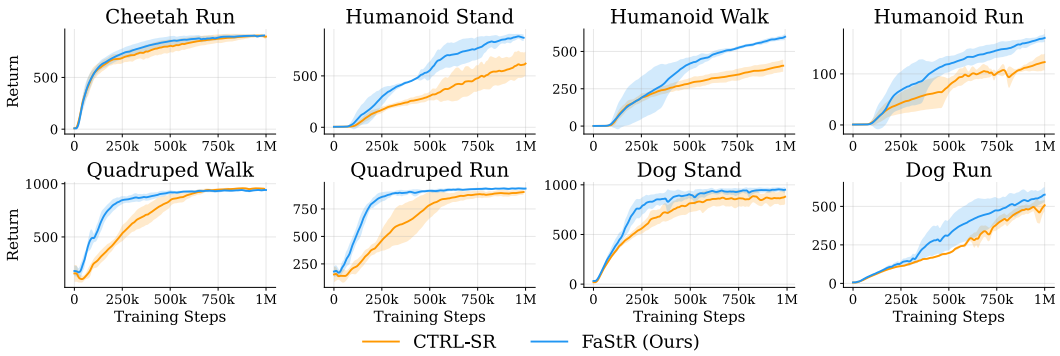


Figure 2: Learning curves across 8 DM Control Suite tasks spanning four morphologies. FaStR's gains are largest on tasks whose dynamics admit a CP factorization, consistent with the bias-estimation view. Shaded: ± 1 std across 8 seeds.

4.2 Sample Efficiency on Locomotion Benchmarks

On the Humanoid tasks, where torso-limb morphology admits a CP factorization (actions scaling shared postural modes in the learned latent space), FaStR exceeds the controlled baseline CTRL-SR by 106, 197, and 56 points on Stand, Walk, and Run, each margin larger than the across-seed standard deviation of either method. On Dog-Stand, FaStR reaches 955 ± 20 against 870 ± 70. On Dog-Run, where four limbs move in coordination and ground contact couples state and action more strongly, the improvement over the joint encoder is smaller than on the other high-dimensional tasks (538 ± 49 vs. 505 ± 97). FaStR also exceeds Diff-SR on all five tasks with $\dim \mathcal{A} \geq 21$ (Table 1), so the gain is not specific to the contrastive objective. TD-MPC2, which learns a compact world model for planning, obtains the highest return on Cheetah-Run and Walker-Walk, while FaStR obtains the highest mean return on all five 21- and 38-dimensional tasks: full dynamics modeling pays off in lower-dimensional control, and a low-rank transition factorization is the more effective target on this suite as the action dimension grows.

On the Quadruped tasks, FaStR reaches CTRL-SR's converged return earlier in training: on Quadruped-Run it then exceeds it (941 ± 14 vs. 910 ± 18), and on Quadruped-Walk the final returns coincide (952 ± 10 vs. 954 ± 8). The estimation advantage on this family appears as faster convergence, the rate at which the representation becomes usable for value learning. On Cheetah-Run, Walker-Walk, and Walker-Run ($\dim \mathcal{A} \leq 6$), FaStR matches the spectral baselines, the regime predicted by Proposition 3.3 in which small $\min(d_s, d_a)$ gives a small estimation advantage in the certificate.

Table 1: Final-10-evaluation mean return on DM Control Suite tasks with proprioceptive observations, averaged across 8 seeds (± 1 std). Bold: best mean return in each column.

	$\dim \mathcal{A} = 6$			$\dim \mathcal{A} = 12$		$\dim \mathcal{A} = 21$			$\dim \mathcal{A} = 38$	
	Cheetah Run	Walker Walk	Walker Run	Quad. Walk	Quad. Run	Hum. Stand	Hum. Walk	Hum. Run	Dog Stand	Dog Run
SAC(2-layer)	371 ± 98	484 ± 12	665 ± 23	717 ± 54	777 ± 52	223 ± 84	170 ± 10	40 ± 65	500 ± 9	39 ± 66
SAC(3-layer)	885 ± 41	961 ± 27	792 ± 26	872 ± 81	884 ± 105	428 ± 64	229 ± 142	88 ± 30	577 ± 11	67 ± 34
TD7	909 ± 31	979 ± 1	813 ± 17	952 ± 3	875 ± 114	660 ± 230	334 ± 208	80 ± 45	840 ± 128	172 ± 51
TD-MPC2	$\mathbf{933 \pm 5}$	$\mathbf{988 \pm 4}$	830 ± 12	953 ± 11	919 ± 7	673 ± 13	589 ± 37	161 ± 24	799 ± 40	181 ± 33
Diff-SR	910 ± 4	978 ± 2	$\mathbf{833 \pm 15}$	953 ± 5	903 ± 27	761 ± 129	440 ± 93	107 ± 35	909 ± 27	310 ± 94
CTRL-SR	898 ± 14	978 ± 1	811 ± 25	$\mathbf{954 \pm 8}$	910 ± 18	752 ± 94	394 ± 37	118 ± 13	870 ± 70	505 ± 97
FaStR (Ours)	899 ± 13	975 ± 3	814 ± 22	952 ± 10	$\mathbf{941 \pm 14}$	$\mathbf{858 \pm 26}$	$\mathbf{591 \pm 15}$	$\mathbf{174 \pm 19}$	$\mathbf{955 \pm 20}$	$\mathbf{538 \pm 49}$

Mechanism check. In a controlled ablation (Appendix D.1), separate encoders feeding a joint MLP improve only slightly on the monolithic baseline, and a full Tucker core trains worse under NCE because of the rank-deficient gradient of Section 3.1: the CP interaction is the source of the gain. The offline CP-misfit diagnostic (Appendix E.1) shows that gains track structural alignment with the dynamics, not action dimension alone. The ranking on the Humanoid tasks is unchanged across encoder widths of 256, 512, and 1024 (Appendix D.2). On seven further tasks with $\dim \mathcal{A} \leq 5$,

FaStR is comparable to CTRL-SR on five and below it on Hopper-Hop and Acrobot-Swingup (Appendix E.2).

4.3 Modular Reuse of ϕ_s under Actuator Shift

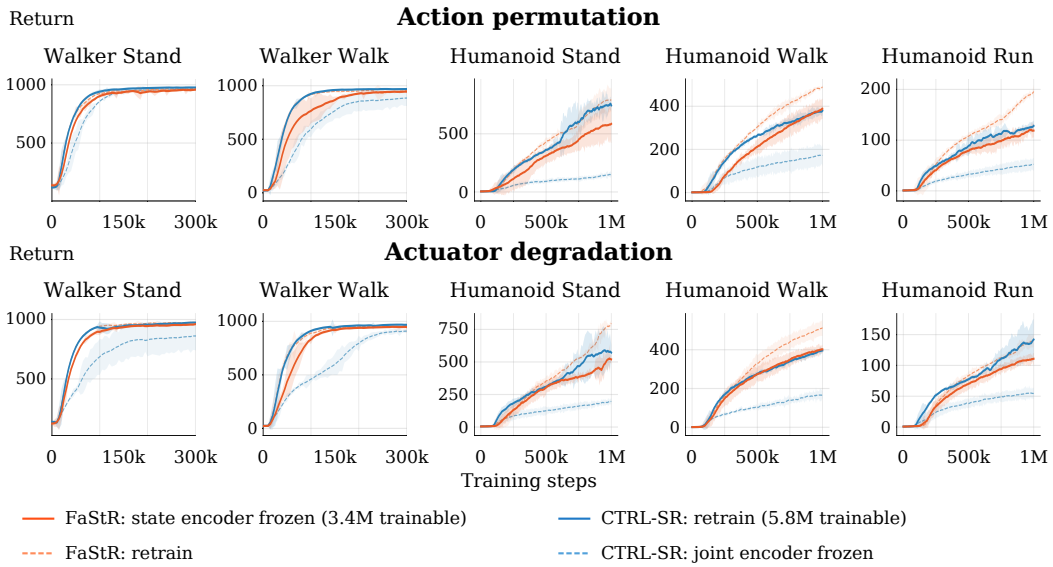


Figure 3: Architectural diagnostic under actuator shift. Top: action permutation; bottom: actuator degradation. Walker ($\dim \mathcal{A}$=6) is a pre-shift parity check; Humanoid ($\dim \mathcal{A}$=21) gives the diagnostic. **What is preserved** (red solid vs. blue dashed): under the same freeze-and-adapt protocol, FaStR's ϕ_s remains usable across the shift while CTRL-SR's φ does not. **At what cost** (red solid vs. blue solid): on Humanoid, FaStR with 18% of its parameters held fixed in ϕ_s reaches the same final return as CTRL-SR retrained from scratch with all parameters trainable; trainable parameter counts for the two solid curves are reported in the legend. Orange dashed (FaStR retrained from scratch) is a reference upper bound. Shaded: ± 1 std across 8 seeds. See Appendix B for parameter setting.

We test transfer in a setting where the physical dynamics of the system are unchanged but the action-to-torque interface is rewritten. The CP factorization predicts what each architecture preserves under such a change. Under FaStR, ϕ_s depends only on the state-side dynamics of the body and should remain valid when only the action-to-torque map shifts; under a joint encoder, state and action are coupled through every parameter, so any change to the interface leaks throughout the encoder and no component is guaranteed to remain valid. The factored representation therefore admits an option a joint encoder does not: keep ϕ_s, adapt only ϕ_a.

We instantiate two such rewrites. Actuator degradation applies independent per-joint gain scaling and polarity inversion. Action permutation randomly reassigns joint channel indices. The transition over true torques and the reward are both unchanged; only the path from policy output to actuation is rewritten. Source and target therefore share the same task, reward, and body, and differ only in the actuator interface, so any retraining cost reflects what the representation failed to preserve. For each architecture we freeze the largest state-dependent component (ϕ_s for FaStR, the joint encoder φ for CTRL-SR) and reinitialize everything downstream. Walker is a parity check on matched pre-shift returns; Humanoid is the diagnostic.

What is preserved, and at what cost. On a CP-factored MDP, which component to keep across an actuator shift is determined by the representation, not a hyperparameter the practitioner has to choose. A monolithic encoder gives no such handle: the only options are to retrain everything or to freeze arbitrary subsets, neither of which uses structural information about the shift. Reusing the frozen ϕ_s costs no final return: on Humanoid-Walk, FaStR with ϕ_s frozen matches CTRL-SR retrained from scratch with all parameters trainable (393.7 vs. 394.1 under permutation, 401.8 vs. 399.2 under degradation; Figure 3).

5 Related Work

State and temporal representations. Many RL methods ask what information a representation should preserve, rather than modeling the transition itself. Successor representations describe which states a policy is likely to visit in the future, and successor features make this information reusable across related reward functions [Dayan, 1993, Barreto et al., 2017]. Forward–backward representations describe whether a state–action pair can reach a future state [Touati and Ollivier, 2021]. Laplacian and proto-value methods capture the geometry of the transition graph [Mahadevan, 2005, Wu et al., 2019], while bisimulation methods group states with similar rewards and transition behavior [Ferns et al., 2004, Zhang et al., 2021]. Contrastive and predictive methods learn state features by matching related observations or predicting future ones [Laskin et al., 2020, Stooke et al., 2021, Schwarzer et al., 2021]. TACO and SALE incorporate actions more explicitly, but still use the learned representations mainly to improve policy or value learning [Zheng et al., 2023, Fujimoto et al., 2023].

Transition representations. Another line of work learns representations directly from the one-step transition. Linear MDP and bilinear-class theory show that low-dimensional transition features can support value estimation and exploration, while FLAMBE and Uehara et al. [2022] study how such features can be recovered from data [Jin et al., 2020, Du et al., 2021, Agarwal et al., 2020, Uehara et al., 2022]. Spectral RL provides practical algorithms for learning these representations. CTRL and SPEDER use contrastive objectives [Zhang et al., 2022, Ren et al., 2023b]. LV-Rep models transitions through latent variables [Ren et al., 2023a], Diff-SR learns from noise-perturbed next states [Shribak et al., 2024], and CTRL-SR combines perturbation with ranking-based contrastive learning [Gao et al., 2025]. Existing methods learn a joint representation of the state–action pair. FaStR instead studies whether transition dynamics admit a compact low-rank structure with separate state and action factors.

Low-rank structure in RL. Low-rank assumptions have also been introduced elsewhere in reinforcement learning. Ni et al. [2023] model the transition as a Tucker tensor learned from batch data for abstraction and policy evaluation. Other work instead imposes low-rank structure on value functions or continuous-action critics and policies [Mahajan et al., 2021, Lobel et al., 2023, Rozada et al., 2024, 2025]. FaStR studies a different question: whether a CP structure already exists in the transition representation itself.

Dynamics representations for planning. A different line of work also learns from transition dynamics, but aims to reconstruct the dynamics themselves. Early model-based methods learn predictive dynamics models for planning or policy optimization [Janner et al., 2019, Hafner et al., 2020]. More recent world-model methods such as DreamerV3 and TD-MPC2 learn latent dynamics models for imagination or model predictive control [Hafner et al., 2023, Hansen et al., 2024]. FaStR instead learns only the transition representation needed for value learning, without reconstructing the full dynamics or performing planning.

6 Discussion and Conclusion

We introduce FaStR, a spectral representation learning framework that decomposes the transition kernel as a three-way tensor with state, action, and next-state factors, estimated through a trilinear structured contrastive objective. Under CP realizability and the corresponding regularity conditions, the factored class obtains a smaller sufficient-sample upper bound: transition samples constrain reusable state and action factors rather than isolated state-action pairs, so the frozen feature can become useful for Bellman backups earlier in training when the CP structure matches the dynamics. The same separation also supports modular adaptation: when the action-to-torque mapping shifts, the state factor transfers intact while only the action factor needs to be retrained. FaStR is most useful when the transition compresses into a small number of reusable latent modes and the state or action dimension is large enough for this reduction to matter; Appendix E maps both conditions empirically. The broader principle is that representations should structurally mirror the physical factorization of future embodied systems into sensors, actuators, and the dynamics connecting them.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation under CAREER Award No. 2622141. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Competing interests: The authors declare no competing interests.

References

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. FLAMBE: Structural complexity and representation learning of low rank MDPs. In Advances in Neural Information Processing Systems, 2020.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In Advances in Neural Information Processing Systems, 2017.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3:463–482, 2002.

J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika, 35(3):283–319, 1970. doi: 10.1007/BF02310791.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. Neural Computation, 5(4):613–624, 1993.

Simon S. Du, Sham M. Kakade, Jason D. Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. In Proceedings of the International Conference on Machine Learning, 2021.

R. M. Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. Journal of Functional Analysis, 1(3):290–330, 1967. doi: 10.1016/0022-1236(67)90017-1.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pages 162–169, 2004.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Machine Learning, 2018.

Scott Fujimoto, Wei-Di Chang, Edward J. Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For SALE: State-action representation learning for deep reinforcement learning. In Advances in Neural Information Processing Systems, 2023.

Chenxiao Gao, Haotian Sun, Na Li, Dale Schuurmans, and Bo Dai. Spectral representation-based reinforcement learning. arXiv preprint arXiv:2512.15036, 2025.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, Proceedings of the International Conference on Artificial Intelligence and Statistics, volume 9 of Proceedings of Machine Learning Research, pages 297–304. PMLR, 2010.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In International Conference on Learning Representations, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. arXiv preprint arXiv:2301.04104, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In International Conference on Learning Representations, 2024.

Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics 16, University of California, Los Angeles, 1970.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In Advances in Neural Information Processing Systems, volume 32, 2019.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. In Proceedings of the Conference on Learning Theory, 2020.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. SIAM Review, 51(3): 455–500, 2009. doi: 10.1137/07070111X.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In Proceedings of the 37th International Conference on Machine Learning, 2020.

Haohong Lin, Wenhao Ding, Jian Chen, Laixi Shi, Jiacheng Zhu, Bo Li, and Ding Zhao. BECAUSE: Bilinear causal representation for generalizable offline model-based reinforcement learning. In Advances in Neural Information Processing Systems, 2024.

Samuel Lobel, Sreehari Rammohan, Bowen He, Shangqun Yu, and George Konidaris. Q-functionals for value-based continuous control. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 8932–8939, 2023.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. arXiv preprint arXiv:1809.01812, 2018.

Sridhar Mahadevan. Proto-value functions: Developmental reinforcement learning. In Proceedings of the 22nd International Conference on Machine Learning, pages 553–560, 2005.

Anuj Mahajan, Mikayel Samvelyan, Lei Mao, Viktor Makoviychuk, Animesh Garg, Jean Kossaifi, Shimon Whiteson, Yuke Zhu, and Animashree Anandkumar. Tesseract: Tensorised actors for multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, volume 139, pages 7301–7312. PMLR, 2021.

Colin McDiarmid. On the method of bounded differences. In J. Siemons, editor, Surveys in Combinatorics, 1989, volume 141 of London Mathematical Society Lecture Note Series, pages 148–188. Cambridge University Press, Cambridge, 1989.

Ofir Nabati, Bo Dai, Shie Mannor, and Guy Tennenholtz. Spectral Bellman method: Unifying representation and exploration in RL. In International Conference on Learning Representations, 2026.

Chengzhuo Ni, Yaqi Duan, Munther Dahleh, Mengdi Wang, and Anru R. Zhang. Learning good state and action representations for Markov decision process via tensor decomposition. Journal of Machine Learning Research, 24, 2023.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In Advances in Neural Information Processing Systems, 2007.

Tongzheng Ren, Chenjun Xiao, Tianjun Zhang, Na Li, Zhaoran Wang, Sujay Sanghavi, Dale Schuurmans, and Bo Dai. Latent variable representation for reinforcement learning. In International Conference on Learning Representations, 2023a.

Tongzheng Ren, Tianjun Zhang, Lisa Lee, Joseph E. Gonzalez, Dale Schuurmans, and Bo Dai. Spectral decomposition representation for reinforcement learning. In International Conference on Learning Representations, 2023b.

Sergio Rozada, Santiago Paternain, and Antonio G. Marques. Tensor and matrix low-rank value-function approximation in reinforcement learning. IEEE Transactions on Signal Processing, 72: 1634–1649, 2024.

Sergio Rozada, Hoi-To Wai, and Antonio G. Marques. Multilinear tensor low-rank approximation for policy-gradient methods in reinforcement learning. IEEE Transactions on Signal Processing, 73: 4906–4920, 2025.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In International Conference on Learning Representations, 2021.

Dmitry Shribak, Chen-Xiao Gao, Yitong Li, Chenjun Xiao, and Bo Dai. Diffusion spectral representation for reinforcement learning. In Advances in Neural Information Processing Systems, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In International Conference on Learning Representations, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In Proceedings of the 38th International Conference on Machine Learning, 2021.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. arXiv preprint arXiv:1801.00690, 2018.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In Advances in Neural Information Processing Systems, volume 34, 2021.

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. Psychometrika, 31(3): 279–311, 1966. doi: 10.1007/BF02289464.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline RL in low-rank MDPs. In International Conference on Learning Representations, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Martin J. Wainwright. High-Dimensional Statistics: A Non-Asymptotic Viewpoint. Cambridge University Press, 2019.

Yifan Wu, George Tucker, and Ofir Nachum. The Laplacian in RL: Learning representations with efficient approximations. In International Conference on Learning Representations, 2019.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In Proceedings of the International Conference on Machine Learning, 2020.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In International Conference on Learning Representations, 2021.

Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph E. Gonzalez, Dale Schuurmans, and Bo Dai. Making linear MDPs practical via contrastive representation learning. In Proceedings of the International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 26447–26466. PMLR, 2022.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In Advances in Neural Information Processing Systems, 2023.

A Limitations

The CP form $\phi_s(s) \odot \phi_a(a)$ treats the state and the action as two atomic axes. Environments with strongly coupled joints and tight state–action interdependence, such as Ant-style locomotion, fall outside the regime this two-way split captures cleanly, since the encoder must absorb all sub-axis interactions internally: on Hopper-Hop (contact switching) and Acrobot-Swingup (strong underactuated coupling), FaStR returns are below those of the joint encoder (Appendix E.2). A hierarchical factorization over actuator or body-part groups, e.g. $\phi_s(s) = \phi_{\text{torso}} \odot \phi_{\text{limbs}}$ or a tree-shaped CP, fits inside the same framework and is a clean next step. Second, diagonal M corresponds to CP rank one along the next-state axis. A low-rank Tucker core $G \in \mathbb{R}^{d \times d}$ recovers more interaction terms but enlarges the effective NCE dimension from d to d^2, reintroducing the constraint that motivated the diagonal choice. Mapping the rank–capacity trade-off, e.g. via low-rank rather than diagonal M, would close the gap between the CP and Tucker ends of this spectrum. Third, all experiments are conducted in the DMControl proprioceptive suite. Pixel-input variants and physical-robot deployment are not evaluated; the latter introduces sensor noise, latency, and actuator drift that the actuator-shift experiments only partially approximate. The evidence therefore supports the representation-learning advantages of FaStR in controlled simulation and does not replace validation on real robotic platforms; simulation experiments on legged robots such as Unitree Go2 or G1 are the next step. Finally, a limitation of the current evaluation is that, due to computational constraints, we did not evaluate the proposed method on every official DMControl task. Future work will extend the evaluation to the remaining tasks.

B Implementation Details

This appendix specifies the full training and evaluation protocol used in Section 4, together with the architectural and optimization hyperparameters of FaStR and the six baselines reported in Table 1: CTRL-SR [Gao et al., 2025], Diff-SR [Shribak et al., 2024], SAC [Haarnoja et al., 2018] in two width-depth variants (3-layer and 2-layer), TD7 [Fujimoto et al., 2023], and TD-MPC2 [Hansen et al., 2024]. Hyperparameters that are common across methods (environment configuration, replay, evaluation cadence, total interaction budget) are listed once in Section B.1; method-specific settings are listed in Sections B.2–B.7 and summarized side-by-side in Table 2. We follow the convention of reporting every hyperparameter that was set in our runs, including those left at the original authors' defaults, so that the protocol is self-contained.

B.1 Shared Training Protocol

All experiments use the DeepMind Control Suite [Tassa et al., 2018] with proprioceptive observations, action repeat (frame skip) of 2, and episode length 1000 environment steps. We train each agent for 1×10^6 environment frames (i.e. 5×10^5 agent steps after action repeat) per task. Each agent maintains a single uniform-sampling replay buffer of capacity 1×10^6 transitions, with the first $10{,}000$ frames collected by a uniform random policy as a warm start before any gradient update. Evaluation is performed every $10{,}000$ environment frames by rolling out the deterministic actor (for TD3-based methods: TD3 mean action; for SAC: tanh-Gaussian mean) for 10 episodes; we report the mean undiscounted return over the last 10 evaluation checkpoints, averaged over 8 random seeds, with shaded bands or ±1 standard deviations across seeds. Observation normalization (running mean and standard deviation) is enabled for the spectral methods (FaStR and CTRL-SR), following Gao et al. [2025], and disabled for the SAC and TD7 baselines, matching their original implementations. All methods use the discount factor γ=0.99, the same set of random seeds, and a single update per environment step (update-to-data ratio 1).

B.2 FaStR (Ours)

Architecture. All encoders, the noise schedule, and design defaults that are not explicitly varied below follow Gao et al. [2025]. Encoders are ResidualMLPs with LayerNorm, ELU activations, and Mish activations inside the contrastive head. The state factor ϕ_s maps $\dim(\mathcal{S}) \to 512 \to 512 \to d$; the action factor ϕ_a maps $\dim(\mathcal{A}) \to 512 \to 512 \to d$; the kernel encoder m maps $\dim(\mathcal{S})+128 \to 512 \to 512 \to d$, where the additional 128 dimensions encode the perturbation level via a sinusoidal time embedding. We set d=512 throughout. The factored feature $\psi(s,a) = \phi_s(s) \odot \phi_a(a)$ is

fed to a twin Q critic consisting of a random Fourier features (RFF) [Rahimi and Recht, 2007] projection of width 1024 followed by a 2-layer MLP head $[512]$ with LayerNorm and ELU; this critic is architecturally identical to the CTRL-SR critic so that the only varying component is the encoder. The actor is a deterministic policy network with hidden sizes $[512, 512, 512]$, ELU activations, and a final $\tanh$ that maps to $[-1, 1]^{\dim(\mathcal{A})}$.

Optimization. All four networks (state factor, action factor, kernel encoder, critic) are trained with Adam (default $(\beta_1, \beta_2) = (0.9, 0.999)$, no weight decay). Critic and actor learning rates are $3{\times}10^{-4}$; the representation learning rate (jointly applied to ϕ_s, ϕ_a, m) is $1{\times}10^{-4}$. Polyak averaging (i.e., $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$) with $\tau{=}0.005$ is used for both the critic and the representation targets; actor and critic updates are performed every environment step (`actor_update_freq`=1). The mini-batch size is $N{=}512$, matching the negative-sample count $K{=}N{=}512$ implicit in the in-batch RP-NCE objective. Exploration uses Gaussian action noise $\mathcal{N}(0, \sigma{=}0.2)$ clipped to $[-1, 1]^{\dim(\mathcal{A})}$; target action smoothing uses $\mathcal{N}(0, \tilde{\sigma}{=}0.2)$ clipped to $[-0.3, 0.3]$. Gradients of the critic loss are not back-propagated through the representation (`back_critic_grad`=false). One training step is summarized in Algorithm 1.

Representation objective. We use the ranking-based perturbed NCE loss (7) to train the trilinear transition-ratio score, with $M{=}25$ noise levels on a variance-preserving (VP) schedule. The ranking variant is enabled (`ranking`=true), and the auxiliary bilinear-MDP regression term of Gao et al. [2025] is kept at its default weight $\beta{=}1.0$. The combined training loss is $\ell_{\text{FaStR}} + \lambda_r\, \ell_{\text{reward}} + \lambda_Q\, \ell_{\text{critic}}$ with $\lambda_{\text{ctrl}}{=}\lambda_r{=}\lambda_Q{=}1.0$; the reward head is a 2-layer MLP $[512]$ on top of ψ, identical in structure to the CTRL-SR reward head.

Algorithm 1 FaStR: one training step. $\mathrm{sg}(\cdot)$: stop-gradient.

Require: Encoders ϕ_s, ϕ_a, m; critic Q_θ; actor π_ξ; target copies $\bar{\phi}_s$, $\bar{\phi}_a$, $\bar{m}$, $\bar{Q}$
1: Sample mini-batch $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$; draw $K{-}1$ negatives per transition
2: **Representation:** $\psi_i \leftarrow \phi_s(s_i) \odot \phi_a(a_i)$ for all i
3: Update (ϕ_s, ϕ_a, m) by minimizing ℓ_{FaStR} (Eq. 7)
4: **Critic:** $\bar{\psi}'_i \leftarrow \mathrm{sg}(\bar{\phi}_s(s'_i)) \odot \mathrm{sg}(\bar{\phi}_a(\pi_\xi(s'_i)))$; $y_i \leftarrow r_i + \gamma\, \bar{Q}(\bar{\psi}'_i)$
5: Update θ by minimizing $\frac{1}{N}\sum_i (Q_\theta(\psi_i) - y_i)^2$
6: **Actor:** Update ξ by maximizing $\frac{1}{N}\sum_i Q_\theta\big(\mathrm{sg}(\phi_s(s_i)) \odot \mathrm{sg}(\phi_a(\pi_\xi(s_i)))\big)$
7: **Targets:** $(\bar{\phi}_s, \bar{\phi}_a, \bar{m}, \bar{Q}) \leftarrow \tau\,(\phi_s, \phi_a, m, Q_\theta) + (1{-}\tau)\,(\bar{\phi}_s, \bar{\phi}_a, \bar{m}, \bar{Q})$

B.3 CTRL-SR

CTRL-SR [Gao et al., 2025] serves as the most controlled baseline: the only architectural difference from FaStR is that the transition-ratio logit uses a single monolithic encoder $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ taking the concatenation $[s; a]$ as input, rather than a CP trilinear score built from separate state and action factors. The encoder φ is a 2-layer ResidualMLP $[\dim(\mathcal{S}) + \dim(\mathcal{A})] \to 512 \to 512 \to d$ with $d{=}512$, LayerNorm, and ELU activations; the kernel encoder m, RFF-MLP critic, deterministic actor $[512, 512, 512]$, reward head, optimizer (Adam), learning rates ($3{\times}10^{-4}$ for actor/critic and $1{\times}10^{-4}$ for representation), Polyak factor $\tau{=}0.005$, target/exploration action noise, batch size $N{=}512$, RP-NCE noise levels $M{=}25$, ranking objective, and update cadence are all identical to those of FaStR. We use the authors' open-source implementation without modification.

B.4 Diff-SR

Diff-SR [Shribak et al., 2024] is the second spectral RL baseline. It uses the same TD3 actor and critic backbone as CTRL-SR and FaStR, but replaces the ranking-based perturbed NCE objective with a denoising-diffusion next-state model: the joint encoder $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and the kernel encoder $\mu : \mathcal{S} \times [1, T] \to \mathbb{R}^{d \times \dim(\mathcal{S})}$ are trained by score matching against the variance-preserving (VP) noise schedule of Shribak et al. [2024], with the predicted noise factorized as $\hat{\epsilon}(s, a, s'_t, t) = \phi(s, a)^\top \mu(s'_t, t)$. The state and action inputs to ϕ are first projected by 2-layer Mish MLPs of widths $[256, 128]$ to a shared embedding dimension of 128 each, concatenated to a 256-dim vector, and passed to a 3-layer ResidualMLP $256 \to 512 \to 512 \to 512 \to d$ with $d{=}512$. The kernel encoder

μ takes (s'_t, t) as input, where t is encoded by a sinusoidal positional embedding of width 128 followed by a 2-layer Mish MLP, and outputs a $d \times \dim(\mathcal{S})$ matrix via a 3-layer ResidualMLP $[\dim(\mathcal{S}) + 128] \to 512 \to 512 \to 512 \to d \cdot \dim(\mathcal{S})$. The diffusion uses T=50 sample steps with the noisy state clamped to $[-10, 10]$. The critic is the same RFF-MLP twin Q as CTRL-SR (RFF projection of width 1024 followed by a 1-hidden-layer $[512]$ head); the deterministic actor is a $[512, 512, 512]$ MLP with LayerNorm and a final $\tanh$; the reward head is an RFF-MLP with the same shape as the critic head.

All four networks are trained with Adam. Critic and actor learning rates are 3×10^{-4}; the representation learning rate (jointly applied to ϕ, μ, and the reward head) is 1×10^{-4}. Polyak averaging uses τ=0.005, and actor and critic updates are performed every environment step (`actor_update_freq`=1, `feature_update_ratio`=1). Mini-batch size is N=512, matching the spectral baselines. Exploration uses Gaussian noise $\mathcal{N}(0, \sigma{=}0.2)$ clipped to $[-1, 1]^{\dim(\mathcal{A})}$; target action smoothing uses $\mathcal{N}(0, \tilde{\sigma}{=}0.2)$ clipped to $[-0.3, 0.3]$. The combined training loss is $\lambda_{\text{diff}}\, \ell_{\text{diffusion}} + \lambda_r\, \ell_{\text{reward}} + \lambda_Q\, \ell_{\text{critic}}$ with λ_{diff}=1.0, λ_r=0.1, and λ_Q=1.0; gradients of the critic loss are not back-propagated through the representation (`back_critic_grad`=false). Observation normalization is enabled, matching CTRL-SR and FaStR. We use the open-source implementation accompanying Shribak et al. [2024] without modification.

B.5 SAC

SAC [Haarnoja et al., 2018] uses a stochastic tanh-Gaussian policy and a twin Q critic with ELU activations and LayerNorm. The critic takes the raw (s, a) concatenation as input; no representation learning is performed. We report two width-depth variants. SAC(3-layer) uses hidden sizes $[512, 512, 512]$ for both the actor and the critic, aligning its hidden width with the spectral methods and removing hidden width as an avoidable source of variation. SAC(2-layer) uses hidden sizes $[1024, 1024]$ for both. All other settings are shared between the two variants. Following Haarnoja et al. [2018], the entropy coefficient α is learned online (automatic entropy tuning) toward a target entropy of $-\dim(\mathcal{A})$, optimized via Adam and learning rate 3×10^{-4} from an initial value α_0=0.2. Actor, critic, and entropy coefficient are all trained with Adam at learning rate 3×10^{-4}; soft target updates use Polyak factor τ=0.005 at every environment step, and the actor is updated every step (`actor_update_freq`=1). Mini-batch size is 512 for parity with the spectral methods. There is no exploration noise injected on top of the policy stochasticity, no observation normalization, and no gradient clipping.

B.6 TD7

TD7 [Fujimoto et al., 2023] extends TD3 with a learned state-action encoder (zs, zsa) trained by latent dynamics regression, Loss-Adjusted Prioritized (LAP) replay with per-transition priorities, behavior-cloning regularization at the policy update, and policy checkpointing. We use the official implementation released by the authors with the original DM Control Suite hyperparameters; the only configuration we set is the task identifier and the seed. For completeness: encoder, actor, and critic each use 2-hidden-layer MLPs of width 256 with ELU activations (ReLU for the actor) and the AvgL1Norm normalization of Fujimoto et al. [2023] (per-layer activation rescaling); the encoded state-action dimension is z_{dim}=256. Optimizer is Adam with learning rate 3×10^{-4} for all three networks. Replay uses LAP with α=0.4, min-priority 1, and Huber critic loss; the discount is γ=0.99 and the target network is updated by a hard copy every 250 updates (rather than Polyak averaging). Exploration uses Gaussian noise $\mathcal{N}(0, 0.1)$ clipped to $[-1, 1]$; target policy smoothing uses $\mathcal{N}(0, 0.2)$ clipped to $[-0.5, 0.5]$; the actor is updated every 2 critic updates with behavior-cloning weight λ_{BC}=0.1. Policy checkpointing uses up to 20 episodes per checkpoint and starts after 7.5×10^5 agent steps with weight-reset coefficient 0.9. Mini-batch size is 256 (the value used in the original TD7 paper); we did not tune this for parity with the spectral methods. Random exploration runs for the first $25{,}000$ agent steps before any training begins.

B.7 TD-MPC2

TD-MPC2 [Hansen et al., 2024] is the model-based baseline. We use the official default 5M-parameter single-task configuration of Hansen et al. [2024] without modification, integrated into the shared training and evaluation pipeline of Section B.1 and trained for the same 1×10^6 environment-interaction budget, with results reported over 8 seeds. DreamerV3 [Hafner et al., 2023] under its

standard configuration requires substantially more compute than our available budget, and running it at a much smaller budget would not constitute a fair comparison, so it is not included.

B.8 Hyperparameter Summary

Table 2 consolidates the values reported above. Dashes (—) indicate that a hyperparameter does not apply to a given method (e.g., a representation learning rate for SAC). Where a method has additional method-specific hyperparameters not listed in the table (LAP α for TD7, BC weight, target entropy for SAC, RP-NCE noise schedule for the spectral methods), we have specified them in the corresponding paragraph above and use the original authors' default values.

Table 2: Hyperparameters for FaStR and the SAC, TD7, and spectral baselines used in Table 1; TD-MPC2 uses its official default single-task configuration (Section B.7). Values shared across all methods (discount γ=0.99, 1×10^6 environment frames, replay capacity 1×10^6, action repeat 2, 8 seeds, 10 evaluation episodes every 10,000 frames) are described in Section B.1.

Parameter	FaStR (Ours)	CTRL-SR	Diff-SR	SAC(3-layer)	SAC(2-layer)	TD7
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Actor learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Critic learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Representation/encoder LR	1×10^{-4}	1×10^{-4}	1×10^{-4}	—	—	3×10^{-4}
Mini-batch size	512	512	512	512	512	256
Target update	Polyak τ=0.005	Polyak τ=0.005	Polyak τ=0.005	Polyak τ=0.005	Polyak τ=0.005	Hard, every 250 updates
Actor update frequency	1	1	1	1	1	2
Actor hidden sizes	[512,512,512]	[512,512,512]	[512,512,512]	[512,512,512]	[1024,1024]	[256,256]
Critic hidden sizes	[512] (after RFF 1024)	[512] (after RFF 1024)	[512] (after RFF 1024)	[512,512,512]	[1024,1024]	[256,256]
Activation	ELU/Mish + LN	ELU/Mish + LN	Mish + LN	ELU + LN	ELU + LN	ELU + AvgL1Norm
Number of critics	2 (twin Q)	2 (twin Q)	2 (twin Q)	2 (twin Q)	2 (twin Q)	2 (twin Q)
Exploration noise σ	0.2	0.2	0.2	policy entropy	policy entropy	0.1
Target policy noise $\tilde{\sigma}$	0.2 (clip $\pm$0.3)	0.2 (clip $\pm$0.3)	0.2 (clip $\pm$0.3)	—	—	0.2 (clip $\pm$0.5)
Feature dim d	512	512	512	—	—	256 (z_{dim})
Encoder hidden sizes	[512,512]	[512,512]	[512,512,512]	—	—	[256,256]
RFF projection width	1024	1024	1024	—	—	—
Representation objective	RP-NCE (ranking)	RP-NCE (ranking)	VP diffusion (T=50)	—	—	latent dynamics regression
RP-NCE noise levels M	25	25	—	—	—	—
Negative samples K	512 (in-batch)	512 (in-batch)	—	—	—	—
Replay sampling	uniform	uniform	uniform	uniform	uniform	LAP (α=0.4)
Observation normalization	running stats	running stats	running stats	none	none	none
Random/warmup frames	10,000	10,000	10,000	10,000	10,000	50,000 (25k agent steps)

C Proof Details

C.1 Notation, Conventions, and Standing Assumptions

We work with a discounted MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ on a continuous state space $\mathcal{S}$ and continuous action space $\mathcal{A}$, with $\gamma \in [0, 1)$ and reward bounded by $R_{\max}$. We assume the existence of a fixed reference *probability measure* ν on $\mathcal{S}$ that dominates every transition kernel, so that $P(\cdot \mid s, a) \ll \nu$ and admits a density $P(s' \mid s, a)$ with respect to ν. All integrals over $\mathcal{S}$ are then taken with respect to ν, i.e. $\int_{\mathcal{S}} f(s')\, ds' \equiv \int_{\mathcal{S}} f(s')\, d\nu(s')$. This is the standard setting for $L^2(\nu)$-based representation-learning analyses and fixes the measure with respect to which all densities and risks are evaluated.

Symbols.

- $\phi_s : \mathcal{S} \to \mathbb{R}^d$, $\phi_a : \mathcal{A} \to \mathbb{R}^d$, $m : \mathcal{S} \to \mathbb{R}^d$: state encoder, action encoder, kernel encoder.
- $\psi(s, a) := \phi_s(s) \odot \phi_a(a) \in \mathbb{R}^d$: *factored* (Hadamard) feature.
- $\varphi(s, a) \in \mathbb{R}^d$, $\mu : \mathcal{S} \to \mathbb{R}^d$: *joint* (monolithic) encoder and kernel encoder.
- $\mathcal{F}_s, \mathcal{F}_a, \mathcal{F}_m, \mathcal{F}_{sa}$: hypothesis classes for $\phi_s, \phi_a, m, \varphi$.
- $\mathcal{G}_{\text{fac}} := \{(s, a, s') \mapsto (\phi_s(s) \odot \phi_a(a))^\top m(s') : \phi_s \in \mathcal{F}_s, \phi_a \in \mathcal{F}_a, m \in \mathcal{F}_m\}$.
- $\mathcal{G}_{\text{joint}} := \{(s, a, s') \mapsto \varphi(s, a)^\top \mu(s') : \varphi \in \mathcal{F}_{sa}, \mu \in \mathcal{F}_m\}$.

Boundedness assumptions. We assume there exist constants $B_\phi, B_\varphi, B_{m,\infty}, B_{m,1}, B_g \geq 1$ such that, uniformly over their hypothesis classes,

$$\sup_{\phi_s \in \mathcal{F}_s} \sup_{s \in \mathcal{S}} \|\phi_s(s)\|_2, \ \sup_{\phi_a \in \mathcal{F}_a} \sup_{a \in \mathcal{A}} \|\phi_a(a)\|_2 \leq B_\phi, \qquad \sup_{\varphi \in \mathcal{F}_{sa}} \sup_{(s,a)} \|\varphi(s,a)\|_2 \leq B_\varphi,$$

$$\sup_{h \in \mathcal{F}_m} \sup_{s' \in \mathcal{S}} \|h(s')\|_2 \leq B_{m,\infty}, \qquad \sup_{h \in \mathcal{F}_m} \int_{\mathcal{S}} \|h(s')\|_2 \, d\nu(s') \leq B_{m,1},$$

and $|g(s,a,s')| \leq B_g$ for every $g \in \mathcal{G}_{\text{fac}} \cup \mathcal{G}_{\text{joint}}$. Thus both the factored kernel encoder m and the joint kernel encoder μ obey the same pointwise and $L^1(\nu)$ bounds. For a single continuous encoder on compact $\mathcal{S}$ these bounds are automatic; for a hypothesis class we impose them uniformly. We separate the pointwise bound $B_{m,\infty}$ (used for covering numbers and sup-norm Lipschitz arguments) from the integral bound $B_{m,1}$ (used to verify that integrals against V^π are finite); the two regimes appear in different proofs.

Rademacher complexity and covering numbers. For a fixed sample $S = (z_1, \ldots, z_n)$ and a scalar-valued class $\mathcal{H}$, define the empirical absolute Rademacher complexity

$$\widehat{\mathcal{R}}_S(\mathcal{H}) := \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n \sigma_i \, h(z_i) \right| \right], \qquad \sigma_i \overset{\text{iid}}{\sim} \text{Unif}\{\pm 1\}.$$

The expected Rademacher complexity is

$$\mathcal{R}_n(\mathcal{H}) := \mathbb{E}_S \left[\widehat{\mathcal{R}}_S(\mathcal{H}) \right].$$

All finite-sample bounds use this expected version. Set-inclusion arguments hold pointwise for $\widehat{\mathcal{R}}_S$ and therefore also after taking expectation over S. For scalar-valued classes, $\mathcal{N}(\mathcal{H}, \epsilon)$ denotes the covering number under $d_\infty(h, h') = \sup_z |h(z) - h'(z)|$. For vector-valued encoder classes such as $\mathcal{F}_s, \mathcal{F}_a, \mathcal{F}_m, \mathcal{F}_{sa}$, the covering metric is

$$d_\infty(f, f') = \sup_x \|f(x) - f'(x)\|_2.$$

Best-in-class CP error. For non-realizable settings we write

$$\epsilon_{\text{CP}}^2 \ := \ \inf_{g \in \mathcal{G}_{\text{fac}}} \ \mathbb{E}_{(s,a) \sim \rho} \left[\int_{\mathcal{S}} \big(P(s' \mid s,a) - g(s,a,s') \big)^2 \, d\nu(s') \right].$$

Under CP realizability (Assumption 1), $\epsilon_{\text{CP}} = 0$.

Proof architecture. The results in this appendix support a single claim: the CP factorization of the transition kernel gives a representation whose statistical complexity decomposes additively in (d_s, d_a), whereas the generic uniform-convergence bound for an unrestricted joint encoder scales with the full joint dimension $d_s + d_a$. Appendix C.2 derives the Q-linearity stated in Section 3.1: under CP factorization, every Q^π is linear in the Hadamard feature, so the factored class is matched to the structure of the transition kernel rather than chosen as an architectural convenience. Appendix C.3 proves Theorem 3.1, the main technical step: the Hadamard product converts the joint covering of $\mathcal{G}_{\text{fac}}$ into a sum of three independent component coverings, and the same set-inclusion argument gives a Rademacher ordering between the two classes. Appendix C.5 states this Rademacher monotonicity on its own, since it follows from set inclusion without needing the covering bound. Appendix C.6 converts the Rademacher bound into an excess-risk bound for the spectral L_2 objective with constants made explicit (Proposition 3.2). Appendix C.7 converts the additive vs. multiplicative entropy gap into the sufficient-sample-size gap implied by these bounds (Proposition 3.3). It ends with a same-n comparison that holds without any Lipschitz instantiation: under the containment condition, both Rademacher terms in Proposition 3.2 are no larger for the factored class, so the certified upper bound on representation error is no larger at every sample size (Remark C.7).

C.2 Derivation of the Q-linearity under CP

The CP transition structure makes every Q^π linear in the Hadamard feature ψ. This appendix gives the linear-MDP derivation specialized to the CP feature, together with norm bounds on the value coefficients used downstream. We spell out the interchange of summation and integration and the norm estimates in detail.

Setup. Suppose Assumption 1 holds. That is, the reward decomposes as

$$r(s,a) = \psi(s,a)^\top \theta_r,$$

where

$$\psi(s,a) = \phi_s(s) \odot \phi_a(a) \in \mathbb{R}^d, \qquad \psi_k(s,a) = \phi_{s,k}(s)\phi_{a,k}(a),$$

and the transition density factorizes as

$$P(s' \mid s,a) = \sum_{k=1}^{d} \phi_{s,k}(s)\, \phi_{a,k}(a)\, m_k(s') = \sum_{k=1}^{d} \psi_k(s,a) m_k(s'). \tag{9}$$

Here each m_k is measurable and belongs to $L^1(\nu)$. We also use the boundedness conventions of Appendix C.1; in particular, writing

$$m(s') = (m_1(s'), \ldots, m_d(s'))^\top,$$

we have

$$\int_{\mathcal{S}} \|m(s')\|_2 \, d\nu(s') \le B_{m,1}.$$

Consequently, for every coordinate k,

$$\int_{\mathcal{S}} |m_k(s')| \, d\nu(s') \le \int_{\mathcal{S}} \|m(s')\|_2 \, d\nu(s') \le B_{m,1}.$$

Claim. For every stationary policy π, there exists $w^\pi \in \mathbb{R}^d$ such that

$$Q^\pi(s,a) = \psi(s,a)^\top w^\pi,$$

where

$$w_k^\pi = \theta_{r,k} + \gamma \theta_k^\pi, \qquad \theta_k^\pi := \int_{\mathcal{S}} m_k(s') V^\pi(s') \, d\nu(s').$$

Moreover,

$$\|w^\pi\|_2 \le \|\theta_r\|_2 + \gamma \, \frac{B_{m,1} R_{\max}}{1-\gamma}.$$

Proof. Fix an arbitrary stationary policy π. We prove the result for this fixed π.

First, by the definition of the discounted value function,

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s \right].$$

Since $|r(s,a)| \le R_{\max}$ for all (s,a) and $0 \le \gamma < 1$, we have, for every s,

$$\begin{aligned}
|V^\pi(s)| &= \left| \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s \right] \right| \\
&\le \mathbb{E}_\pi \left[\left| \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right| \,\middle|\, s_0 = s \right] \\
&\le \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t |r(s_t, a_t)| \,\middle|\, s_0 = s \right] \\
&\le \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{\max} \,\middle|\, s_0 = s \right] \\
&= R_{\max} \sum_{t=0}^{\infty} \gamma^t = \frac{R_{\max}}{1-\gamma}.
\end{aligned}$$

Therefore,

$$\|V^\pi\|_\infty \le \frac{R_{\max}}{1-\gamma}. \tag{10}$$

The Bellman equation for Q^π is

$$Q^\pi(s,a) = r(s,a) + \gamma \int_{\mathcal{S}} P(s' \mid s,a) V^\pi(s')\, d\nu(s'). \tag{11}$$

We now expand the transition term in (11). Substituting the CP density representation (9) gives

$$\begin{aligned}\int_{\mathcal{S}} P(s' \mid s,a) V^\pi(s')\, d\nu(s') &= \int_{\mathcal{S}} \left(\sum_{k=1}^{d} \phi_{s,k}(s)\phi_{a,k}(a) m_k(s') \right) V^\pi(s')\, d\nu(s') \\ &= \int_{\mathcal{S}} \left(\sum_{k=1}^{d} \psi_k(s,a) m_k(s') \right) V^\pi(s')\, d\nu(s'). \end{aligned} \tag{12}$$

Before moving the finite sum outside the integral, we check that each term is integrable. For a fixed coordinate k, using (10),

$$\begin{aligned}\int_{\mathcal{S}} |m_k(s') V^\pi(s')|\, d\nu(s') &\le \int_{\mathcal{S}} |m_k(s')|\, |V^\pi(s')|\, d\nu(s') \\ &\le \frac{R_{\max}}{1-\gamma} \int_{\mathcal{S}} |m_k(s')|\, d\nu(s') \\ &\le \frac{R_{\max}}{1-\gamma} B_{m,1} < \infty. \end{aligned}$$

Thus $m_k V^\pi$ is integrable for every k, and the quantity

$$\theta_k^\pi := \int_{\mathcal{S}} m_k(s') V^\pi(s')\, d\nu(s')$$

is well-defined.

Since the sum over k contains only finitely many terms, exchanging the sum and the integral uses only the elementary linearity of the integral. Therefore,

$$\begin{aligned}\int_{\mathcal{S}} \left(\sum_{k=1}^{d} \psi_k(s,a) m_k(s') \right) V^\pi(s')\, d\nu(s') &= \int_{\mathcal{S}} \sum_{k=1}^{d} \psi_k(s,a) m_k(s') V^\pi(s')\, d\nu(s') \\ &= \sum_{k=1}^{d} \int_{\mathcal{S}} \psi_k(s,a) m_k(s') V^\pi(s')\, d\nu(s') \\ &= \sum_{k=1}^{d} \psi_k(s,a) \int_{\mathcal{S}} m_k(s') V^\pi(s')\, d\nu(s') \\ &= \sum_{k=1}^{d} \psi_k(s,a) \theta_k^\pi. \end{aligned} \tag{13}$$

In the third line above, $\psi_k(s,a)$ is taken outside the integral because, for fixed (s,a), it is a scalar that does not depend on s'.

Next, we expand the reward term. Since $r(s,a) = \psi(s,a)^\top \theta_r$, we have

$$r(s,a) = \sum_{k=1}^{d} \psi_k(s,a)\theta_{r,k} = \sum_{k=1}^{d} \phi_{s,k}(s)\phi_{a,k}(a)\theta_{r,k}. \tag{14}$$

Combining (11), (13), and (14), we obtain

$$\begin{aligned} Q^\pi(s,a) &= r(s,a) + \gamma \int_{\mathcal{S}} P(s' \mid s,a) V^\pi(s')\, d\nu(s') \\ &= \sum_{k=1}^{d} \psi_k(s,a)\theta_{r,k} + \gamma \sum_{k=1}^{d} \psi_k(s,a)\theta_k^\pi \\ &= \sum_{k=1}^{d} \psi_k(s,a)\theta_{r,k} + \sum_{k=1}^{d} \psi_k(s,a)\gamma\theta_k^\pi \\ &= \sum_{k=1}^{d} \psi_k(s,a)\left(\theta_{r,k} + \gamma\theta_k^\pi\right). \end{aligned}$$

Define $w^\pi \in \mathbb{R}^d$ coordinatewise by

$$w_k^\pi := \theta_{r,k} + \gamma\theta_k^\pi, \qquad k = 1, \ldots, d.$$

Then the previous display becomes

$$Q^\pi(s,a) = \sum_{k=1}^{d} \psi_k(s,a) w_k^\pi = \psi(s,a)^\top w^\pi.$$

This proves the claimed linear representation of Q^π.

It remains to prove the stated norm bound. By definition,

$$\theta^\pi = \left(\int_{\mathcal{S}} m_1(s') V^\pi(s')\, d\nu(s'), \ldots, \int_{\mathcal{S}} m_d(s') V^\pi(s')\, d\nu(s') \right)^\top.$$

Equivalently, in vector notation,

$$\theta^\pi = \int_{\mathcal{S}} m(s') V^\pi(s')\, d\nu(s'),$$

where the vector integral is understood coordinatewise.

We show that

$$\|\theta^\pi\|_2 \le \frac{B_{m,1} R_{\max}}{1-\gamma}.$$

If $\theta^\pi = 0$, this inequality is immediate. Otherwise, let

$$u = \frac{\theta^\pi}{\|\theta^\pi\|_2}.$$

Then $\|u\|_2 = 1$, and hence

$$\begin{aligned} \|\theta^\pi\|_2 &= u^\top \theta^\pi \\ &= u^\top \int_{\mathcal{S}} m(s') V^\pi(s')\, d\nu(s') \\ &= \int_{\mathcal{S}} u^\top m(s') V^\pi(s')\, d\nu(s') \\ &\le \int_{\mathcal{S}} \left| u^\top m(s') V^\pi(s') \right| d\nu(s') \\ &= \int_{\mathcal{S}} \left| u^\top m(s') \right| |V^\pi(s')|\, d\nu(s') \\ &\le \int_{\mathcal{S}} \|u\|_2 \|m(s')\|_2 |V^\pi(s')|\, d\nu(s') \\ &= \int_{\mathcal{S}} \|m(s')\|_2 |V^\pi(s')|\, d\nu(s') \\ &\le \frac{R_{\max}}{1-\gamma} \int_{\mathcal{S}} \|m(s')\|_2\, d\nu(s') \\ &\le \frac{B_{m,1} R_{\max}}{1-\gamma}. \end{aligned}$$

The inequality

$$|u^\top m(s')| \le \|u\|_2 \|m(s')\|_2$$

is the Cauchy–Schwarz inequality. The key point is that this argument bounds the Euclidean norm of the whole vector θ^π directly, and therefore does not introduce an unnecessary $\sqrt{d}$ factor.

Finally, since

$$w^\pi = \theta_r + \gamma \theta^\pi,$$

the triangle inequality gives

$$\begin{aligned} \|w^\pi\|_2 &= \|\theta_r + \gamma\theta^\pi\|_2 \\ &\le \|\theta_r\|_2 + \|\gamma\theta^\pi\|_2 \\ &= \|\theta_r\|_2 + \gamma\|\theta^\pi\|_2 \\ &\le \|\theta_r\|_2 + \gamma \, \frac{B_{m,1} R_{\max}}{1-\gamma}. \end{aligned}$$

This proves the claimed norm bound and completes the proof. □

Remark (Linear-MDP structure and feature bound) Defining signed measures $\mu_k(A) := \int_A m_k(s')\, d\nu(s')$, the CP transition density is equivalent to $P(\cdot \mid s,a) = \sum_{k=1}^d \psi_k(s,a)\mu_k(\cdot)$. Together with $r(s,a) = \psi(s,a)^\top \theta_r$, this matches the linear-MDP definition of Jin et al. [2020, Definition 2] with feature map ψ, used in Appendix C.7. The same swap-of-sum-and-integral argument as the proof of Q-linearity above gives, for any bounded measurable $f : \mathcal{S} \to \mathbb{R}$, $\int_{\mathcal{S}} P(s' \mid s,a) f(s')\, d\nu(s') = \psi(s,a)^\top \theta_f$ with $(\theta_f)_k := \int_{\mathcal{S}} m_k(s') f(s')\, d\nu(s')$ and $\|\theta_f\|_2 \le B_{m,1}\|f\|_\infty$. Finally, the Hadamard-product inequality $\|u \odot v\|_2 \le \|u\|_2 \|v\|_2$ applied to $\psi = \phi_s \odot \phi_a$ gives the uniform feature bound $\|\psi(s,a)\|_2 \le B_\phi^2$.

C.3 Proof of Theorem 3.1: Additive Covering Number Decomposition

This is the main technical step. The Hadamard product structure converts the joint covering of $\mathcal{G}_{\text{fac}}$ into a product of three independent coverings on the state, action, and next-state components. After taking logarithms, this product becomes an additive decomposition of metric entropies. Without using this decomposition, a generic uniform-convergence treatment of the state-action pair would cover the whole state-action feature class directly and would therefore give the same $(d_s + d_a)$-dimensional entropy scaling as the joint class. The Hadamard structure reduces the factored upper bound to the largest of the state and action dimensions.

Throughout this appendix, covering numbers for the component classes are taken with respect to the uniform sup-ℓ_2 metrics

$$d_s(\phi_s, \widetilde{\phi_s}) := \sup_{s \in \mathcal{S}} \|\phi_s(s) - \widetilde{\phi_s}(s)\|_2,$$

$$d_a(\phi_a, \widetilde{\phi_a}) := \sup_{a \in \mathcal{A}} \|\phi_a(a) - \widetilde{\phi_a}(a)\|_2,$$

and

$$d_m(m, \widetilde{m}) := \sup_{s' \in \mathcal{S}} \|m(s') - \widetilde{m}(s')\|_2.$$

For scoring functions $g : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, the covering number is taken with respect to the usual uniform norm

$$\|g - \widetilde{g}\|_\infty := \sup_{(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}} |g(s,a,s') - \widetilde{g}(s,a,s')|.$$

Theorem (Restatement of Theorem 3.1)**.**

(i) Additive decomposition. *For every* $\epsilon > 0$*,*

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \le \log \mathcal{N}\Big(\mathcal{F}_s, \tfrac{\epsilon}{3B_{m,\infty}B_\phi}\Big) + \log \mathcal{N}\Big(\mathcal{F}_a, \tfrac{\epsilon}{3B_{m,\infty}B_\phi}\Big) + \log \mathcal{N}\Big(\mathcal{F}_m, \tfrac{\epsilon}{3B_\phi^2}\Big). \tag{15}$$

(ii) Containment. *If*

$$\mathcal{F}_{sa} \supseteq \{\phi_s \odot \phi_a : \phi_s \in \mathcal{F}_s, \phi_a \in \mathcal{F}_a\},$$

then $\mathcal{G}_{\mathrm{fac}} \subseteq \mathcal{G}_{\mathrm{joint}}$*, hence*

$$\mathcal{N}(\mathcal{G}_{\mathrm{fac}}, \epsilon) \leq \mathcal{N}(\mathcal{G}_{\mathrm{joint}}, \epsilon)$$

for every $\epsilon > 0$.

Lemma C.1 (Perturbation bound for factored CP scores)**.** *Let*

$$g(s, a, s') = (\phi_s(s) \odot \phi_a(a))^\top m(s')$$

and

$$\widetilde{g}(s, a, s') = (\widetilde{\phi_s}(s) \odot \widetilde{\phi_a}(a))^\top \widetilde{m}(s').$$

Assume

$$\sup_s \|\phi_s(s)\|_2,\ \sup_s \|\widetilde{\phi_s}(s)\|_2 \leq B_\phi, \qquad \sup_a \|\phi_a(a)\|_2,\ \sup_a \|\widetilde{\phi_a}(a)\|_2 \leq B_\phi,$$

and

$$\sup_{s'} \|\widetilde{m}(s')\|_2 \leq B_{m,\infty}.$$

Define

$$\Delta_s := \sup_s \|\phi_s(s) - \widetilde{\phi_s}(s)\|_2, \qquad \Delta_a := \sup_a \|\phi_a(a) - \widetilde{\phi_a}(a)\|_2,$$

and

$$\Delta_m := \sup_{s'} \|m(s') - \widetilde{m}(s')\|_2.$$

Then

$$\|g - \widetilde{g}\|_\infty \leq B_\phi^2 \Delta_m + B_{m,\infty} B_\phi \Delta_s + B_{m,\infty} B_\phi \Delta_a. \tag{16}$$

Proof. Write

$$x(s, a) := \phi_s(s) \odot \phi_a(a), \qquad \widetilde{x}(s, a) := \widetilde{\phi_s}(s) \odot \widetilde{\phi_a}(a).$$

For fixed (s, a, s'),

$$g(s, a, s') - \widetilde{g}(s, a, s') = x(s, a)^\top \big(m(s') - \widetilde{m}(s')\big) + \big(x(s, a) - \widetilde{x}(s, a)\big)^\top \widetilde{m}(s').$$

By Cauchy–Schwarz,

$$|g(s, a, s') - \widetilde{g}(s, a, s')| \leq \|x(s, a)\|_2 \|m(s') - \widetilde{m}(s')\|_2 + \|x(s, a) - \widetilde{x}(s, a)\|_2 \|\widetilde{m}(s')\|_2.$$

The Hadamard product bound gives

$$\|x(s, a)\|_2 = \|\phi_s(s) \odot \phi_a(a)\|_2 \leq \|\phi_s(s)\|_2 \|\phi_a(a)\|_2 \leq B_\phi^2.$$

Moreover,

$$x(s, a) - \widetilde{x}(s, a) = (\phi_s(s) - \widetilde{\phi_s}(s)) \odot \phi_a(a) + \widetilde{\phi_s}(s) \odot (\phi_a(a) - \widetilde{\phi_a}(a)).$$

Using

$$\|u \odot v\|_2 \leq \|v\|_\infty \|u\|_2 \leq \|v\|_2 \|u\|_2,$$

we obtain

$$\|x(s, a) - \widetilde{x}(s, a)\|_2 \leq B_\phi \|\phi_s(s) - \widetilde{\phi_s}(s)\|_2 + B_\phi \|\phi_a(a) - \widetilde{\phi_a}(a)\|_2.$$

Taking the supremum over (s, a, s') proves (16). □

Proof. We prove the two statements separately.

Part (i): additive decomposition. Choose component covers $\widehat{\mathcal{F}}_s$, $\widehat{\mathcal{F}}_a$, $\widehat{\mathcal{F}}_m$ of $\mathcal{F}_s, \mathcal{F}_a, \mathcal{F}_m$ at radii

$$\epsilon_s := \frac{\epsilon}{3B_{m,\infty}B_\phi}, \qquad \epsilon_a := \frac{\epsilon}{3B_{m,\infty}B_\phi}, \qquad \epsilon_m := \frac{\epsilon}{3B_\phi^2},$$

respectively, under the corresponding uniform sup-ℓ_2 metrics. Take any

$$g(s,a,s') = (\phi_s(s) \odot \phi_a(a))^\top m(s') \in \mathcal{G}_{\text{fac}}.$$

By the definitions of the three covers, choose

$$\widehat{\phi_s} \in \widehat{\mathcal{F}}_s, \qquad \widehat{\phi_a} \in \widehat{\mathcal{F}}_a, \qquad \widehat{m} \in \widehat{\mathcal{F}}_m$$

such that

$$\sup_s \|\phi_s(s) - \widehat{\phi_s}(s)\|_2 \le \epsilon_s, \qquad \sup_a \|\phi_a(a) - \widehat{\phi_a}(a)\|_2 \le \epsilon_a, \qquad \sup_{s'} \|m(s') - \widehat{m}(s')\|_2 \le \epsilon_m.$$

Define

$$\widehat{g}(s,a,s') = (\widehat{\phi_s}(s) \odot \widehat{\phi_a}(a))^\top \widehat{m}(s').$$

Applying Lemma C.1 gives

$$\begin{aligned}\|g - \widehat{g}\|_\infty &\le B_\phi^2 \epsilon_m + B_{m,\infty} B_\phi \epsilon_s + B_{m,\infty} B_\phi \epsilon_a \\ &= \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} = \epsilon.\end{aligned}$$

Thus the product class

$$\widehat{\mathcal{G}}_{\text{fac}} := \left\{(s,a,s') \mapsto (\widehat{\phi_s}(s) \odot \widehat{\phi_a}(a))^\top \widehat{m}(s') : \widehat{\phi_s} \in \widehat{\mathcal{F}}_s,\ \widehat{\phi_a} \in \widehat{\mathcal{F}}_a,\ \widehat{m} \in \widehat{\mathcal{F}}_m\right\}$$

is an ϵ-cover of $\mathcal{G}_{\text{fac}}$. Each element of $\widehat{\mathcal{G}}_{\text{fac}}$ is determined by a triple from the three component covers, so

$$|\widehat{\mathcal{G}}_{\text{fac}}| \le |\widehat{\mathcal{F}}_s|\,|\widehat{\mathcal{F}}_a|\,|\widehat{\mathcal{F}}_m|.$$

Taking the smallest possible component covers, or equivalently taking the infimum over such covers, yields

$$\mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \le \mathcal{N}(\mathcal{F}_s, \epsilon_s)\mathcal{N}(\mathcal{F}_a, \epsilon_a)\mathcal{N}(\mathcal{F}_m, \epsilon_m).$$

Taking logarithms and substituting the definitions of $\epsilon_s, \epsilon_a, \epsilon_m$ gives (15).

Part (ii): containment. Take any $g \in \mathcal{G}_{\text{fac}}$. Then, for some $\phi_s \in \mathcal{F}_s$, $\phi_a \in \mathcal{F}_a$, and $m \in \mathcal{F}_m$,

$$g(s,a,s') = (\phi_s(s) \odot \phi_a(a))^\top m(s').$$

By the assumption on $\mathcal{F}_{sa}$,

$$\varphi(s,a) := \phi_s(s) \odot \phi_a(a)$$

belongs to $\mathcal{F}_{sa}$. Taking $\mu := m$, we get

$$g(s,a,s') = \varphi(s,a)^\top \mu(s'),$$

so $g \in \mathcal{G}_{\text{joint}}$. Hence

$$\mathcal{G}_{\text{fac}} \subseteq \mathcal{G}_{\text{joint}}.$$

The covering-number inequality follows because any ϵ-cover of a set also covers all of its subsets. □

C.4 Lipschitz Instantiation: an Upper-Bound vs. Upper-Bound Comparison

We now explain how the additive decomposition in (15) translates into the stated entropy scaling under standard Lipschitz covering bounds. Assume that

$$\mathcal{S} \subseteq \mathbb{R}^{d_s}, \qquad \mathcal{A} \subseteq \mathbb{R}^{d_a}$$

are compact, and that $\mathcal{F}_s, \mathcal{F}_a, \mathcal{F}_m$ are uniformly bounded Lipschitz classes. For L-Lipschitz maps from a compact set $X \subseteq \mathbb{R}^p$ into a bounded subset of $\mathbb{R}^d$, the standard entropy bound gives

$$\log \mathcal{N}(\mathcal{F}, \epsilon) \le c\, d\big(L \operatorname{diam}(X)/\epsilon\big)^p$$

for the relevant range of ϵ; see Wainwright [2019, Theorem 5.7]. The important point is that the exponent is the dimension p of the input domain. Applying this bound to the three terms in (15) yields finite constants A_s, A_a, A_m such that

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \leq A_s \epsilon^{-d_s} + A_a \epsilon^{-d_a} + A_m \epsilon^{-d_s}.$$

Therefore, for

$$D_{\max} := \max(d_s, d_a)$$

and $0 < \epsilon \leq 1$,

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \leq A_{\text{fac}} \epsilon^{-D_{\max}}, \tag{17}$$

where $A_{\text{fac}} < \infty$ depends on the Lipschitz constants, diameters, output dimension, and uniform radius bounds, but not on ϵ. For the joint class, the unrestricted state-action encoder is a Lipschitz class on the product domain

$$\mathcal{S} \times \mathcal{A} \subseteq \mathbb{R}^{d_s + d_a}.$$

The same two-term covering argument for

$$g(s, a, s') = \varphi(s, a)^\top \mu(s')$$

gives

$$\log \mathcal{N}(\mathcal{G}_{\text{joint}}, \epsilon) \leq \log \mathcal{N}\Big(\mathcal{F}_{sa}, \tfrac{\epsilon}{2B_{m,\infty}}\Big) + \log \mathcal{N}\Big(\mathcal{F}_m, \tfrac{\epsilon}{2B_{sa}}\Big),$$

where B_{sa} is a uniform bound on $\|\varphi(s, a)\|_2$. Applying the Lipschitz entropy bound to $\mathcal{F}_{sa}$ on the product domain and to $\mathcal{F}_m$ on $\mathcal{S}$ yields finite constants A_{sa}, A'_m such that

$$\log \mathcal{N}(\mathcal{G}_{\text{joint}}, \epsilon) \leq A_{sa} \epsilon^{-(d_s + d_a)} + A'_m \epsilon^{-d_s}.$$

Since $d_s \leq d_s + d_a$, for $0 < \epsilon \leq 1$ this implies

$$\log \mathcal{N}(\mathcal{G}_{\text{joint}}, \epsilon) \leq A_{\text{joint}} \epsilon^{-(d_s + d_a)} \tag{18}$$

for a finite constant A_{joint} independent of ϵ. Combining (17) and (18), the factored and joint upper bounds scale as

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \epsilon) \lesssim \epsilon^{-\max(d_s, d_a)}, \qquad \log \mathcal{N}(\mathcal{G}_{\text{joint}}, \epsilon) \lesssim \epsilon^{-(d_s + d_a)}.$$

This is an upper-bound comparison: under the same covering template, the factored entropy certificate depends on $\max(d_s, d_a)$ while the joint certificate depends on $d_s + d_a$. We discuss the absence of a matching lower bound in Remark C.7.

C.5 Rademacher Monotonicity Component of Theorem 3.1(ii)

Theorem 3.1(ii) states that the inclusion $\mathcal{G}_{\text{fac}} \subseteq \mathcal{G}_{\text{joint}}$ propagates to both covering numbers and Rademacher complexities at every sample size. The covering-number half follows from monotonicity of $\mathcal{N}(\cdot, \epsilon)$ under set inclusion (Appendix C.3, Part (ii)). The Rademacher half follows from a simpler argument than the covering bound: it requires only set inclusion, not entropy.

Proposition (Rademacher inequality from Theorem 3.1(ii))**.** *Under the containment hypothesis of Theorem 3.1(ii), for every n and every i.i.d. sample $z_1, \ldots, z_n$,*

$$\mathcal{R}_n(\mathcal{G}_{\text{fac}}) \;\leq\; \mathcal{R}_n(\mathcal{G}_{\text{joint}}).$$

Proof. The inclusion $\mathcal{G}_{\text{fac}} \subseteq \mathcal{G}_{\text{joint}}$ was established in Theorem 3.1(ii). For every fixed sample S and every sign realization σ,

$$\sup_{g \in \mathcal{G}_{\text{fac}}} \left| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \, g(z_i) \right| \;\leq\; \sup_{g \in \mathcal{G}_{\text{joint}}} \left| \frac{1}{n} \sum_{i=1}^{n} \sigma_i \, g(z_i) \right|,$$

because the supremum over a subset is no larger than the supremum over the superset. Taking expectation over σ gives $\widehat{\mathcal{R}}_S(\mathcal{G}_{\text{fac}}) \leq \widehat{\mathcal{R}}_S(\mathcal{G}_{\text{joint}})$; taking expectation over S gives the stated $\mathcal{R}_n$ inequality. □

Remark (Why not via Dudley?) An alternative would be to bound both Rademacher complexities via Dudley's entropy integral and observe that $\mathcal{N}(\mathcal{G}_{\mathrm{fac}}, \epsilon) \le \mathcal{N}(\mathcal{G}_{\mathrm{joint}}, \epsilon)$ propagates through the integral. This argument only bounds the *upper bounds* on the two complexities and does not establish the ordering of the complexities themselves: the joint Rademacher complexity could in principle be much smaller than its Dudley upper bound. Set inclusion gives the stronger statement.

C.6 Proof of Proposition 3.2: Excess Risk for the Spectral L_2 Objective

We now address estimation. The objective trained in practice is the noise-contrastive estimator (NCE) [Gutmann and Hyvärinen, 2010, Ma and Collins, 2018, Zhang et al., 2022, Gao et al., 2025] for the transition-ratio score, but the theoretical guarantees that bound representation error are derived in the spectral L_2 framework of Ren et al. [2023b]; we follow the same convention and analyze the L_2 ERM here. The relationship between NCE optimization and L_2 population structure is discussed in Remark C.6 and is *not* part of the formal claim of this proposition.

Let

$$z = (s, a, s')$$

denote one observation, where

$$(s, a) \sim \rho, \qquad s' \sim P(\cdot \mid s, a).$$

For an i.i.d. sample

$$z_i = (s_i, a_i, s_i'), \qquad i = 1, \dots, n,$$

define the population L_2 risk

$$L(g) := \mathbb{E}_\rho\left[\int_{\mathcal{S}} \big(P(s' \mid s, a) - g(s, a, s')\big)^2 \, d\nu(s')\right]. \tag{19}$$

Also define

$$C_P := \mathbb{E}_\rho\left[\int_{\mathcal{S}} P(s' \mid s, a)^2 \, d\nu(s')\right], \tag{20}$$

and the centered population risk

$$\bar{L}(g) := L(g) - C_P. \tag{21}$$

The empirical objective is

$$\hat{L}_n(g) := \frac{1}{n} \sum_{i=1}^{n} \ell(g; z_i), \tag{22}$$

where the per-sample loss is

$$\ell(g; z) := -2g(s, a, s') + \int_{\mathcal{S}} g^2(s, a, u) \, d\nu(u). \tag{23}$$

We assume $C_P < \infty$ throughout. This holds, for example, whenever $P(\cdot \mid s, a) \in L^2(\nu)$ uniformly on the support of ρ. In particular, under CP realizability with a uniformly bounded density representation $|P(s' \mid s, a)| \le B_g$ and with $\nu(\mathcal{S}) = 1$, we have

$$C_P = \mathbb{E}_\rho\left[\int_{\mathcal{S}} P(s' \mid s, a)^2 \, d\nu(s')\right] \le \mathbb{E}_\rho\left[\int_{\mathcal{S}} B_g^2 \, d\nu(s')\right] = B_g^2.$$

We use the Rademacher-complexity convention fixed in Appendix C.1. For the quadratic class below, the sample is the projected sample $(s_i, a_i)_{i=1}^n$.

Lemma C.2 (Covering transfer to the integrated quadratic class)**.** *Assume $\nu(\mathcal{S}) = 1$ and $|g(s, a, s')| \le B_g$ for all $g \in \mathcal{G}$ and all (s, a, s'). Define*

$$h_g(s, a) := \int_{\mathcal{S}} g^2(s, a, u) \, d\nu(u), \qquad \mathcal{H}_{\mathcal{G}} := \{h_g : g \in \mathcal{G}\}.$$

Then, for every $\epsilon > 0$,

$$\mathcal{N}(\mathcal{H}_{\mathcal{G}}, \epsilon) \le \mathcal{N}\left(\mathcal{G}, \frac{\epsilon}{2B_g}\right). \tag{24}$$

Proof. If $\|g - g'\|_\infty \le \eta$, then for every (s, a),

$$\begin{aligned}|h_g(s,a) - h_{g'}(s,a)| &= \left| \int_{\mathcal{S}} \big(g^2(s,a,u) - g'^2(s,a,u)\big)\, d\nu(u) \right| \\ &\le \int_{\mathcal{S}} |g(s,a,u) - g'(s,a,u)|\, |g(s,a,u) + g'(s,a,u)|\, d\nu(u) \\ &\le \int_{\mathcal{S}} \eta \cdot 2B_g\, d\nu(u) = 2B_g\eta.\end{aligned}$$

Thus any η-cover of $\mathcal{G}$ in sup norm induces a $2B_g\eta$-cover of $\mathcal{H}_\mathcal{G}$. Setting $\eta = \epsilon/(2B_g)$ proves (24). □

Proposition (Restatement of Proposition 3.2)**.** *Let $\mathcal{G}$ be either $\mathcal{G}_{\text{fac}}$ or $\mathcal{G}_{\text{joint}}$, and assume*

$$\sup_{s,a,s'} |g(s,a,s')| \le B_g \qquad \text{for all } g \in \mathcal{G}.$$

Assume also that $C_P < \infty$. Let

$$\hat{g}_n \in \arg\min_{g \in \mathcal{G}} \hat{L}_n(g)$$

and

$$g^*_\mathcal{G} \in \arg\min_{g \in \mathcal{G}} L(g),$$

both assumed to exist. If the empirical minimizer exists only approximately, namely if

$$\hat{L}_n(\hat{g}_n) \le \inf_{g \in \mathcal{G}} \hat{L}_n(g) + \eta,$$

then the same bound below holds with an additional $+\eta$ term on the right-hand side.

Define the induced quadratic class

$$\mathcal{H}_\mathcal{G} := \left\{ h_g : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \;:\; h_g(s,a) = \int_{\mathcal{S}} g^2(s,a,u)\, d\nu(u), \quad g \in \mathcal{G} \right\}.$$

Then, with probability at least $1 - \delta$,

$$L(\hat{g}_n) \le L(g^*_\mathcal{G}) + 8\,\mathcal{R}_n(\mathcal{G}) + 4\,\mathcal{R}_n(\mathcal{H}_\mathcal{G}) + 2(2B_g + B_g^2)\sqrt{\frac{2\log(1/\delta)}{n}}. \tag{25}$$

Here $\mathcal{R}_n(\mathcal{H}_\mathcal{G})$ is computed on the projected sample $(s_i, a_i)_{i=1}^n$.

Moreover, both Rademacher-complexity terms admit Dudley upper bounds in terms of the covering number of $\mathcal{G}$. In particular,

$$\mathcal{N}(\mathcal{H}_\mathcal{G}, \epsilon) \le \mathcal{N}\left(\mathcal{G}, \frac{\epsilon}{2B_g}\right). \tag{26}$$

Thus the bound (25) *can be expressed entirely in terms of the metric entropy of $\mathcal{G}$.*

Proof. First, expanding the square in $L(g)$ gives

$$\begin{aligned}L(g) &= \mathbb{E}_\rho \int_{\mathcal{S}} \big(P(s' \mid s,a) - g(s,a,s')\big)^2\, d\nu(s') \\ &= C_P - 2\mathbb{E}_\rho \int_{\mathcal{S}} P(s' \mid s,a) g(s,a,s')\, d\nu(s') + \mathbb{E}_\rho \int_{\mathcal{S}} g^2(s,a,s')\, d\nu(s').\end{aligned}$$

Since $s' \sim P(\cdot \mid s,a)$ conditional on (s,a),

$$\mathbb{E}[g(s,a,s')] = \mathbb{E}_\rho \int_{\mathcal{S}} P(s' \mid s,a) g(s,a,s')\, d\nu(s').$$

Therefore

$$\bar{L}(g) = L(g) - C_P = \mathbb{E}[\ell(g;z)]. \tag{27}$$

In particular, $\hat{L}_n(g)$ is an unbiased empirical estimate of the centered risk $\bar{L}(g)$, and L and $\bar{L}$ have the same minimizers because they differ only by the constant C_P. Using empirical optimality of $\hat{g}_n$,

$$\begin{aligned} L(\hat{g}_n) - L(g^*_{\mathcal{G}}) &= \bar{L}(\hat{g}_n) - \bar{L}(g^*_{\mathcal{G}}) \\ &= [\bar{L}(\hat{g}_n) - \hat{L}_n(\hat{g}_n)] + [\hat{L}_n(\hat{g}_n) - \hat{L}_n(g^*_{\mathcal{G}})] + [\hat{L}_n(g^*_{\mathcal{G}}) - \bar{L}(g^*_{\mathcal{G}})] \\ &\le 2 \sup_{g \in \mathcal{G}} |\bar{L}(g) - \hat{L}_n(g)|. \end{aligned} \tag{28}$$

If $\hat{g}_n$ is only an η-approximate empirical minimizer, the same argument adds an extra $+\eta$ term. Let

$$\mathcal{L}_{\mathcal{G}} := \{\ell_g : z \mapsto \ell(g; z) : g \in \mathcal{G}\}.$$

By the standard symmetrization inequality for the centered empirical process [Bartlett and Mendelson, 2002, Wainwright, 2019],

$$\mathbb{E} \sup_{g \in \mathcal{G}} |\bar{L}(g) - \hat{L}_n(g)| \le 2\mathcal{R}_n(\mathcal{L}_{\mathcal{G}}). \tag{29}$$

Now write

$$\ell(g; z) = -2g(s, a, s') + h_g(s, a), \qquad h_g(s, a) := \int_{\mathcal{S}} g^2(s, a, u)\, d\nu(u).$$

By subadditivity and homogeneity of Rademacher complexity,

$$\mathcal{R}_n(\mathcal{L}_{\mathcal{G}}) \le 2\mathcal{R}_n(\mathcal{G}) + \mathcal{R}_n(\mathcal{H}_{\mathcal{G}}). \tag{30}$$

Combining (29) and (30) gives

$$\mathbb{E} \sup_{g \in \mathcal{G}} |\bar{L}(g) - \hat{L}_n(g)| \le 4\mathcal{R}_n(\mathcal{G}) + 2\mathcal{R}_n(\mathcal{H}_{\mathcal{G}}). \tag{31}$$

The quadratic term is not controlled by the usual pointwise contraction principle, since

$$h_g(s, a) = \int_{\mathcal{S}} g^2(s, a, u)\, d\nu(u)$$

integrates over all possible next states u rather than applying a scalar function to the observed value $g(s_i, a_i, s'_i)$. Instead, Lemma C.2 gives

$$\mathcal{N}(\mathcal{H}_{\mathcal{G}}, \epsilon) \le \mathcal{N}\left(\mathcal{G}, \frac{\epsilon}{2B_g}\right),$$

which is the entropy estimate (26) stated in the proposition. Dudley's entropy integral [Dudley, 1967, Wainwright, 2019] therefore bounds both $\mathcal{R}_n(\mathcal{G})$ and $\mathcal{R}_n(\mathcal{H}_{\mathcal{G}})$ in terms of the covering number of $\mathcal{G}$, with the second bound using only the radius rescaling above. It remains to pass from expectation to high probability. Define

$$F(z_1, \ldots, z_n) := \sup_{g \in \mathcal{G}} |\bar{L}(g) - \hat{L}_n(g)|.$$

Since $|g| \le B_g$ and $\nu(\mathcal{S}) = 1$,

$$|\ell(g; z)| \le 2|g(s, a, s')| + \int_{\mathcal{S}} g^2(s, a, u)\, d\nu(u) \le 2B_g + B_g^2 =: M_g.$$

Changing one sample point changes $\hat{L}_n(g)$ by at most $2M_g/n$ uniformly over g, and therefore changes F by at most $2M_g/n$. McDiarmid's inequality [McDiarmid, 1989] yields, with probability at least $1 - \delta$,

$$F \le \mathbb{E}F + M_g \sqrt{\frac{2\log(1/\delta)}{n}}.$$

Combining this display with (31) and then using (28), we obtain

$$\begin{aligned} L(\hat{g}_n) - L(g^*_{\mathcal{G}}) &\le 2F \\ &\le 8\mathcal{R}_n(\mathcal{G}) + 4\mathcal{R}_n(\mathcal{H}_{\mathcal{G}}) + 2M_g \sqrt{\frac{2\log(1/\delta)}{n}}. \end{aligned}$$

Substituting $M_g = 2B_g + B_g^2$ proves (25). □

Remark (On the constants) The prefactors in (25) come directly from the proof. The factor 2 in the excess-risk decomposition (28), the factor 2 from symmetrization (29), and the decomposition

$$\mathcal{R}_n(\mathcal{L}_\mathcal{G}) \le 2\,\mathcal{R}_n(\mathcal{G}) + \mathcal{R}_n(\mathcal{H}_\mathcal{G})$$

together give the coefficients 8 and 4 in front of $\mathcal{R}_n(\mathcal{G})$ and $\mathcal{R}_n(\mathcal{H}_\mathcal{G})$, respectively. The tail coefficient comes from the uniform bound

$$|\ell(g;z)| \le 2B_g + B_g^2$$

and the bounded-differences constant

$$\frac{2(2B_g + B_g^2)}{n}.$$

Thus these constants are explicit and problem-dependent. We avoid the phrase "universal constant" here because the dependence on B_g is essential: it enters both through the per-sample loss bound and through the covering-number rescaling

$$\mathcal{N}(\mathcal{H}_\mathcal{G}, \epsilon) \le \mathcal{N}\left(\mathcal{G}, \frac{\epsilon}{2B_g}\right).$$

Remark (Relationship to the NCE estimator) The ranking-based NCE objective of Ma and Collins [2018] used by FaStR is consistent with maximum likelihood in the limit of infinitely many negative samples, but is not identical to the L_2 spectral ERM analyzed here. Establishing a sharp finite-sample equivalence between the two estimators is separate from the claim proved above.

C.7 Proof of Proposition 3.3: Bound-Implied Sufficient Sample Size

We combine the additive covering bound from Theorem 3.1 with the excess-risk template from Proposition 3.2. The goal is to compare the sample sizes that this particular uniform-convergence analysis *certifies* as sufficient for the factored class and for the joint class.

Throughout this section, the word "certificate" means an upper bound produced by the displayed finite-sample theorem. Thus the comparison below is a comparison between two sufficient sample sizes obtained by inverting two upper bounds. It is not a comparison between the true minimax sample complexities of the two statistical problems. In particular, the joint-class bound below is the standard uniform-convergence certificate for an unrestricted Lipschitz state-action encoder. It is not a lower bound and may be loose for structured subclasses.

Lemma C.3 (Polynomial entropy implies a Rademacher rate)**.** *Let $\mathcal{F}$ be a uniformly bounded real-valued function class with*

$$\operatorname{diam}_\infty(\mathcal{F}) := \sup_{f,f' \in \mathcal{F}} \|f - f'\|_\infty \le 2R.$$

Assume that, for some $A < \infty$ and $D > 2$,

$$\log \mathcal{N}(\mathcal{F}, \tau) \le A\tau^{-D}, \qquad 0 < \tau \le R.$$

Then there exists a constant $K(A, D, R) < \infty$, independent of n, such that

$$\mathcal{R}_n(\mathcal{F}) \le K(A, D, R) n^{-1/D}. \tag{32}$$

Proof. Dudley's entropy integral, together with the fact that a sup-norm cover is also an empirical-L_2 cover, gives for any $\alpha \in (0, R]$,

$$\mathcal{R}_n(\mathcal{F}) \le 4\alpha + \frac{12}{\sqrt{n}} \int_\alpha^R \sqrt{\log \mathcal{N}(\mathcal{F}, \tau)}\, d\tau.$$

Using the polynomial entropy bound,

$$\mathcal{R}_n(\mathcal{F}) \le 4\alpha + \frac{12\sqrt{A}}{\sqrt{n}} \int_\alpha^R \tau^{-D/2}\, d\tau.$$

Since $D > 2$,

$$\int_\alpha^R \tau^{-D/2}\, d\tau \le \frac{2}{D-2} \alpha^{1-D/2}.$$

Choose $\alpha = Rn^{-1/D}$, which lies in $(0, R]$ for $n \geq 1$. Then

$$4\alpha = 4Rn^{-1/D},$$

and

$$\frac{1}{\sqrt{n}}\alpha^{1-D/2} = \frac{1}{\sqrt{n}}(Rn^{-1/D})^{1-D/2} = R^{1-D/2}n^{-1/D}.$$

Therefore

$$\mathcal{R}_n(\mathcal{F}) \leq \left(4R + \frac{24\sqrt{A}}{D-2}R^{1-D/2}\right) n^{-1/D}.$$

The coefficient is finite and independent of n, proving the claim. □

Proposition (Restatement of Proposition 3.3)**.** *Assume CP realizability, that is, $\epsilon_{\mathrm{CP}} = 0$. Also assume the containment condition of Theorem 3.1(ii), so that the true CP transition model lies in both $\mathcal{G}_{\mathrm{fac}}$ and $\mathcal{G}_{\mathrm{joint}}$.*

Fix $\delta \in (0, 1)$. Let

$$\mathcal{F}_s, \qquad \mathcal{F}_a, \qquad \mathcal{F}_m, \qquad \mathcal{F}_{sa}$$

be L-Lipschitz classes on compact domains. Let

$$D_{\max} := \max(d_s, d_a), \qquad D_{\mathrm{sum}} := d_s + d_a,$$

and assume

$$D_{\max} \geq 3.$$

Let

$$\hat{g}_n^{\mathrm{fac}} \in \arg\min_{g \in \mathcal{G}_{\mathrm{fac}}} \hat{L}_n(g), \qquad \hat{g}_n^{\mathrm{joint}} \in \arg\min_{g \in \mathcal{G}_{\mathrm{joint}}} \hat{L}_n(g)$$

be empirical minimizers for the two classes.

Then, with probability at least $1 - \delta$, the high-probability bound of Proposition 3.2 yields the certificates

$$L(\hat{g}_n^{\mathrm{fac}}) \leq C_{\mathrm{fac}}(\delta)\, n^{-1/D_{\max}}, \qquad L(\hat{g}_n^{\mathrm{joint}}) \leq C_{\mathrm{joint}}(\delta)\, n^{-1/D_{\mathrm{sum}}}, \tag{33}$$

where the constants

$$C_{\mathrm{fac}}(\delta), \qquad C_{\mathrm{joint}}(\delta)$$

are finite, positive, independent of n and of the target accuracy ϵ, and depend polynomially on the problem parameters

$$B_g, \quad B_\phi, \quad B_{m,\infty}, \quad L, \quad \mathrm{diam}(\mathcal{S}), \quad \mathrm{diam}(\mathcal{A}), \quad d,$$

and logarithmically on $1/\delta$, with fixed dependence on the dimensions d_s, d_a.

Define the bound-implied sufficient sample sizes

$$n_{\mathrm{fac}}^{\mathrm{suff}}(\epsilon) := \inf\left\{n \in \mathbb{N} : C_{\mathrm{fac}}(\delta)\, n^{-1/D_{\max}} \leq \epsilon^2\right\},$$

and

$$n_{\mathrm{joint}}^{\mathrm{suff}}(\epsilon) := \inf\left\{n \in \mathbb{N} : C_{\mathrm{joint}}(\delta)\, n^{-1/D_{\mathrm{sum}}} \leq \epsilon^2\right\}.$$

These are the smallest sample sizes at which the certificates in (33) *guarantee representation error at most ϵ^2. Then*

$$\frac{n_{\mathrm{joint}}^{\mathrm{suff}}(\epsilon)}{n_{\mathrm{fac}}^{\mathrm{suff}}(\epsilon)} = \Theta\left(\epsilon^{-2\min(d_s, d_a)}\right) \qquad \text{as } \epsilon \to 0. \tag{34}$$

Proof. Under CP realizability, there exists

$$g_{\mathrm{CP}} \in \mathcal{G}_{\mathrm{fac}}$$

such that

$$g_{\mathrm{CP}}(s, a, s') = P(s' \mid s, a).$$

Hence

$$L(g_{\mathrm{CP}}) = \mathbb{E}_\rho \int_{\mathcal{S}} \left(P(s' \mid s, a) - g_{\mathrm{CP}}(s, a, s')\right)^2 d\nu(s') = 0.$$

Since $L(g) \geq 0$ for all g, we have

$$L(g^*_{\mathcal{G}_{\text{fac}}}) = 0.$$

The containment condition gives

$$\mathcal{G}_{\text{fac}} \subseteq \mathcal{G}_{\text{joint}},$$

so the same g_{CP} also belongs to $\mathcal{G}_{\text{joint}}$. Thus

$$L(g^*_{\mathcal{G}_{\text{joint}}}) = 0.$$

Therefore the finite-sample certificates contain only estimation and concentration terms. By Theorem 3.1 and the Lipschitz entropy bound, there is a constant $A_{\text{fac}} < \infty$ such that

$$\log \mathcal{N}(\mathcal{G}_{\text{fac}}, \tau) \leq A_{\text{fac}} \tau^{-D_{\max}}, \qquad D_{\max} := \max(d_s, d_a).$$

For the unrestricted joint Lipschitz class, the standard product-domain entropy bound gives

$$\log \mathcal{N}(\mathcal{G}_{\text{joint}}, \tau) \leq A_{\text{joint}} \tau^{-D_{\text{sum}}}, \qquad D_{\text{sum}} := d_s + d_a.$$

Moreover, by Lemma C.2, the associated quadratic classes have the same entropy exponents up to constants:

$$\log \mathcal{N}(\mathcal{H}_{\mathcal{G}_{\text{fac}}}, \tau) \leq A_{\text{fac},H} \tau^{-D_{\max}},$$

and

$$\log \mathcal{N}(\mathcal{H}_{\mathcal{G}_{\text{joint}}}, \tau) \leq A_{\text{joint},H} \tau^{-D_{\text{sum}}},$$

for suitable finite constants $A_{\text{fac},H}$ and $A_{\text{joint},H}$. Since $D_{\max} \geq 3$ and

$$D_{\text{sum}} = d_s + d_a \geq D_{\max},$$

both entropy exponents are larger than 2. Applying Lemma C.3 gives constants $K_{\text{fac}}, K_{\text{joint}} < \infty$, independent of n and ϵ, such that

$$\mathcal{R}_n(\mathcal{G}_{\text{fac}}) + \mathcal{R}_n(\mathcal{H}_{\mathcal{G}_{\text{fac}}}) \leq K_{\text{fac}} n^{-1/D_{\max}},$$

and

$$\mathcal{R}_n(\mathcal{G}_{\text{joint}}) + \mathcal{R}_n(\mathcal{H}_{\mathcal{G}_{\text{joint}}}) \leq K_{\text{joint}} n^{-1/D_{\text{sum}}}.$$

Apply Proposition 3.2 to the two classes with confidence levels $\delta/2$ and take a union bound. With probability at least $1 - \delta$, both bounds hold simultaneously. Using the zero oracle risks above,

$$L(\hat{g}_n^{\text{fac}}) \leq K'_{\text{fac}} n^{-1/D_{\max}} + 2(2B_g + B_g^2)\sqrt{\frac{2\log(2/\delta)}{n}},$$

and

$$L(\hat{g}_n^{\text{joint}}) \leq K'_{\text{joint}} n^{-1/D_{\text{sum}}} + 2(2B_g + B_g^2)\sqrt{\frac{2\log(2/\delta)}{n}},$$

for finite constants K'_{fac} and K'_{joint}. Because $D_{\max} \geq 3$ and $D_{\text{sum}} \geq D_{\max}$,

$$n^{-1/2} \leq n^{-1/D_{\max}}, \qquad n^{-1/2} \leq n^{-1/D_{\text{sum}}}, \qquad n \geq 1.$$

Absorbing the concentration terms into the constants yields

$$L(\hat{g}_n^{\text{fac}}) \leq C_{\text{fac}}(\delta) n^{-1/D_{\max}}, \qquad L(\hat{g}_n^{\text{joint}}) \leq C_{\text{joint}}(\delta) n^{-1/D_{\text{sum}}}, \tag{35}$$

which proves the certificates in (33). The constants are finite, positive, independent of n and ϵ, and have the parameter dependence stated in the proposition. It remains to invert the certificates. The factored certificate guarantees

$$L(\hat{g}_n^{\text{fac}}) \leq \epsilon^2$$

whenever

$$C_{\text{fac}}(\delta) n^{-1/D_{\max}} \leq \epsilon^2.$$

This is equivalent to

$$n \geq C_{\text{fac}}(\delta)^{D_{\max}} \epsilon^{-2D_{\max}}.$$

Similarly, the joint certificate guarantees

$$L(\hat{g}_n^{\text{joint}}) \leq \epsilon^2$$

whenever

$$n \geq C_{\text{joint}}(\delta)^{D_{\text{sum}}} \epsilon^{-2D_{\text{sum}}}.$$

If sample sizes are required to be integers, ceilings change these thresholds by a multiplicative $1 + o(1)$ factor as $\epsilon \to 0$. Hence

$$\frac{n^{\text{suff}}_{\text{joint}}(\epsilon)}{n^{\text{suff}}_{\text{fac}}(\epsilon)} = \frac{C_{\text{joint}}(\delta)^{D_{\text{sum}}}}{C_{\text{fac}}(\delta)^{D_{\max}}} \epsilon^{-2(D_{\text{sum}} - D_{\max})}(1 + o(1)).$$

Finally,

$$D_{\text{sum}} - D_{\max} = d_s + d_a - \max(d_s, d_a) = \min(d_s, d_a).$$

Therefore

$$\frac{n^{\text{suff}}_{\text{joint}}(\epsilon)}{n^{\text{suff}}_{\text{fac}}(\epsilon)} = \Theta\Big(\epsilon^{-2\min(d_s, d_a)}\Big),$$

as claimed. Since the thresholds were obtained by inverting upper-bound certificates, this is an algebraic comparison of certified sufficient sample sizes, not a minimax lower bound. □

Remark (Scope and interpretation of the sample-size comparison) Equation (34) compares the sample sizes that the same uniform-convergence template certifies as sufficient for the two function classes. The comparison is therefore between

$$C_{\text{fac}}(\delta) n^{-1/D_{\max}}$$

and

$$C_{\text{joint}}(\delta) n^{-1/D_{\text{sum}}},$$

not between the true optimal risks achievable by all possible algorithms.

In particular, (34) does *not* prove that learning the joint class has minimax sample complexity

$$\Omega(\epsilon^{-2D_{\text{sum}}}).$$

Such a statement would require a matching lower bound. A matching lower bound would in turn require a packing construction showing that $\mathcal{G}_{\text{joint}}$ contains many well-separated functions at scale ϵ. That kind of packing argument generally requires an additional richness assumption on $\mathcal{F}_{sa}$, ensuring that the joint state-action encoder can realize a sufficiently large family of distinct functions over the full $d_s + d_a$ dimensional product domain.

We do not impose such a richness assumption here. Therefore, the conservative interpretation is the following: under the same generic uniform-convergence analysis, the factored class receives a certificate with exponent $D_{\max} = \max(d_s, d_a)$, whereas the unrestricted joint Lipschitz class receives the standard certificate with exponent $D_{\text{sum}} = d_s + d_a$. Inverting these two certificates yields a polynomially smaller bound-implied sufficient sample size for the factored class. Whether a more refined, algorithm-specific analysis of the joint class can beat the generic uniform-convergence certificate is a separate question. The empirical results in the main paper are consistent with this conservative reading.

Remark (Non-realizable case) When $\epsilon_{\text{CP}} > 0$, the true transition model is not represented exactly by the CP-factored class. In that case, the factored class has an irreducible approximation term. If

$$\inf_{g \in \mathcal{G}_{\text{fac}}} L(g) \leq \epsilon^2_{\text{CP}},$$

then the same finite-sample argument gives the factored certificate

$$L(\hat{g}^{\text{fac}}_n) \leq \epsilon^2_{\text{CP}} + C_{\text{fac}}(\delta) n^{-1/D_{\max}}.$$

If the joint class still contains the true transition model, then its oracle risk remains zero, and its certificate is

$$L(\hat{g}^{\text{joint}}_n) \leq C_{\text{joint}}(\delta) n^{-1/D_{\text{sum}}}.$$

Thus the non-realizable setting involves a bias–variance tradeoff. The factored class has the smaller entropy exponent and hence the faster estimation term, but it may plateau at the approximation level ϵ^2_{CP}. For moderate sample sizes and small CP approximation error, the factored certificate can still be sharper; for sufficiently large sample sizes, if the joint class is well-specified and the factored class is not, the irreducible bias term may dominate the factored bound. This is why the exact-realizability result above is stated separately.

Remark (Same-n comparison without Lipschitz instantiation) The sample-ratio in (34) relies on the Lipschitz covering bounds, which give the explicit dimensional exponents $D_{\max}$ and $D_{\sum}$. A weaker but more robust statement is available without any Lipschitz instantiation: under CP realizability and the containment condition of Theorem 3.1(ii), at every fixed n, the certified upper bound on $L^2(\nu)$ representation error from Proposition 3.2 is no larger for the factored class than for the joint class.

The argument uses only set inclusion. Appendix C.5 gives $\mathcal{R}_n(\mathcal{G}_{\mathrm{fac}}) \le \mathcal{R}_n(\mathcal{G}_{\mathrm{joint}})$ from $\mathcal{G}_{\mathrm{fac}} \subseteq \mathcal{G}_{\mathrm{joint}}$. The induced quadratic class $\mathcal{H}_{\mathcal{G}} = \{(s,a) \mapsto \int g^2(s,a,u)\, d\nu(u) : g \in \mathcal{G}\}$ inherits this monotonicity, since h_g is determined by g alone, so $\mathcal{R}_n(\mathcal{H}_{\mathcal{G}_{\mathrm{fac}}}) \le \mathcal{R}_n(\mathcal{H}_{\mathcal{G}_{\mathrm{joint}}})$. Under CP realizability both approximation terms $L(g^*_{\mathcal{G}})$ in Proposition 3.2 equal zero; with the shared envelope B_g giving the same McDiarmid tail, the certificate (25) yields a no-larger upper bound on $L(\hat{g}_n)$ for the factored class at the same n. This certifies an ordering of two upper bounds, not a minimax separation.

C.8 Conditional Implication for LSVI-UCB

The representation results above control an average transition-density error in $L^2(\nu)$. This is the right object for spectral representation learning, but LSVI-UCB requires a stronger condition: the Bellman backup must be uniformly approximable as a linear function of the feature used by the algorithm. We state this downstream implication conditionally. The point is not to derive this condition from $L^2(\nu)$ error alone, but to record what the factored structure would buy once the learned feature satisfies the standard approximate-linear condition required by LSVI-UCB.

Definition 1 ($(\eta, \mathcal{V}, W)$-approximate Bellman linearity)**.** A feature map $\widehat{\psi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ satisfies $(\eta, \mathcal{V}, W)$-approximate Bellman linearity if for every $V \in \mathcal{V}$ there exists a vector $w_V \in \mathbb{R}^d$ with $\|w_V\|_2 \le W$ such that

$$\sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| r(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}[V(s')] - \widehat{\psi}(s,a)^\top w_V \right| \le \eta. \tag{36}$$

Here $\mathcal{V}$ can be taken as the value-function class generated by the LSVI-UCB dynamic program. A stronger but more conservative version is obtained by taking $\mathcal{V}$ to be all bounded functions with $\|V\|_\infty \le R_{\max}/(1-\gamma)$.

Proposition C.4 (Conditional LSVI-UCB compatibility)**.** *Let $\widehat{\psi}$ be a learned d-dimensional feature map, with $\sup_{s,a} \|\widehat{\psi}(s,a)\|_2 \le B_\psi$. Suppose that $\widehat{\psi}$ satisfies $(\eta, \mathcal{V}_{\mathrm{LSVI}}, W)$-approximate Bellman linearity in the sense of Definition 1. Assume that LSVI-UCB is run with $\widehat{\psi}$, the standard elliptical bonus, and the usual discounted-to-effective-horizon reduction $H = \widetilde{O}((1-\gamma)^{-1})$.*

If

$$\eta \le c_0(1-\gamma)\epsilon$$

for a sufficiently small numerical constant c_0, then LSVI-UCB returns an ϵ-optimal policy after

$$\widetilde{O}\big(\mathrm{poly}\big(d, W, B_\psi, (1-\gamma)^{-1}, \epsilon^{-1}\big)\big)$$

episodes. Under the standard bounded-norm normalization used in linear-MDP analyses, this specializes to the familiar certificate

$$\widetilde{O}\left(\frac{d^2}{(1-\gamma)^4\epsilon^2}\right).$$

Proof. Under approximate Bellman linearity, the Bellman target used by LSVI-UCB is linear in $\widehat{\psi}$ up to a uniform misspecification error η. Standard misspecified-linear-MDP analyses for LSVI-UCB then yield a regret decomposition of the form

$$\mathrm{Regret}(T) \le \widetilde{O}\left(\frac{d\sqrt{T}}{(1-\gamma)^{3/2}}\right) + C_{\mathrm{mis}}\frac{T\eta}{1-\gamma}, \tag{37}$$

where the hidden factors depend polynomially on the feature and weight norm bounds. The first term is the usual exact-linear-MDP regret term, and the second term is the cumulative Bellman misspecification error.

To make the average regret at most ϵ, it suffices to choose T so that

$$\frac{d}{(1-\gamma)^{3/2}\sqrt{T}} \lesssim \epsilon,$$

which gives

$$T = \widetilde{O}\left(\frac{d^2}{(1-\gamma)^3\epsilon^2}\right).$$

At this value of T, the misspecification term in (37) contributes at most order ϵT whenever

$$\eta \lesssim (1-\gamma)\epsilon.$$

Thus the condition $\eta \le c_0(1-\gamma)\epsilon$ ensures that the misspecification term does not dominate the exact-feature regret term. Converting the average-regret guarantee to a single output policy by the standard PAC conversion for discounted problems introduces one additional factor of $(1-\gamma)^{-1}$, yielding

$$\widetilde{O}\left(\frac{d^2}{(1-\gamma)^4\epsilon^2}\right)$$

under the usual bounded-norm normalization. Keeping the norm constants explicit gives the stated polynomial dependence on $d, W, B_\psi, (1-\gamma)^{-1}$ and ϵ^{-1}. □

Implication for the factored representation. Under exact CP realizability, the true feature

$$\psi^*(s,a) = \phi_s^*(s) \odot \phi_a^*(a)$$

satisfies Bellman linearity exactly: for every bounded V,

$$r(s,a) + \gamma \mathbb{E}[V(s') \mid s,a] = \psi^*(s,a)^\top w_V$$

for an appropriate coefficient vector w_V. Thus the CP feature is a valid linear-MDP feature. The learned feature $\widehat{\psi}$ can be used by LSVI-UCB once it satisfies the stronger uniform condition (36). The representation-learning results in Appendix C.6 do not by themselves prove this uniform condition from average $L^2(\nu)$ error; rather, they provide a route for reducing the number of representation-learning samples needed to obtain an accurate transition feature.

The potential advantage of the factored structure is therefore conditional but clear. If the factored and joint learned features are both evaluated through the same downstream LSVI-UCB analysis, then the online exploration guarantee depends on the feature dimension and the Bellman misspecification level η. The factored representation can improve the upstream sample requirement for reaching a small η because its representation class has a smaller certified estimation term in the low-CP-bias regime. Once the threshold $\eta \le c_0(1-\gamma)\epsilon$ is reached, the downstream LSVI-UCB guarantee is the standard one.

C.9 Operational Reading of the Covering-Number Bound

In sample-size terms, the covering number is the number of transitions needed for the contrastive loss to separate two encoder triples within a target tolerance. A smaller covering number means the empirical loss is close to its population value at smaller n, so the resulting $\psi = \phi_s \odot \phi_a$ is available as a TD feature at smaller n. Take humanoid as a concrete case. The state has $d_s = 67$ proprioceptive coordinates, the action has $d_a = 21$ actuator commands, and the standard approach fits the transition density on the joint 88-dimensional input. The CP form fits two smaller pieces instead: a feature of the 67-dimensional state, and a feature of the 21-dimensional action that says how the action scales the latent state modes. Theorem 3.1 states this same split at the level of the covering number.

D Ablation Study and Analysis

This appendix gathers two analyses of the FaStR design. Section D.1 is a controlled encoder ablation that isolates which architectural ingredient drives the empirical gain by interpolating between CTRL-SR and FaStR through two intermediate architectures. Section D.2 varies the encoder hidden width of all three spectral methods. The complementary question, *when the underlying CP assumption is a good fit for the environment*, is treated in Appendix E.

D.1 Interaction Structure of the Spectral Encoder

FaStR differs from CTRL-SR in three design choices: the encoder is split into separate state and action networks, their outputs are combined through element-wise multiplication, and the NCE logit is a diagonal trilinear transition-ratio score rather than a fused state-action logit. We isolate the contribution of each choice with a four-condition ablation in which two intermediate architectures (concat-of-separate and Tucker) sit between the monolithic baseline and FaStR.

D.1.1 Encoder Structure of the Four Conditions

The four conditions share the TD3 backbone, optimization, RP-NCE protocol, and evaluation setup of Appendix B (Sections B.1 and B.2); they differ *only* in the encoder used to produce the spectral feature and, where the encoder structure forces it, in the matching critic.

Condition (a): CTRL-SR. The baseline of Gao et al. [2025], identical to Appendix B.3: a monolithic encoder $\varphi(s,a) \in \mathbb{R}^{512}$ on $[s;a]$, bilinear NCE logit $\varphi(s,a)^\top \mu(s')$, and an RFF-MLP critic on φ. Effective NCE dimension d=512.

Condition (b): Concat-of-separate. Two independent networks $\phi_s \colon \mathcal{S} \to \mathbb{R}^{256}$ and $\phi_a \colon \mathcal{A} \to \mathbb{R}^{256}$, concatenated and projected through a learned linear layer with Mish activation: $\psi = \sigma(W[\phi_s;\, \phi_a] + b)$, $W \in \mathbb{R}^{512\times512}$. The 256-dim outputs are sized so that the joint feature matches the 512-dim ψ of the other conditions. Each component of ψ depends additively on ϕ_s and ϕ_a. The critic is the same RFF-MLP on ψ as in FaStR. Effective NCE dimension d=512.

Condition (c): Tucker. Two independent networks $\phi_s \colon \mathcal{S} \to \mathbb{R}^{64}$ and $\phi_a \colon \mathcal{A} \to \mathbb{R}^{64}$ feeding a full trilinear score: $h(s,a,s') = \phi_s(s)^\top M(s')\, \phi_a(a)$ with $M(s') \in \mathbb{R}^{64\times64}$. This is the Tucker parameterization of Section 3.1, with effective NCE dimension $d_s \cdot d_a = 4096$. Because full trilinear scoring is incompatible with the Hadamard critic, we use a structurally matched critic: a bilinear term $\phi_s^\top W \phi_a$ plus an RFF-MLP residual on $[\phi_s;\phi_a]$. The dimension d_s=d_a=64 keeps total parameters and per-step floating-point operations (FLOPs) within 10% of FaStR's.

Condition (d): FaStR (ours). Identical to Appendix B.2: diagonal trilinear NCE score $(\phi_s(s) \odot \phi_a(a))^\top m(s')$ with $\phi_s, \phi_a, m \to \mathbb{R}^{512}$, factored critic feature $\psi = \phi_s(s) \odot \phi_a(a)$, and an RFF-MLP critic on ψ. Effective NCE dimension d=512=B.

The critic architecture varies only where the interaction structure forces it: FaStR's diagonal trilinear score induces the same Hadamard feature used by the critic, while Tucker's full interaction matrix requires a bilinear-residual head.

D.1.2 Results

Figure 4 reports learning curves on five DM Control Suite tasks at two action dimensions. Humanoid tasks ($\dim \mathcal{A}$=21) are the primary comparison because the factored advantage is largest in high-dimensional action spaces; quadruped tasks ($\dim \mathcal{A}$=12) act as a control where all methods are expected to perform comparably.

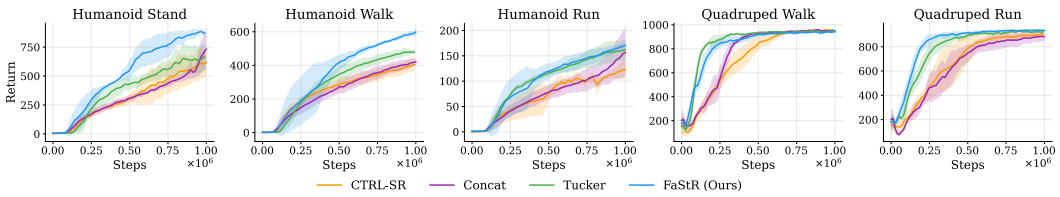


Figure 4: NCE transition-ratio score structure ablation. Learning curves on five DM Control Suite tasks at two action dimensions: Humanoid ($\dim \mathcal{A}$=21) is the primary comparison; Quadruped ($\dim \mathcal{A}$=12) is a control. Curves: mean return across 8 seeds; shaded: ± 1 std.

Encoder separation alone does not explain the gain. The (a)-vs-(b) contrast isolates the effect of using separate ϕ_s and ϕ_a networks without multiplicative interaction. Concat preserves the split but

combines through a learned affine transformation, so each component of ψ is a linear mixture of ϕ_s and ϕ_a. On humanoid tasks, concat gives a modest improvement over the monolithic encoder; on quadruped, the two are within noise. The improvement is well below FaStR's gain on the same tasks, ruling out the hypothesis that the split architecture alone accounts for FaStR's advantage.

The diagonal trilinear interaction is the source of the gain. The (b)-vs-(d) contrast isolates the CP interaction in the transition-ratio score versus linear projection with encoder separation held constant. FaStR gives higher returns than concat on every humanoid task, despite using the same 512-dimensional ψ and the same RFF-MLP critic. The structural reason is that concat gives an additive scoring function $\psi_k = \sum_j (W_s)_{kj}\phi_{s,j} + \sum_j (W_a)_{kj}\phi_{a,j} + b_k$, mixing all dimensions, while FaStR's NCE logit contains the per-component product $\phi_{s,k}(s) \cdot \phi_{a,k}(a) \cdot m_k(s')$: the k-th action feature directly scales the k-th state feature before comparison with the k-th next-state factor. This per-component product is the algebraic structure required by Theorem 3.1 for the log covering number to decompose additively over $\mathcal{F}_s$, $\mathcal{F}_a$, $\mathcal{F}_m$. Concat loses this decomposition at the projection layer, where W introduces cross-mode coupling.

Tucker is more expressive but trains worse under NCE. The (c)-vs-(d) contrast tests whether the strictly more expressive Tucker trilinear score translates into better contrastive performance. Tucker can represent arbitrary cross-component products $\phi_{s,i} \cdot \phi_{a,j}$, $i \neq j$, that diagonal CP excludes. Empirically Tucker gives lower returns than FaStR on humanoid tasks and matches FaStR on quadruped, where the gradient-coverage constraint is weaker.

The cause is the gradient rank deficiency of Section 3.1: $\nabla_{\mathrm{vec}\, M(s'_j)}\ell$ lies in $\mathrm{span}\{\phi_a(a_i) \otimes \phi_s(s_i)\}_{i=1}^B$, a subspace of dimension at most $\min(B, d_s d_a)$. In condition (c), $d_s d_a$=4096 while B=512, so each mini-batch constrains only 512 of 4096 parameter directions of $M(s')$. Three requirements then conflict: encoder capacity, NCE validity ($B \geq d_s d_a$), and gradient coverage in a single batch. At standard batch size, Tucker cannot satisfy all three simultaneously.

Low-rank Tucker does not recover the gap. The natural fix is to lower d_s, d_a until $d_s d_a \leq B$. The boundary case d_s=d_a=22 ($d_s d_a$=484 $\leq B$) on humanoid-walk gives near-zero return: encoder dimension becomes the binding constraint. Any low-rank Tucker variant within the validity constraint lies between this point and dense Tucker, so neither end matches CP.

FaStR avoids the trilemma by restricting $M(s')$ to a diagonal $m(s') \in \mathbb{R}^d$, with d parameters per next-state and gradient spanning all of $\mathbb{R}^d$ whenever $B \geq d$ (d=B=512 here). The rank-one chain-rule structure of Section 3.1 holds for any trilinear-score loss $\ell = f(\phi_s^\top M \phi_a)$, so the same mechanism applies to the energy-based score matching of SPEDER [Ren et al., 2023b] and the diffusion objective of Diff-SR [Shribak et al., 2024].

What CP gives up. CP excludes cross-component products $\phi_{s,i} \cdot \phi_{a,j}$ with $i \neq j$, which introduces approximation bias when action dimensions are tightly coupled. Whenever the Tucker core is approximately superdiagonal after a change of basis, the deep encoders can absorb the rotation internally and the CP restriction does not reduce expressiveness. CP's statistical advantage can also outweigh the approximation bias under limited training data, which is the typical online RL regime. Tucker is preferable when the training objective is not constrained by contrastive trilinear scoring (e.g., regression or MLE with enough data to span $\mathbb{R}^{d_s d_a}$), as in prior bilinear-MDP work [Du et al., 2021, Yang and Wang, 2020], and when the environment has cross-component couplings that no encoder rotation can absorb.

D.2 Encoder Hidden-Width Ablation

The main comparison aligns the hidden width of the spectral methods rather than their exact parameter count (Section 4.1): at a fixed width, the remaining parameter differences arise from the representation architecture itself, whereas matching parameter counts would require a different width per method and introduce another capacity confound. To check that the conclusions do not depend on the default width, we vary the encoder hidden width for FaStR and both spectral baselines (CTRL-SR and Diff-SR) under a fully controlled protocol. All methods share the same codebase, TD3 backbone, training budget (1×10^6 frames), optimization settings, evaluation protocol, and representation dimension (d=512); only the encoder hidden width is varied (256/512/1024), without per-method tuning.

Table 3 reports the final return (mean over the last 10 evaluations) as mean $\pm$ std over 8 seeds, following the protocol of Table 1.

Table 3: Encoder hidden-width ablation for the three spectral methods. Final return, mean $\pm$ std over 8 seeds. Bold: best mean return at each width.

Task	Width	FaStR	CTRL-SR	Diff-SR
Humanoid-Stand	256	$\mathbf{848 \pm 18}$	846 ± 80	825 ± 44
	512	$\mathbf{858 \pm 26}$	752 ± 94	761 ± 129
	1024	$\mathbf{870 \pm 22}$	698 ± 119	715 ± 42
Humanoid-Walk	256	$\mathbf{583 \pm 50}$	408 ± 11	439 ± 35
	512	$\mathbf{591 \pm 15}$	394 ± 37	440 ± 93
	1024	$\mathbf{743 \pm 85}$	439 ± 35	645 ± 81
Humanoid-Run	256	$\mathbf{200 \pm 69}$	118 ± 65	105 ± 70
	512	$\mathbf{174 \pm 19}$	118 ± 13	107 ± 35
	1024	$\mathbf{168 \pm 34}$	98 ± 16	118 ± 69
Dog-Stand	256	$\mathbf{921 \pm 15}$	855 ± 21	819 ± 101
	512	$\mathbf{955 \pm 20}$	870 ± 70	909 ± 27
	1024	$\mathbf{923 \pm 44}$	877 ± 24	910 ± 31
Dog-Run	256	377 ± 29	$\mathbf{393 \pm 44}$	275 ± 60
	512	$\mathbf{538 \pm 49}$	505 ± 97	310 ± 94
	1024	$\mathbf{521 \pm 40}$	$\mathbf{521 \pm 43}$	450 ± 73

FaStR obtains the best mean return on the Humanoid tasks at all three widths, so the gains do not depend on the default width of 512. On Humanoid-Stand, both CTRL-SR and Diff-SR become worse as the width increases, and on Humanoid-Walk and Dog-Stand FaStR remains the best at all three widths. Since the ranking is stable across the three widths, it is not specific to the default hidden width. Dog-Run is the only exception: CTRL-SR is slightly ahead at width 256, and the two methods tie at width 1024. This is consistent with Section 4.2, where stronger state-action coupling in the Dog tasks partly offsets the factored advantage.

E When Does the Factorization Help, and When Does It Hurt?

FaStR is a deliberate structural bias, and it does not provide a uniform improvement over joint encoders. This appendix locates the boundary of its applicability. Section E.1 is an offline diagnostic that estimates the cost of imposing the CP factorization on the dynamics of each task, independently of the RL training pipeline. Section E.2 reports seven additional official DMControl tasks chosen to stress the assumption, with a structural explanation for each outcome. Section E.3 lists every evaluated task.

E.1 Empirical CP Misfit Diagnostic

The finite-sample certificate of Proposition 3.2 decomposes representation error into a CP approximation bias ϵ_{CP}^2 and an estimation term whose rate depends on the covering number. Proposition 3.3 shows that, under low CP bias, the estimation advantage of the factored class over the joint class scales with $\min(d_s, d_a)$. The single-task experiments are consistent with this prediction but do not measure ϵ_{CP} directly. We construct an offline diagnostic that estimates the transition-modeling cost of enforcing CP, independently of the RL training pipeline, providing a structural complement to the per-task return numbers in Table 1.

E.1.1 Implementation Details

For each of the 13 DM Control Suite tasks in Figure 5, we collect a replay buffer of 500k transitions from a trained CTRL-SR agent across 5 seeds. Buffers are sourced from CTRL-SR rather than FaStR so that the state-action distribution is not shaped by the CP-factored encoder under evaluation, and they cover a range from early random exploration to trained locomotion, reflecting the RL-relevant state distribution.

We train two transition-modeling predictors on each buffer: the CP predictor scores transitions as $g_{\mathrm{CP}}(s, a, s') = (\phi_s(s) \odot \phi_a(a))^\top m(s')$ with separate state and action encoders, and the joint predictor uses $g_{\mathrm{joint}}(s, a, s') = \varphi(s, a)^\top \mu(s')$ with a single monolithic encoder. Both share feature dimension (d=512), encoder depth (2 residual blocks with LayerNorm and Mish), learning rate (10^{-4}), batch size (512), training epochs (50), and noise schedule (25 VP levels). The CP predictor has more parameters in every task, making the comparison conservative: if it still incurs higher test loss, the gap reflects structural misfit rather than underfitting. Each predictor is trained with the RP-NCE ranking loss plus an auxiliary conditional-moment head (weight 0.1) that predicts standardized next-state features (raw observations and random Fourier projections).

Data is split 70/15/15 into train, validation, and test. We select the checkpoint with the lowest validation NCE loss and report two test-set metrics: the NCE misfit $\widehat{\Delta}_{\mathrm{CP}}^{\mathrm{NCE}} = \mathcal{L}_{\mathrm{CP}}^{\mathrm{test}} - \mathcal{L}_{\mathrm{joint}}^{\mathrm{test}}$, which matches the contrastive objective used in the RL pipeline, and the moment misfit $\widehat{\Delta}_{\mathrm{CP}}^{\mathrm{mom}}$, which is independent of NCE. Both are distribution-weighted empirical proxies for ϵ_{CP}^2 rather than direct estimates.

E.1.2 Results

Figure 5 reports the diagnostic across 13 tasks with action dimensions from 2 to 38. The CP factorization incurs a small excess transition-modeling loss in nearly all tasks. Walker-Walk is an outlier on both metrics: it has the highest NCE misfit and the highest moment misfit of any task in the figure.

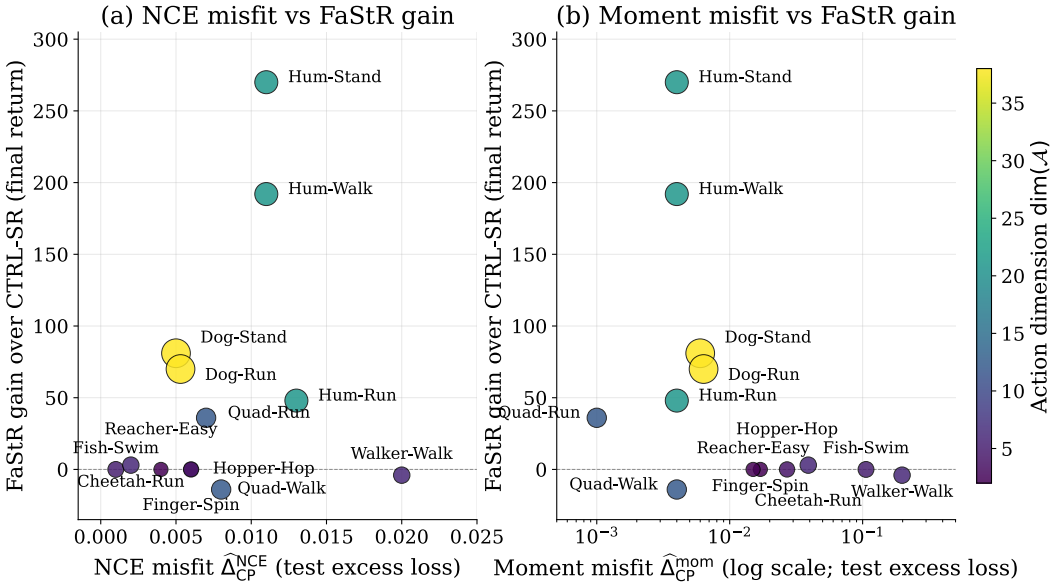


Figure 5: Empirical CP misfit diagnostic across 13 DM Control Suite tasks. Each marker is one task; the x-axis is the test-set excess loss of the CP predictor over the joint predictor (positive means CP is worse), measured by (a) NCE misfit $\widehat{\Delta}_{\mathrm{CP}}^{\mathrm{NCE}}$ and (b) moment misfit $\widehat{\Delta}_{\mathrm{CP}}^{\mathrm{mom}}$ (log scale); the y-axis is the final-return gain FaStR − CTRL-SR from Figure 2. Marker color and size encode action dimension $\dim \mathcal{A}$. The CP predictor has more parameters than the joint predictor in every task.

Two-condition pattern. The gains match the prediction of Propositions 3.2 and 3.3. Among low-dimensional tasks ($\dim \mathcal{A} \leq 6$, lower-left cluster of Figure 5), CP fits the dynamics well, but the covering-number separation at small $\min(d_s, d_a)$ is too small to yield a measurable return difference. Among high-dimensional tasks ($\dim \mathcal{A} \geq 12$), where the theory predicts a larger estimation advantage, gains appear in every case where CP misfit is low: the Humanoid and Dog points sit in the upper region of both panels, while the high-misfit Walker-Walk point, with the same action dimension as Cheetah-Run, has no gain, which corresponds to the regime of Remark C.7 in which the bias term offsets the estimation advantage. Neither low misfit nor high action dimensionality alone is sufficient: both conditions must hold together.

Scope of the diagnostic. The misfit metrics do not rank-order gain magnitudes within the low-misfit group. This is expected, since the estimation advantage depends both on the action dimension and on the effective complexity of the dynamics within the factored class, neither of which is captured by a scalar proxy. We therefore use the diagnostic as a compatibility check for the CP assumption rather than as a continuous predictor of gain magnitude.

E.2 Complementary Tasks and Failure Modes

The main evaluation targets domains with $\dim \mathcal{A} \geq 6$. Here we report seven additional official DMControl tasks outside that set: Acrobot-Swingup, Hopper-Hop, Hopper-Stand, Fish-Swim, Finger-Turn-Hard, Finger-Spin, and Reacher-Hard. They cover manipulation, underactuated control, and low-dimensional locomotion, and were selected to match, as closely as possible within DMControl, the structures under which the CP assumption is expected to be strained: strongly state-dependent underactuation, repeated contact switching, object contact, and whole-body coupling. In these regimes, Proposition 3.3 predicts either a small estimation advantage (small $\min(d_s, d_a)$) or a CP bias large enough to offset it. Finger-Spin, Fish-Swim, and Hopper-Hop already appear among the 13 tasks of the diagnostic in Section E.1. All runs follow the protocol of Table 1 (8 seeds, same interaction budget). CTRL-SR is the controlled comparison; SAC, in the 3-layer configuration of Appendix B.5 whose hidden width matches the spectral methods, and TD7 provide additional points of comparison. Results are reported per task rather than in aggregate.

Table 4: Final return on seven complementary DMControl tasks, mean $\pm$ std over 8 seeds, same protocol as Table 1. Bold: best mean return in each column.

	Acrobot Swingup	Hopper Hop	Hopper Stand	Fish Swim	Finger Turn Hard	Finger Spin	Reacher Hard
$\dim \mathcal{A}$	1	4	4	5	2	2	2
FaStR (Ours)	184 ± 70	318 ± 124	910 ± 41	664 ± 60	964 ± 19	$\mathbf{983 \pm 29}$	970 ± 17
CTRL-SR	265 ± 63	$\mathbf{564 \pm 36}$	919 ± 52	660 ± 48	$\mathbf{972 \pm 45}$	982 ± 16	$\mathbf{989 \pm 30}$
SAC(3-layer)	$\mathbf{288 \pm 87}$	209 ± 56	$\mathbf{943 \pm 10}$	$\mathbf{799 \pm 35}$	906 ± 59	951 ± 35	916 ± 44
TD7	88 ± 75	207 ± 104	670 ± 285	86 ± 166	470 ± 399	435 ± 178	898 ± 12

FaStR provides no gain on any of the seven tasks. It is 246 points below CTRL-SR on Hopper-Hop and 81 below on Acrobot-Swingup, and comparable on the other five, where the two spectral methods are within one standard deviation of each other. Both spectral methods trail SAC on Acrobot-Swingup, Hopper-Stand, and Fish-Swim. Each outcome corresponds to a specific way in which one of the two conditions of the theory is not met.

Acrobot-Swingup: state-dependent underactuation. The single elbow torque is transmitted through the current joint configuration, so the effect of the same command changes substantially with the pose, and a successful swing-up also requires the action to be timed with the phase of the motion. These strongly state-dependent action effects place more burden on a small set of matched latent factors. At the same time, the action dimension is only 1, so even a good factorization offers almost no reduction in estimation complexity. The theory therefore predicts no gain in this setting.

Hopper-Hop versus Hopper-Stand: contact switching. The two Hopper tasks share the same robot, action space, and encoder architecture. During Hopper-Hop the system repeatedly switches between flight and ground support, and the role of an action changes sharply across these contact modes. Encoding this dependence through the state features requires contact-specific latent factors and leaves less reusable low-rank structure. Hopper-Stand, in contrast, spends most of its time in a stable support mode and is only 9 points below CTRL-SR. Since the embodiment and action dimension are unchanged, this contrast is consistent with frequent contact switching being an important source of structural mismatch.

Finger-Turn-Hard and Finger-Spin: object contact. These tasks create a related challenge through changing finger–object contact. FaStR provides no gain on either task.

Fish-Swim: whole-body coupling. The force generated by each fin propagates through the entire deformable body and depends on the global body pose. The effect of an action is therefore distributed

across many state variables rather than being reused through a small number of stable local motion patterns. This type of whole-body coupling calls for a richer interaction model.

Reacher-Hard: compatible structure, low dimension. Reacher-Hard is the complementary control case. It is fully actuated and contact-free, and its dynamics are comparatively compatible with a factored representation. Its action dimension is only 2, so the joint representation is already inexpensive to estimate and the factorization provides little statistical advantage. As expected, FaStR and CTRL-SR perform similarly.

Summary. The results separate two cases. When the dynamics require many context-specific state–action interactions, the low-rank structural bias introduces enough approximation error to lower the return (Hopper-Hop, Acrobot-Swingup). On the remaining five tasks the action dimension is at most 5, the factorization has little estimation complexity to reduce, and the two spectral methods perform alike. Together with the gains on Humanoid and Dog-Stand and the competitive result on Dog-Run, these experiments support the two-condition view: FaStR is most useful when the transition can be compressed into a small number of reusable latent modes *and* the relevant state or action dimension is large enough for this reduction to matter.

E.3 Task Coverage

Table 5 lists every task on which RL returns are reported, its action dimension, and whether it belongs to the main dimension-scaling evaluation (Section 4.2) or to the complementary set (Section E.2). In DMControl, action dimensions of at least 6 appear only in the articulated locomotion domains of the main set; all other official tasks have action dimensions between 1 and 5.

Table 5: Coverage of evaluated DMControl tasks.

Domain	Task	$\dim \mathcal{A}$	Evaluation set
Cheetah	Run	6	Main (dimension scaling)
Walker	Walk, Run	6	Main (dimension scaling)
Quadruped	Walk, Run	12	Main (dimension scaling)
Humanoid	Stand, Walk, Run	21	Main (dimension scaling)
Dog	Stand, Run	38	Main (dimension scaling)
Acrobot	Swingup	1	Complementary
Reacher	Hard	2	Complementary
Finger	Spin, Turn Hard	2	Complementary
Hopper	Stand, Hop	4	Complementary
Fish	Swim	5	Complementary

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 Section 1 list three contributions: (1) a factored CP view of spectral RL representations (Section 3.1, Assumption 1); (2) a smaller sufficient-sample upper bound under CP structure, established by the additive covering-number decomposition (Theorem 3.1, Appendix C.3), the matching finite-sample bound (Proposition 3.2, Appendix C.6), the sufficient-sample-size separation (Proposition 3.3, Appendix C.7), and the conditional LSVI-UCB compatibility (Proposition C.4, Appendix C.8); and (3) empirical evaluation and modular transfer on the DM Control Suite, supported by Figure 2 and Table 1 (Section 4.2) and Figure 3 (Section 4.3). The scope of the empirical claim is stated in the introduction: FaStR obtains the highest mean return on the five tasks with $\dim \mathcal{A} \geq 21$ and matches or exceeds the joint encoder at lower action dimensions, and Appendix E reports seven further tasks on which it provides no gain or lowers the return. The CP-rank assumption is stated explicitly in Assumption 1, and the regime in which the bound predicts a gain (non-trivial $\min(d_s, d_a)$ and small CP bias) is made explicit in the discussion of Proposition 3.3, tested empirically in Section 4.2, and measured directly by the CP-misfit diagnostic of Appendix E.1.

   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: Limitations of the framework are discussed at four points in the paper. (i) The discussion following Proposition 3.3 (Section 3.2) makes explicit that two conditions are required for the bound to predict an empirical gain: non-trivial $\min(d_s, d_a)$ and small CP bias ϵ_{CP}; the comparison is between two upper bounds rather than between minimax complexities (Remark C.7). (ii) Section 4.2 reports that on low-action-dimensional tasks (Cheetah-Run, Walker-Walk, Walker-Run with $\dim \mathcal{A} \leq 6$) FaStR matches the spectral baselines without a significant gain, that on Quadruped-Walk the final returns coincide, and that on Dog-Run the improvement is within the across-seed variability; TD-MPC2 obtains the highest return on Cheetah-Run and Walker-Walk. (iii) Appendix E locates the boundary of the method: the offline CP-misfit diagnostic (Appendix E.1) measures the cost of the CP assumption on 13 tasks, and Appendix E.2 reports seven additional tasks on which FaStR provides no gain, including two (Hopper-Hop, Acrobot-Swingup) on which it is below the joint encoder, each with a structural explanation (contact switching, state-dependent underactuation, object contact, whole-body coupling). (iv) Appendix A discusses the two-axis CP form, the diagonal-versus-Tucker core trade-off (with the controlled ablation in Appendix D.1), and the restriction to proprioceptive simulation. Appendix D.2 checks that the Humanoid ranking does not depend on the encoder width. Settings outside the DM Control Suite proprioceptive suite (e.g., pixel observations, real-robot deployment) and the official DMControl tasks not listed in Table 5 are not evaluated and are not claimed.

   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: The single substantive structural assumption is Assumption 1 (CP-Factored MDP), formally stated in Section 3.1; standing boundedness, measure-theoretic, and Lipschitz conventions are collected in Appendix C.1. Each theoretical result is numbered and proved in a self-contained appendix: the Q-linearity claim (Equation 6) is derived in Appendix C.2; Theorem 3.1 (covering-number decomposition) is proved in Appendix C.3, with the Rademacher monotonicity component stated separately in Appendix C.5; Proposition 3.2 (finite-sample generalization) is proved in Appendix C.6; Proposition 3.3 (sufficient-sample-size separation) is proved in Appendix C.7; and Proposition C.4 (conditional LSVI-UCB compatibility) is proved in Appendix C.8. Each proof states which auxiliary results from the literature it invokes and supplies all intermediate inequalities inline.

   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: Section 4.1 specifies the environment suite, action-dimensionality groups, training budget (1×10^6 environment frames), action repeat, evaluation cadence and protocol, and the number of seeds. Appendix B provides per-method implementation details: shared training protocol (Appendix B.1), full FaStR architecture and optimization (Appendix B.2), the matched CTRL-SR configuration (Appendix B.3), and the Diff-SR, SAC, TD7, and TD-MPC2 baseline configurations (Appendices B.4–B.7). Table 2 (Appendix B.8) consolidates the hyperparameters of FaStR and the spectral, SAC, and TD7 baselines; TD-MPC2 uses its official default single-task configuration. Algorithm 1 gives the full per-step training procedure for FaStR. The actuator-shift protocol used in Section 4.3 (gain-scaling and polarity-inversion ranges, permutation procedure, freeze-and-adapt setup) is described in that section. The encoder-structure ablation (four conditions, Appendix D.1.1), the encoder-width ablation (Appendix D.2), the offline CP-misfit diagnostic (buffer collection, predictor architectures, data split, model selection, and metrics; Appendix E.1.1), and the seven complementary tasks (Appendix E.2) each state their own protocol, and all share the training and evaluation protocol of Appendix B.1.

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: Code is publicly available at `https://github.com/junyiwu-dev/fastr`. The repository contains all training and evaluation code for FaStR and the baselines in Table 1 (CTRL-SR, Diff-SR, SAC in both width-depth variants, TD7, and the integration of the official TD-MPC2 implementation into the shared pipeline), the configuration files

used in every reported run, and the scripts that produce Figures 2, 3, 4, and 5 and Tables 1, 3, and 4 from TensorBoard event files. The `README.md` gives installation steps, the exact command lines used to reproduce the single-task, actuator-transfer, ablation, width-sweep, CP-misfit, and complementary-task experiments, hardware and runtime expectations, and a mapping from each script output to the corresponding figure or table in the paper. The repository is derived from the open-source `spectral-rl` codebase (`https://github.com/spectral-rl/spectral-rl`); the provenance of every component is documented in `ATTRIBUTION.md`, and the original `LICENSE` file is preserved (see the Licenses item below). The DM Control Suite benchmark [Tassa et al., 2018] is publicly available and is consumed through standard `dm_control`/`shimmy` interfaces; no new data are introduced. Raw training logs and the replay buffers used by the CP-misfit diagnostic are not included due to size; the scripts regenerate them from the released configurations.

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: Section 4.1 states the environment suite, action-dimension groups, training duration, action repeat, evaluation protocol, and the number of seeds (8). Appendix B provides full per-method details: optimizer (Adam), learning rates, batch size, target update schedule, exploration noise, target action smoothing, encoder and critic architectures, RP-NCE noise schedule and negative-sample count, observation normalization, and warm-start procedure. Table 2 consolidates every hyperparameter side-by-side across FaStR, CTRL-SR, Diff-SR, SAC(2-layer), SAC(3-layer), and TD7. Hyperparameters for FaStR and CTRL-SR are matched to the open-source CTRL-SR configuration [Gao et al., 2025] without per-task tuning; Diff-SR uses the open-source implementation of Shribak et al. [2024] on the same TD3 backbone; TD7 and TD-MPC2 use the original authors' values without modification; and SAC(3-layer) uses the hidden width of the spectral methods, as documented in Appendices B.5–B.7. The width ablation (Appendix D.2) varies only the encoder hidden width ($256/512/1024$) with no per-method retuning, and the CP-misfit diagnostic (Appendix E.1.1) states its 70/15/15 train/validation/test split, checkpoint selection on validation loss, and the two test-set metrics.

   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: All learning-curve figures (Figures 2, 3, and 4) include ± 1 standard deviation shading, and Tables 1, 3, and 4 report mean ± 1 standard deviation. The source of variability is random seeds: 8 seeds for every reported RL experiment (single-task comparison, actuator-transfer, encoder-structure ablation, width ablation, and the seven complementary tasks), as stated in the figure and table captions. The CP-misfit diagnostic uses replay buffers from 5 seeds per task. Standard deviations are taken across seeds (raw, not bootstrapped). Final-return numbers in the tables are computed as the mean over the last 10 evaluation checkpoints per seed, then averaged across seeds; bolding marks the best mean return in each column. In the text, a gain is called larger than noise only when the margin exceeds the across-seed standard deviation of both methods, and results within one standard deviation (Dog-Run, Quadruped-Walk, and five of the seven complementary tasks) are described as comparable.

   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: Reported in the repository `README.md` and consistent with the per-task budget stated in Appendix B.1. All experiments were run on NVIDIA RTX 4090 (24 GB) GPUs rented from a cloud provider; the original experiments used one GPU, and the experiments added during the review period (TD-MPC2, the encoder-width ablation, the seven complementary tasks, the CP-misfit diagnostic, and the additional seeds of Table 1) used up to four GPUs in parallel. GPU memory usage stays under 4 GB per run, so two runs typically share one GPU; the physics simulation runs on CPU, which is the main bottleneck. On one GPU, a single seed on the 1×10^6-frame schedule takes approximately 4–6 hours of wall-clock time for the spectral methods (FaStR, CTRL-SR, Diff-SR), depending on task dimensionality (Cheetah/Walker ~ 4 h; Humanoid/Dog ~ 6 h); 1.5–4 hours for SAC and

TD7; and 3–16 hours for TD-MPC2 (Cheetah/Walker 3–6 h; Quadruped 5–10 h; Humanoid 8–16 h). The offline CP-misfit diagnostic trains two small predictors for 50 epochs on a stored buffer and costs a small fraction of one RL run. The reported experiments comprise roughly 1,200 training runs (Table 1: 560; Figure 3: 128; Figure 4: 80 beyond Table 1; Table 3: 240 beyond Table 1; Table 4: 224), which we estimate at roughly 2,800 GPU-hours, counting one GPU running two jobs for one hour as one GPU-hour. Preliminary experiments, debugging, and hyperparameter exploration consumed an additional comparable amount of compute that is not reflected in the final results.

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: This work develops a representation learning method for reinforcement learning in simulated continuous-control environments. It does not involve human subjects, personal data, scraped content, or deceptive practices, and uses only standard publicly available benchmarks and open-source code. The research conforms with 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: This is foundational research on the statistical structure of representations for reinforcement learning, evaluated on standard simulated locomotion benchmarks. The contribution is an algorithmic and theoretical refinement of an existing representation-learning pipeline rather than a deployable system, a generative model, a new dataset, or a model whose capabilities create a direct path to specific malicious or unintended applications. The work inherits the same general dual-use considerations as the broader continuous-control RL literature it builds on (improvements in sample efficiency could, in the long run, lower the cost of building autonomous systems), but does not introduce capabilities or artifacts that warrant a separate societal impact analysis beyond what applies to that literature as a whole.

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: This paper does not release pre-trained models, scraped datasets, language models, image generators, or other artifacts with high misuse risk. The contribution is a representation learning algorithm evaluated on standard simulated benchmarks; the public code repository contains training scripts and configurations, not released model weights.

    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: [Yes]

    Justification: All external assets are credited at the point of use and listed in the bibliography. The DeepMind Control Suite [Tassa et al., 2018] is used as the evaluation benchmark and is released under the Apache 2.0 License; the underlying physics simulator MuJoCo is used through `dm_control` and is also released under the Apache 2.0 License. The FaStR implementation is derived from the open-source `spectral-rl` codebase (`https://github.com/spectral-rl/spectral-rl`; project page `https://spectral-rl.github.io/`), which provides the CTRL-SR [Gao et al., 2025] and Diff-SR [Shribak et al., 2024] implementations used as baselines; it is released under the MIT License (Copyright (c) 2022 typoverflow). Its authors ask users to cite the papers the library implements, and the ones relevant to this work are cited in Sections 2 and 5: CTRL [Zhang et al., 2022], SPEDER [Ren et al., 2023b], LV-Rep [Ren et al.,

2023a], Diff-SR [Shribak et al., 2024], and CTRL-SR [Gao et al., 2025]. Our repository (`https://github.com/junyiwu-dev/fastr`) preserves that `LICENSE` file and copyright notice unchanged, documents in `ATTRIBUTION.md` which components are reused unchanged, modified, newly added, or removed relative to the upstream, and releases our own modifications under the MIT License. The TD7 implementation [Fujimoto et al., 2023] and the official TD-MPC2 implementation [Hansen et al., 2024] are incorporated as separate subpackages with their original `LICENSE` files; both are released under the MIT License. The SAC baseline [Haarnoja et al., 2018] is reimplemented within the same codebase and the original publication is cited.

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 datasets, pre-trained model weights, or benchmark suites. The contribution is algorithmic and theoretical. The public code repository (`https://github.com/junyiwu-dev/fastr`) is a reproducibility artifact rather than a new dataset or model; its provenance, license terms, dependencies, and usage are nonetheless documented in its `README.md` and `ATTRIBUTION.md` files.

    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: This work does not involve crowdsourcing 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: This work does not involve human subjects research.

    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: LLMs are not a component of the core methodology. Any LLM usage was limited to writing assistance (grammar and phrasing), which does not require declaration per NeurIPS policy.

    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.