SQUARED FAMILY MDPS: PROVABLY EFFICIENT RL WITH TRACTABLE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Regret bounds in continuous reinforcement learning require structural assumptions on the underlying MDPs. Many model-based parametric approaches are either difficult to theoretically analyse or hard to implement. We address this issue by proposing the Squared Family MDPs of which the transition kernels are parameterised by a family of tractable probabilistic models called squared families. Theoretically, we derive a $\tilde{O}(dH\sqrt{T})$ regret bound, sub-linear in total time T, linear in feature dimension d and horizon length H. Practically, we propose Representation RL with the Squared Neural Family (RESNEF), which uses Maximum Likelihood Estimation to learn the transition density and extract useful representations for downstream RL tasks. RESNEF brings up to 40% performance improvements compared with state-of-the-art baselines across diverse continuous control tasks.

1 INTRODUCTION

Reinforcement learning (RL) is a central tool for sequential decision making with unknown environments. Agents in RL need to explore the world to gain new information, but it naturally incurs regrets that measure the differences between the best rewards an agent can collect and the reward it gains during learning.

Early theoretical work indicates that near-optimal regret bounds can be achieved sub-linearly in time T, $O(|S|\sqrt{|A|T})$ (Auer et al., 2008; Osband et al., 2013), but such bounds also depend on the cardinality of the state and action spaces, making them insufficient for continuous cases (e.g. robotics). Recently, linear MDPs (Jin et al., 2020; Agarwal et al., 2020; Uehara et al., 2021) have been proposed to bound regret in terms of the intrinsic feature dimension instead of the cardinality of the space. However, not all transition densities can be captured by linear MDP formulations, which motivates Chowdhury et al. (2021) to consider densities that are parameterised with exponential families. Later, more abstract notions such as Eluder dimensions (Russo & Van Roy, 2013; Osband & Van Roy, 2014; Jin et al., 2021) and bilinear classes (Du et al., 2021) have also been proposed to analyse continuous MDPs.

Although theoretical analysis has become more general, practical difficulties remain. It is often unrealistic to assume that features are known in advance (Jin et al., 2020; Chowdhury et al., 2021; Yang & Wang, 2020; Ouhamma et al., 2023). Some works show that MLEs can be used to learn these features while keeping sample efficiency (Chowdhury et al., 2021; Ouhamma et al., 2023; Uehara et al., 2021; Agarwal et al., 2020), but practically the normalisation terms of the underlying density are often intractable to compute. Indeed, existing empirical works (Ren et al., 2022b;a; Zhang et al., 2022; Shribak et al., 2024) have to abandon MLEs and resort to score matching or contrastive noise estimation to learn features and models, substantially deviating from the theory. We aim to bridge this gap by using squared families (Tsuchida et al., 2025) to model transition densities. Such families are expressive and tractable when restricted to their subclasses, the squared neural families (SNEFY). We theoretically analyse the regret bounds incurred from MLE errors and empirically design a representational RL algorithm that uses MLE to extract useful representations for fast downstream policy training.

Our specific contributions are,

- We propose a new general parameterised MDP class, namely the *Squared Family MDPs*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We design a theoretical algorithm, *Squared Family UCRL*, that uses MLE to estimate model parameters. We use a martingale approach to derive a MLE concentration result and use optimism in the face of uncertainties to analytically show a $\tilde{O}(dH\sqrt{T})$ regret bound.
- Empirically, we propose *Representation RL with the Squared Neural Families* (RESNEF) that keep MLE as the estimation method. The representation learnt from the density is used for downstream RL learning. RESNEF outperforms many state-of-the-art baselines across many continuous control problems. Code will be released upon acceptance.

2 Related Works

Sample Efficient RL. The sample efficiency of continuous RL under different forms of structural assumptions has been investigated recently. When both the transition kernel and the reward function can be represented linearly via a known feature, Jin et al. (2020) proposed a least squares type algorithm that achieves regret in $\tilde{O}\sqrt{d^3H^3T}$. Yang & Wang (2020) investigates the feature embedding setting, $P(s' \mid s, a) = \phi(s, a)^\top M\psi(s')$, where M is the parameter to be estimated and shows a $\tilde{O}(H^2d\sqrt{T})$ regret bound. Such linear forms can result in density models that are not necessarily non-negative and do not incorporate many useful settings. Chowdhury et al. (2021); Ouhamma et al. (2023) resolve the issue by generalising to exponential family cases, and they achieve a $\tilde{O}(d\sqrt{H^3T})$ regret bound. However, the log-partition functions of exponential families are hard to compute, hindering their practical benefits. Another line of work looks at more abstract complexity notions like Eluder dimensions (Jin et al., 2021) and bilinear classes (Du et al., 2021), but it is hard in practice to ensure the algorithm stays within the low complexity region.

Tractable Representational RL. Practical approaches that adopt the linear MDP framework need to learn the typically unknown feature that admits linear decompositions. Instead of using MLEs that are proven to be sample efficient (Agarwal et al., 2020; Uehara et al., 2021), they have to bypass computing normalisations by resolving to other techniques, such as Gaussian assumptions (Ren et al., 2022b), contrastive noise estimations (Zhang et al., 2022), spectral decompositions (Ren et al., 2022a), and diffusion score matching (Shribak et al., 2024). General representational RL has also been investigated by Barreto et al. (2017); Yarats et al. (2021); Stooke et al. (2021), but the features they leverage are principally different from our line of work with minimal theoretical understanding.

Unlike these methods, *Squared Family MDPs admits general density modelling, sharp theoretical bounds, and practical algorithms that closely adapts from the theory*.

3 Preliminaries

3.1 Markov Decision Processes (MDPs)

A finite horizon episodic Markov Decision Process (MDP) (Puterman, 2014) is defined by the tuple $\langle S, A, P, r, H, \gamma, \mu_0\rangle$. The state S and action A spaces are allowed to be continuous. In an episode k, ($1 \leq k \leq K \in \mathbb{N}$), a trajectory can be generated as $(s_0^k, a_0^k, r_0^k, \ldots, s_H^k, a_H^k, r_H^k, s_{H+1}^k)$. The subscript index denotes the time step within the episode that starts from $h = 1$ to $h = H$. The total step is denoted as $T = KH$. When the episode number does not matter, we simply drop the superscript for cleanliness. The agent always starts in an initial state s_0 drawn from the initial distribution μ_0. The probability of transit from state s_h^k to s_{h+1}^k after executing a_h^k is given by the transition kernel $P_{\theta^*}(s_{h+1}^k \mid s_h^k, a_h^k)$, where θ^* denotes the ground truth parameters. Actions are selected according to the policy $\pi = (\pi_1, \ldots, \pi_H)$, where $\pi_h : S \to A$ maps from the state space to the action space at step h. A reward is given according to a state and an action $r : S \times A \to [0, 1]$. An agent would like to find a policy π that maximises the value function $V_{\theta^*}^{\pi}(s_h^k) = \mathbb{E}_{\theta^*}\left[\sum_{j=h}^{H} r\big(s_j^k, \pi_j(s_j^k)\big)\right]$, which is defined as the expected total reward starting from state $s \in S$ at step $h \in [H]$. Similarly, we can also assign values to state-action pairs known as Q-values,

$$Q_{\theta^*}^{\pi}(s_h^k, a_h^k) = r(s_h^k, a_h^k) + \mathbb{E}_{\theta^*}\left[\sum_{j=h+1}^{H} r\big(s_j^k, \pi_j(s_j^k)\big)\right]. \tag{1}$$

The subscript θ^* denotes that the expectation is taken with respect to a true target transition P_{θ^*}. Let the policy class be denoted as Π, an optimal policy π^* maximises the value function, $\pi^\star = \arg\max_{\pi\in\Pi} V^\pi_{\theta^*}(s)$. In RL, the transition kernel and the reward function are unknown to the agent, and any strategy the agent deploys needs to carefully balance how much it exploits the best current estimate with how much it should explore and expose to new information. This trade-off contributes to regret, which is the difference between the value function under the optimal policy and the value function of the current policy deployed by the agent, $\text{Regret}(K) = \sum_{k=0}^{K} V^{\pi^*}_{\theta^*,0} - V^{\pi^k}_{\theta^*,0}$. When the transition kernel P can be decomposed as an inner product between two possibly non-linear features $\phi : S \times A \to \mathbb{R}^{\tilde{d}}$ and $\nu : S \to \mathbb{R}^{\tilde{d}}$,

$$P(s^k_{h+1} \mid s^k_h, a^k_h) = \langle \phi(s^k_h, a^k_h), \nu(s^k_{h+1}) \rangle_{\mathbb{R}^{\tilde{d}}} \tag{2}$$

we call it a *Linear* MDP (Agarwal et al., 2020; Jin et al., 2020; Yang & Wang, 2020). In such settings, the Q-values defined in Eq.1 can be written as,

$$Q^\pi(s^k_h, a^k_h) = r(s^k_h, a^k_h) + \Big\langle \phi(s^k_h, a^k_h), \underbrace{\int_S \nu(s^k_{h+1}) V^\pi(s^k_{h+1}) \mathrm{d}s^k_{h+1}}_{\omega_\pi} \Big\rangle_{\mathbb{R}^{\tilde{d}}}. \tag{3}$$

Importantly, Q-values can be linearly represented as an inner product between the feature ϕ and some learnable weights ω_π. Linear MDP may seem restrictive at first; however, it has been shown that many infinitely dimensional cases can be characterised under such settings; some examples are given in Appendix B.

3.2 Squared Families

Squared families (Tsuchida et al., 2025) with universal approximation properties are proposed as an alternative to exponential families.

Definition 3.1 (Squared family (Tsuchida et al., 2025))**.** *Let $(\mathbb{X}, \mathcal{F}, \mu)$ be a measure space, with $\mathbb{X} \subseteq \mathbb{R}^d$. Let $\psi(x_1, \cdot) : \mathbb{X} \to \mathbb{R}^n$ and $\theta \in \Theta \subseteq \mathbb{R}^n$, where x_1 is some conditioning variable. We say that*

$$\tilde{p}(x_2 \mid x_1, \theta) = \frac{(\theta^\top \psi(x_1, x_2))^2}{z_\theta(x_1)} \tag{4}$$

where $z_\theta(x_1) = \int_{\mathbb{X}} (\theta^\top \psi(x_1, x_2))^2 \mu(dx_2)$, belongs to a squared family. We call the collection $\{\tilde{p}(x_2 \mid x_1, \theta)\}_{\theta\in\Theta}$ a squared family, where $\Theta \subseteq \{\theta \in \mathbb{R}^n \mid \forall x_2 \in \mathbb{X}, 0 < z_\theta(x_1) < \infty\}$.

Its normalisation factorises into parameters and an integral

$$z_\theta(x_1) = \text{Tr}(\theta\theta^\top \underbrace{\int_{\mathbb{X}} \psi(x_1, x_2)\psi(x_1, x_2)^\top \mu(dx_2))}_{\mathcal{K}_{\mu,\psi}(x_1)},$$

where $\mathcal{K}_{\mu,\psi} \in \mathbb{R}^{n\times n}$ is called the squared family kernel. Since the parameters θ are separated from the kernel (which is often available in closed-form), it is possible to construct models with closed form normalisation terms. In particular, when setting $\psi(x_1, x_2) = \sigma(W_1 x_1 + W_2 x_2 + b)$, a one layer neural network with parameters (W_1, W_2, b) and activations $\sigma(\cdot)$, we recover the special **squared neural families** (SNEFY) (Tsuchida et al., 2023). With many different parameterisation choices, the squared family kernel is analytically tractable. Details of SNEFY are provided in Appendix A.

A useful divergence for squared families is the squared Hellinger distance. Given two densities p, q defined over a measurable space $\mathbb{X}$ with a measure μ, the squared Hellinger distance is,

$$\text{SH}(p : q) = \frac{1}{2} \int_{\mathbb{X}} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mu(\mathrm{d}x). \tag{5}$$

Furthermore, if the features ψ are rich enough, squared families satisfy the universal approximation properties for arbitrary distributions. It has many other statistical properties, we defer interested readers to Tsuchida et al. (2025).

4 Squared Families Parameterised Transition Dynamics

We now introduce the Squared Family MDPs and show that near-optimal policy can be obtained in a sample efficient manner following the principle of optimism in the face of uncertainty.

4.1 Definitions and Assumptions

Motivated by Definition 3.1, we define the following conditional squared families,

Definition 4.1 (Squared Family MDP). *An episodic MDP is said to be parameterised by a squared family if for a given $\epsilon \in (0,1)$, in any episode k, at any step h, for any states $s_h^k, s_{h+1}^k \in S$ and action $a_h^k \in A$, the transition kernel satisfies,*

$$^{\epsilon}P_\theta(s_{h+1}^k \mid s_h^k, a_h^k) = (1-\epsilon)\tilde{p}\left(s_{h+1}^k \mid s_h^k, a_h^k, \theta\right) + \epsilon \tag{6}$$

where $\tilde{p}(\cdot)$ is a squared density with respect to base measure μ,

$$\tilde{p}(s_{h+1}^k \mid s_h^k, a_h^k, \theta) = \frac{(\psi(s_h^k, a_h^k, s_{h+1}^k)^T\theta)^2}{z_\theta(s_h^k, a_h^k)},$$

*and the normalisation constant z satisfies $z_\theta(s_h^k, a_h^k) = \int(\psi(s_h^k, a_h^k, s_{h+1}^k)^T\theta)^2 \ \mu(\mathrm{d}s_{h+1}^k)$. Here $\psi : S \times A \times S \to \mathbb{R}^d$ is a **known** feature mapping, $\theta \in \{\theta \in \mathbb{R}^d \mid \forall s_h^k \in S a_h^k \in A, 0 < z_\theta(s_h^k, a_h^k) < \infty\}$ is the unknown parameter to be estimated.*

We assume that the reward function is known (Chowdhury et al., 2021) and the initial state s_0 is fixed (Agarwal et al., 2020; Jin et al., 2020; Du et al., 2021; Chowdhury et al., 2021), so that we can focus on learning the squared family density. The ϵ in Eq.6 serves as a smoothing factor between the squared density and the base measure μ. It provides absolute continuity with the base measure and imposes tail controls; however, it does not destroy any useful properties of squared families. Indeed, Proposition C.1 shows that such setting retains the universal approximation property of the original squared family at a point-wise level. The squared families kernel is $\mathcal{K}_{\psi,\mu}(s_h^k, a_h^k) = \int \psi(s_h^k, a_h^k, s')\psi(s_h^k, a_h^k, s')^T\mu(\mathrm{d}s')$. The local geometry that governs two densities with parameters θ, θ' is better captured by the **path averaged squared kernel** $K^{\epsilon}_{\psi,\mu}(s, a; \theta, \theta')$. Let $g(u) = \sqrt{(1-\epsilon)u^2 + \epsilon}$, so that $\sqrt{^{\epsilon}P_\theta(s_{h+1}^k \mid s_h^k, a_h^k)} = g(\psi(s_h^k, a_h^k, s_{h+1}^k)^\top\theta)$. Write $\theta_\tau = \theta' + \tau(\theta - \theta')$, $\tau \in [0,1]$, for the segment between them, and define $\iota_{\theta,\theta'}(s, a, s') := \int_0^1 g'\big(\psi(s, a, s')^\top\theta_\tau\big)\, d\tau$, then

$$K^{\epsilon}_{\psi,\mu}(s, a; \theta, \theta') := \int \iota^2_{\theta,\theta'}(s, a, s')\, \psi(s, a, s')\psi(s, a, s')^\top\, \mu(ds').$$

The feature ψ itself can take many different forms. Some examples include,

Feature Embedding Setting: The feature embedding setting (Yang & Wang, 2020; Chowdhury et al., 2021; Ouhamma et al., 2023) considers the following specific $\psi(\cdot)$ structure,

$$\psi(s_h^k, a_h^k, s_{h+1}^k)_i = (y(s_h^k, a_h^k)^\top\xi(s_{h+1}^k))_{i \le d}, \tag{7}$$

for some given feature maps y, ξ.

Random Features: ψ can be composed of multiple random features and θ acts as coefficients that linearly combine these features together. A well-known instance of random features are random Fourier features (Rahimi & Recht, 2007), which are used to approximate kernel methods with stationary kernels (Bochner, 1948). Our empirical section also links back to these examples.

We are interested to learn the parameters θ of the squared families given the trajectories generated from the episodic MDP. This can be done via the following MLE objective,

$$\hat{\theta}_K = \mathrm{argmin}_{\theta\in\Omega} - \sum_{k=1}^{K}\sum_{h=1}^{H} \log\Big(^{\epsilon}P_\theta(s_{h+1}^k \mid s_h^k, a_h^k)\Big). \tag{8}$$

Here $\Omega = \{\theta \in \mathbb{R}^d \mid \theta_1 \ge \eta > 0, ||\theta||_2 \le \mathcal{R}, z(\theta, s_h^k, a_h^k) = 1, \forall s_h^k, a_h^k \in S \times A\}$ is the normalised parameter set with radius $\mathcal{R} > 0$. This set Ω is non-trivial, for instance, when $\mathcal{K}_{\psi,\mu}(s, a) = \mathcal{K}_0 \succ 0$, then the parameter set becomes an ellipsoid. In particular, consider the feature embedding example in Eq. 7. If one picks ξ to satisfy $\int \xi(s)\xi(s)^T\mathrm{d}s = I$ and sets $y : S \times A \to O(d)$ to output an orthogonal matrix, then $K_{\psi,\mu} = I$, providing us with a non-trivial parameter set.

We consider the realisable setting, that is, $\theta^* \in \Omega$. The following assumptions are needed for the rest of theoretical analysis, but they are dropped in practice.

Algorithm 1 Squared Family UCRL

Require: Features $\phi(s, a, s')$, Reward $R(s, a)$, $\hat{\theta}_0$ Parameters $\epsilon, \delta, \alpha, K$
1: Initialise $D_0 = \emptyset$.
2: **for** $k = 1, \ldots, K$ **do**
3: Solve $\hat{\theta}_k = \text{argmax}_{\theta \in \Omega} \sum_{i<k} \sum_{h=1}^{H} \log {}^{\epsilon}P_{\theta}(s_{h+1}^i \mid s_h^i, a_h^i)$.
4: Set $C_k = \left\{\theta \in \Omega : \sum_{i<k} \sum_{h=1}^{H} \text{SH}(P_{\hat{\theta}_k,h}^i : P_{\theta,h}^i) \leq \beta_{k-1}\right\}$.
5: Solve $\pi_k = \arg\max_{\theta \in C_k, \pi \in \Pi} V_{\theta}^{\pi}(s_0)$.
6: **for** $h = 1, \ldots, H$ **do**
7: Execute $a_h^k \sim \pi_k(s_h^k)$.
8: Observe s_{h+1}^k.
9: Set $D_k = D_{k-1} \cup \{s_h^k, a_h^k, s_{h+1}^k\}$.
10: **end for**
11: **end for**

Assumption 4.2. *For any s, a, s', the feature is bounded above $||\psi(s, a, s')||_2 \leq B < \infty$.*

Assumption 4.3. *The squared family kernel is strictly positive definite (PD), $\mathcal{K}_{\psi,\mu}(s, a) \succ 0$.*

Assumption 4.4. *There exists a constant $c_\epsilon > 0$ such that $c_\epsilon \mathcal{K}_{\psi,\mu}(s, a) \preceq \mathcal{K}^{\epsilon}_{\psi,\mu}(s, a; \theta, \theta')$ for all $s \times a \in S \times A$ and $\theta, \theta' \in \Omega$.*

The first two assumptions are standard, and the third assumption is a common uniform lower bound on the local geometry (Filippi et al., 2010; Li et al., 2017; Chowdhury et al., 2021). See Appendix. D for more discussions and examples of why the third assumption is valid.

4.2 Regret Bounds for Squared Family RL

We apply the optimality in the face of uncertainty principle (Auer et al., 2008) to the Squared Family MDP setting. We aim to use MLE to estimate θ^* from the data collected after $T = KH$ total steps of interactions with the transition kernel, and the concentration phenomenon of MLE in turn allows us to design a confidence set that contains θ^* with high probabilities. By iteratively computing the confidence set and solving for the best policy based on the confidence set as shown in Algorithm 1, we eventually show that the generated policies produce regret that grows sub-linearly in time T.

Let Θ_K denote the confidence set after K episodes and $\lambda_{\max}$ denote the maximum eigenvalue of $\mathcal{K}_h^k$ across all k and h. We also discretise the entire parameter set Ω with an α-cover and use $|\Omega_\alpha|$ to denote the cover number. We define ${}^{\epsilon}P_{\hat{\theta},h}^k := {}^{\epsilon}P_{\hat{\theta}}(s_{h+1}^k \mid s_h^k, a_h^k)$ and also for θ^* and θ. The following theorem provides the concentration result for squared families.

Theorem 4.5 (Confidence Set Construction)**.** *For any $0 < \delta \leq 1$ and $s_h^k, a_h^k \in S \times A$, define the confidence set C_K as*

$$C_K := \left\{ \theta \in \Omega \mid \sum_{k=1}^{K} \sum_{h=1}^{H} \text{SH}({}^{\epsilon}P_{\hat{\theta},h}^k; {}^{\epsilon}P_{\theta,h}^k) \leq \beta_K(\Omega, \delta, \alpha, \epsilon) \right\}, \tag{9}$$

where $\beta_K(\Omega, \delta, \alpha, \epsilon) = 2 \log \frac{|\Omega_\alpha|}{\delta} + \frac{2(1-\varepsilon)B^2 R}{\varepsilon} + \frac{(1-\varepsilon)\lambda_{\max}}{KH}$, then $\mathbb{P}\left(\theta^* \in \bigcap_{K=1}^{\infty} C_K\right) \geq 1 - \delta$.

The proof is given in Appendix C. Central to the proof is the construction of a squared family martingale from two random variables, the one-step log-likelihood ratio $L_h^k(\theta) := \frac{{}^{\epsilon}P_{\theta,h}^k}{{}^{\epsilon}P_{\theta^*,h}^k}$ and the Squared Hellinger affinity $A_h^k(\theta) = 1 - SH({}^{\epsilon}P_{\theta^*,h}^k : {}^{\epsilon}P_{\theta,h}^k)$. Their ratio forms the random variable $Z_h^k(\theta) := \frac{\sqrt{L_h^k}}{A_h^k}$ that measures the square-root of the log-likelihood around its mean. This quantity cannot grow too fast, and it controls the averaged squared-root log-likelihood ratios. From this property, we show that $\prod_k^K \prod_h^H Z_h^k(\theta)$ is a martingale. It is then controlled by standard Ville's inequality and a discretisation argument. The result of Theorem 4.5 is expected, as the confidence radius β_K is mainly controlled by the feature dimension d due to the term $|\Omega_\alpha|$ and the total time step $T = KH$.

Algorithm 2 RESNEF

Require: Network parameters $\theta_1, \theta_2, \mathbf{W}_q, W_r$. Policy π. Global parameters γ, K, H and $\mathcal{D} = \emptyset$.

1: **for** k = 1, . . . , K **do**
2: **for** h = 1, . . . , H **do**
3: Sample action, $a_h^k \sim \pi(\cdot \mid s_h^k)$
4: Simulate, $r_h^k = r(s_h^k, a_h^k), s_{h+1}^k \sim P(\cdot \mid s_h^k, a_h^k)$
5: Update $\mathcal{D}_h^k \to \mathcal{D}_{h-1}^k \cup \{s_h^k, a_h^k, r_h^k, s_{h+1}^k\}$
6: Learn θ_1, θ_2 via MLE $\arg\max_{\theta_1,\theta_2} \mathbb{E}_{\mathcal{D}_h^k}\left[\ln \phi_{\theta_1}(s_h^k, a_h^k)^\top \nu_{\theta_2}(s_{h+1}^k)\right]$
7: Learn θ_1, W_r via reward MSE, $\arg\min_{\theta_1,W_r} \mathbb{E}_{\mathcal{D}_h^k}\left[||\langle \text{vec}(Y_{\theta_1}(s_h^k, a_h^k)), W_r\rangle - r||^2\right]$
8: Update the critic $\omega, \bar{\omega}$ by Eq.12
9: Update the policy π by $\max_\pi \mathbb{E}_{\mathcal{D}_k^h, a_h^k \sim \pi}[\min\{Q_\omega(s_h^k, a_h^k), Q_{\bar{\omega}}(s_h^k, a_h^k)\}]$
10: Update the squared kernel $\mathcal{K}_{\xi_{\theta_2},\mu}$ (e.g. Eq 13).
11: **end for**
12: **end for**
13: **Return** π

Based on Theorem 4.5, we present Squared Family UCRL (Algorithm 1) that maintains the best current estimation of the parameter $\hat{\theta}_k$ and use it to construct the confidence set. Optimism in the face of uncertainties is carried out in line 5 to solve for the best possible policy π_k by finding the current optimal parameter. The next theorem provides the regret bound for Algorithm 1.

Theorem 4.6 (Regret bound.)**.** *Under the aforementioned assumptions and setups. For any* $0 < \delta \leq 1$*, Algorithm 1 achieves with probability at least* $1 - \delta$*,*

$$\text{Regret}(K) \leq 4\sqrt{KH^3(1-\epsilon)d\log(1+T)\left(\frac{8\beta_K}{c_\epsilon} + 4H\lambda_{\max}\mathcal{R}^2\right)} + 3H\sqrt{T\log\frac{2}{\delta}}.$$

Expanding β_K and ignores constants and logarithms, the regret bound is $\tilde{O}(dH\sqrt{T})$. Details of the proof of Theorem 4.6 are given in Appendix C. In the proof, we follow the simulation lemma (Kakade et al., 2020) approach by showing that the per episode regret is bounded by the estimation errors from MLE. Specifically, it can be captured by squared Hellinger distances between the two distributions. The sum of squared Hellinger distances is controlled by the squared kernels, which is controlled by our carefully designed elliptical potential analysis. Since H is fixed for each problem, the regret grows sub-linearly in terms of the total time T. The result is very close to that of exponential family MDPs for which a regret bound of $O(dH^{3/2}\sqrt{T})$ is proven by Chowdhury et al. (2021). See Appendix D for more involved comparisons of the two theoretical results.

5 Tractable Representation RL with Squared Families

The sharp regret bound of Algorithm 1 calls for practical methods, however, we face those major practical difficulties, unknown feature map ψ, intractable confidence set C_K, computing MLE in line 4 and optimising the policy in line 5. Fortunately, these difficulties can be resolved by parameterising the transition kernel with a subclass of the squared family, namely, the Squared Neural Family (SNEFY). In such setting, we obtain tractable normalisation and hence MLE. We use MLE to learn the unknown feature map and show that these learnt features can be used to directly learn the Q-values and hence solve for the optimal policy.

5.1 SNEFY Parameterised MDPs

We take a forward view to derive the SNEFY parameterisation from the theory. Directly learning $\psi(s_h^k, a_h^k, s_{h+1}^k)$ as in Definition 4.1 is problematic, as the feature itself is highly non-linear and does not admit any factorisations that are useful for downstream RL. Instead, we take the feature embedding setting from (7), and decompose ψ into two additional features y and ξ. We generalise y to output matrices instead of vectors and use $Y_{\theta_1} : S \times A \to \mathbb{R}^{m\times n}$ to denote the matrix form with learnable parameters θ_1. Indicated by the theory, the feature Y_{θ_1} can be arbitrary non-linear functions and to stick to SNEFY definition (see Section 3.2), we set $\xi_{\theta_2}(s_{h+1}^k) = \sigma(Wt(s_{h+1}^k) + b)$ for some

activation function σ, sufficient statistics $t : S \to \mathbb{R}^d$, weights $W \in \mathbb{R}^{n\times d}$ and bias term $b \in \mathbb{R}^n$. We use $\theta_2 = (W, b)$ to denote the learnable parameters for ξ_{θ_2} and abuse $\theta := (\theta_1, \theta_2)$ to denote the total parameters.

Definition 5.1. *Let μ denote the base probability measure and $\bar{\epsilon} \in \mathbb{R}$ be a learnable mixture. An MDP is said to be parameterised by the Squared Neural Family if for any $s_h^k, a_h^k, s_{h+1}^k \in S \times A \times S$, the transition kernel satisfies,*

$$P(s_{h+1}^k \mid s_h^k, a_h^k) = \frac{\mathrm{Tr}(Y_{\theta_1}(s_h^k, a_h^k)^\top Y_{\theta_1}(s_h^k, a_h^k)\tilde{\mathcal{K}}_{\xi_{\theta_2}}(s_{h+1}^k)) + \bar{\epsilon}^2}{z_\theta(s_h^k, a_h^k) + \bar{\epsilon}^2}\mu(\mathrm{d}s_{h+1}^k). \tag{10}$$

And the reward function satisfies,

$$R(s_h^k, a_h^k) = \langle \mathrm{vec}(Y_{\theta_1}(s_h^k, a_h^k)), W_r\rangle.$$

Here $\tilde{\mathcal{K}}_{\xi_{\theta_2}}(s_{h+1}^k) = \xi_{\theta_2}(s_{h+1}^k)\xi_{\theta_2}(s_{h+1}^k)^\top \in \mathbb{R}^{n\times n}$ is a low rank PSD matrix whose parameter is $\theta_2 \in \mathbb{R}^{n\times(d+1)}$, $\mathcal{K}_{\xi_{\theta_2},\mu}$ is the squared family kernel, μ is the base measure, $z_\theta(s_h^k, a_h^k) = \mathrm{Tr}(Y_{\theta_1}(s_h^k, a_h^k)^\top Y_{\theta_1}(s_h^k, a_h^k)\mathcal{K}_{\xi_{\theta_2},\mu})$ is the normalisation term and $W_r \in \mathbb{R}^{nm}$ are the learnable parameters for the reward function.

This definition is motivated by the fact that squaring the 2-norm of a linear statistic is equivalent to computing the trace of the linear statistic in its matrix form. The learnable mixture $\bar{\epsilon}$ here serves similar roles as the ϵ mixture from Definition 4.1. Its square ensures that the SNEFY density does not collapse to zero during learning. Indeed, by defining $\lambda(s, a) = \frac{\bar{\epsilon}^2}{Z(s,a)+\bar{\epsilon}^2}$, we can re-write the density in Eq. 10 in a similar form to our theory given in Definition 4.1,

$$P(s_{h+1}^k \mid s_h^k, a_h^k) = (1 - \lambda(s_h^k, a_h^k))\frac{\mathrm{Tr}(Y_{\theta_1}(s_h^k, a_h^k)^\top Y_{\theta_1}(s_h^k, a_h^k)\tilde{\mathcal{K}}_\xi(s_{h+1}^k))}{\mathrm{Tr}(Y_{\theta_1}(s_h^k, a_h^k)^\top Y_{\theta_1}(s_h^k, a_h^k)\mathcal{K}_{\xi,\mu})} + \lambda(s_h^k, a_h^k).$$

The key observation here is that the λ mixture, compared to ϵ from Definition 4.1, is more general and depends on the actual state and action pairs and also learnable.

5.2 Representation RL with Squared Neural Families (RESNEF)

Definition 5.1 admits linear decomposition and motivates RESNEF shown in Algorithm 2.

Proposition 5.2. *Under the SNEFY parameterisation in Definition 5.1, define*

$$\phi_{\theta_1}(s, a) := \frac{1}{z_\theta(s, a) + \bar{\epsilon}^2}\begin{bmatrix}\mathrm{vec}\left(Y_{\theta_1}(s, a)^\top Y_{\theta_1}(s, a)\right)\\ 1\end{bmatrix}, \nu_{\theta_2}(s') := \begin{bmatrix}\mathrm{vec}\left(\widetilde{K}_{\xi_{\theta_2}}(s')\right)\\ \bar{\epsilon}^2\end{bmatrix}\mu(ds').$$

Then, for any s_h^k, a_h^k, s_{h+1}^k, we have $P(s_{h+1}^k \mid s_h^k, a_h^k) = \langle\phi_{\theta_1}(s_h^k, a_h^k), \nu_{\theta_2}(s_{h+1}^k)\rangle$.

The proof simply follows from the properties of trace and basic algebra. The linear decomposition required in Eq. 2 is obtained through the trace of the linear statistics in their matrix forms. This contrasts with modelling the transition kernel as energy based models, which obtain their linear decomposition through spectral analysis (Ren et al., 2022a; Shribak et al., 2024). Since the squared kernel is tractable (see Appendix A.1), both features ϕ and ν can be computed and learnt. By substituting the result into Eq.3, and for any s_h^k, a_h^k, the corresponding Q-values of the MDP is

$$Q^\pi(s_h^k, a_h^k) = \langle \mathrm{vec}(Y_{\theta_1}(s_h^k, a_h^k)), W_r\rangle + \left\langle \phi_{\theta_1}(s_h^k, a_h^k), \int_S \nu_{\theta_2}(s_{h+1}^k)V^\pi(s_{h+1}^k)\mu(\mathrm{d}s_{h+1}^k)\right\rangle. \tag{11}$$

The integral is learnable and serves as a key parameter for the Q-value. Directly working with ϕ_{θ_1} could be problematic. In typical RL, the dimension n is often set to 256 or 512, but $\mathrm{vec}(Y^\top Y)$ has dimension $n \times n$. This issue is resolved by noticing that the entire Q-value ultimately depends on the feature Y_{θ_1}, as the first term in Eq.11 is linear in terms of Y_{θ_1} and the feature ϕ_{θ_1} is a non-linear transformation of Y_{θ_1}. Therefore, we parameterise the Q-value with two weights $W_q \in \mathbb{R}^{nm\times p}, W_q' \in \mathbb{R}^{p\times 1}$. We can hence pick m to be small so Y_{θ_1} to make the internal dimensions low. Define $\mathbf{W}_q = (W_Q, W_Q')$,

$$Q_{\mathbf{W}_q}^\pi(s_h^k, a_h^k) = \langle \mathrm{vec}(Y_{\theta_1}(s_h^k, a_h^k)), W_r\rangle + \tilde{\sigma}\left(\mathrm{vec}(Y_{\theta_1}(s_h^k, a_h^k))^\top W_q\right)W_q'.$$

Here $\tilde{\sigma}$ is an arbitrary activation function. The Q values and hence the actor can be learnt through an actor and critic fashion. The entire algorithm is shown in Algorithm 2. At the beginning of each episode, we use MLE to learn the parameters θ for the SNEFY model and the mean squared error (MSE) to learn a linear layer on top of Y_{θ_1} to predict the reward. Then the standard temporal difference error (Sutton et al., 1998) is used with the double Q-network trick (Fujimoto et al., 2018) to learn the Q values in line 8,

$$l_{\text{critic}}(\mathbf{W}_q) = \mathbb{E}_{s_h^k, a_h^k, r_h^k, s_{h+1}^k \sim D}\left[(r + \gamma \mathbb{E}_{a' \sim \pi}[Q_{\bar{\mathbf{W}}_q}(s_h^k, a')] - Q_{\mathbf{W}_q}(s_h^k, a_h^k))^2\right] \tag{12}$$

where $\bar{\mathbf{W}}_q$ is the target network with delayed parameter updates and D is the replay buffer. The actor can be learnt by maximising the critic predicted performances in line 9. And in the end of each update, we need to update $\mathcal{K}$, since it depends on ξ_{θ_2} whose parameters are updated in each gradient step.

RESNEF with Random Features. RESNEF needs to estimate the square kernel in Line 10 whenever θ_2 is updated, but the computational cost for $\mathcal{K}$ is $O(m^2 n + dn^2)$ which incurs a long training time. Observing that the entire Q-value in Eq. 11 depends only on Y_{θ_1} but the kernel $\mathcal{K}_{\xi,\mu}$ depends only on θ_2, we propose a random feature variant, called RESNEF-RF, which randomly initialises parameters θ_2 and freezes them during training. Then, we only need to learn θ_1 via MLE with a pre-computed fixed kernel, substantially reducing computational cost during training.

6 Experiments

Figure 1: Training curves of Gym-Mujoco tasks. X-axis is the training steps and Y-axis is the averaged evaluation rewards across 4 random seeds. Solid lines represent average performances and shaded areas represent standard errors.

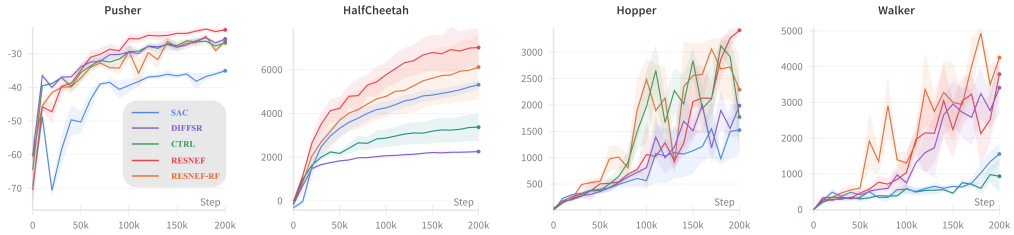


Table 1: Evaluation results on Gym-Mujoco tasks. Average returns over four seeds are reported with standard errors in parentheses. Red indicates the best and blue indicates the second best.

	HalfCheetah	InvDoublePend.	Hopper	Reacher	Walker	Pusher	Humanoid	Ant
SAC	5315 (230)	9359 (1.24)	1499 (468)	-5.92 (0.80)	1556 (293)	-35.1 (0.63)	2434 (391)	1737 (178)
DIFFSR	2255 (50.5)	9349 (5.63)	1987 (580)	-3.28 (0.352)	3148 (835)	-25.6 (0.66)	532 (66)	6240 (133)
CTRL	3372 (810)	9326 (3.67)	1771 (738)	-3.19 (0.348)	932 (460)	-26.8 (0.645)	335 (40.5)	169 (48)
RESNEF-RF	6122 (723)	9356 (1.20)	2294 (417)	-3.22 (0.345)	4254 (134)	-26.3 (0.62)	1733 (608)	3913 (751)
RESNEF	7019 (871)	9357 (3.08)	3417 (99)	-3.54 (0.354)	3780 (553)	-22.9 (0.492)	533 (37)	2867 (948)

Table 2: Evaluation results on DMC tasks. Red indicates the best and blue indicates the second best.

	Reacher-Easy	Cheetah-Run	Fish-Swim	Walker-Run	Hopper-Stand
SAC	976 (1.5)	615 (6.1)	214 (21)	632 (19)	308 (7)
CTRL	944 (27)	697 (14)	115 (19)	724 (6)	898 (15)
DIFFSR	945 (31)	628 (46)	314 (26)	699 (11)	650 (62)
RESNEF-RF	975 (4.0)	750 (26)	372 (38)	706 (16)	868 (32)
RESNEF	955 (11)	744 (21)	379 (12)	725 (10)	303 (101)

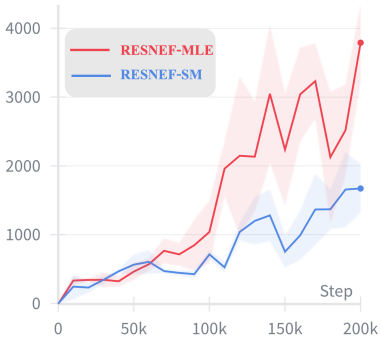
Setup. We conduct experiments across continuous control problems from Gym-Mujoco (Todorov et al., 2012; Towers et al., 2024) and DeepMind Control Suites (DMC) (Tassa et al., 2018) and

compare our methods with strong baselines, SAC (Haarnoja et al., 2018), CTRL (Zhang et al., 2022) and DIFFSR (Shribak et al., 2024). We drop some old representation approaches like Barreto et al. (2017) and Ren et al. (2022b) as they have been shown to be less competitive than existing baselines here. The latter two are recent representation RL methods that also leverage linear MDP formulation but relies on estimation methods other than MLE to learn the linear features. All methods are run with 200k steps with 4 random seeds.

For the feature ξ, we use Gaussian mixtures with 8 mixing components as base measure μ, Snake$_a$ activations for σ, and the identity function as sufficient statistics. These choices are not arbitrary, as they ensure flexible models with tractable squared kernel and hence normalisations (see Appendix A.1). We set $n = 512$ for all experiments, $m = 4$ for Gymnasium tasks and $m = 8$ for DMC tasks. For Y_{θ_1}, we use residual MLP blocks with one hidden layer. More details can be found in Appendix E.

Results. The performance curves depicted in Figure 1 show the distinctive performance of our methods during training. As RESNEF and RESNEF-RF gradually found better representations, they are able to surpass other methods early in training. We believe that the representation power of RESNEF comes from lifting the linear statistics to its matrix form in Definition 5.1. Extra ranks are added to the features by setting $m > 1$, enriching their information content and hence RL conditioned on these features can learn faster online. The final results are reported in Table 1 and Table 2. As seen, RESNEF and RESNEF-RF achieves the best performance across many domains. While RESNEF achieves 40% gains across hopper and half-cheetah domains, RESNEF-RF performs better in harder control tasks like walker, humanoid and ant. Further, by focusing on learning the parameter θ_1 while keeping θ_2 fixed, RESNEF-RF is able to learn more sophisticated representations, e.g. the humanoid task from Gymnasium. It also performs consistently well in DMC tasks. The good performance of RESNEF-RF shows that our underlying method is more about learning good representations and less about learning accurate models. The wall-clock training times of all methods are given in Fig. 3, RESNEF-RF successfully saves 34% of training time from RESNEF. Additional experimental details are given in Appendix. E.

Figure 2: MLE vs Score Matching (SM) for RESNEF in Walker.



Ablations To isolate the effect of MLE in RESNEF, we also implement a score matching ablation of RESNEF. This ablation (RESNEF-SM) keeps everything the same as RESNEF except that the model density is learnt via score matching. The training plot for the walker environment shown in Fig. 2 suggests that MLE is much more suited for squared families, empirically strengthening our theory (more ablation results for different environments are given in Fig. 6). We also study the importance of feature dimensions and report the results in Appendix. E.3. We found that higher n (e.g 512) are generally good for complex tasks, whereas keeping m to be at least 4 ensures the underlying SNEFY model does not collapse into a mixture model.

7 Discussions

Limitations Although RESNEF is inspired by our theory, there is still a large gap between our theory and practice. Closing these gaps, such as incorporating learnable representations into the theory, are worthy of further studies. Furthermore, RESNEF obtains a linear decomposition via tensorisation of vectors to matrices. As a result, the dimensionality of the matrix incurs substantial computational costs for processing high dimensional inputs like images. We leave the design of more scalable RL algorithms that leverage SNEFY to future works.

Conclusions We proposed the squared family MDP and derived a $\tilde{O}(dH\sqrt{T})$ regret bound using a martingale control and simulation lemma approach adapted to the squared family setting. Our practical algorithm RESNEF adapted from the theory keeps MLE as the estimation approach and demonstrates strong performance in multiple continuous control tasks compared to other strong baselines.

AI USE STATEMENT

In this work, we used generative AI tools for coding the RESNEF algorithm and assisting the writing of proofs found in Appendix C. We have not used generative AI tools for coming up with the fundamental ideas, including the definitions of Squared Family MDPs and its SNEFY subclass, designing the RESNEF algorithm, obtaining the theoretical and experimental results, assisting with translation and interpreting results. The rest of the required disclosure tasks, such as dataset reformatting and translations are not applicable to this work. Additionally, we used generative AI tools for improving the readability of the paper. We have reviewed all AI-assisted work. We have carefully read through all the code AI produces and ensures that the code matches our intentions. We have checked all the experimental outcomes and convince ourselves their validity. The main structure of the mathematical proofs are laid down by us authors and AI is used to check the rigour and identify potential loose ends. We have checked the maths produced by AI before integrating them into our proofs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The following practices ensure the reproducibility of this work. The theoretical setups and assumptions are clearly discussed in the main text, and all proofs are detailed in the appendix. C with all steps spelt out. We have tried to make the proof intuitive by adding intuitive explanations. The theoretical sections have been proof read multiple times with AI assistance to ensure coherence and formality. Our experiments are carried out on standard control benchmarks and we use the latest update of the software. For all baselines involved, we use their official open source repositories instead of implementing our own. All training and results are carried across 4 random seeds to obtain meaningful and interpretable statistical results. Comprehensive experiments are carried out to understand the capability of our practical algorithms and also to reflect on our theory. Our code will also be made public upon acceptance to facilitate further research.

REFERENCES

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Salomon Bochner. Vorlesungen über fouriersche integrale. *(No Title)*, 1948.

Sayak Ray Chowdhury, Aditya Gopalan, and Odalric-Ambrym Maillard. Reinforcement learning in parametric mdps with exponential families. In *International Conference on Artificial Intelligence and Statistics*, pp. 1855–1863. PMLR, 2021.

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pp. 2826–2836. PMLR, 2021.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. *Advances in neural information processing systems*, 23, 2010.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Jiacheng Guo, Zihao Li, Huazheng Wang, Mengdi Wang, Zhuoran Yang, and Xuezhou Zhang. Provably efficient representation learning with tractable planning in low-rank pomdp. In *International Conference on Machine Learning*, pp. 11967–11997. PMLR, 2023.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17:257–317, 2020. doi: 10.1214/18-PS321.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pp. 2137–2143. PMLR, 2020.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems*, 34:13406–13418, 2021.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pp. 2071–2080. PMLR, 2017.

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. *Advances in neural information processing systems*, 27, 2014.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

Reda Ouhamma, Debabrota Basu, and Odalric Maillard. Bilinear exponential family of mdps: frequentist regret bound with tractable exploration & planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9336–9344, 2023.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Tongzheng Ren, Tianjun Zhang, Lisa Lee, Joseph E Gonzalez, Dale Schuurmans, and Bo Dai. Spectral decomposition representation for reinforcement learning. *arXiv preprint arXiv:2208.09515*, 2022a.

Tongzheng Ren, Tianjun Zhang, Csaba Szepesvári, and Bo Dai. A free lunch from the noise: Provable and practical exploration for representation learning. In *Uncertainty in Artificial Intelligence*, pp. 1686–1696. PMLR, 2022b.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

Igal Sason and Sergio Verdú. *f-divergence inequalities*. IEEE Transactions on Information Theory, 2016.

Dmitry Shribak, Chen-Xiao Gao, Yitong Li, Chenjun Xiao, and Bo Dai. Diffusion spectral representation for reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 110028–110056, 2024.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pp. 9870–9879. PMLR, 2021.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Russell Tsuchida, Cheng Soon Ong, and Dino Sejdinovic. Squared neural families: A new class of tractable density models. *Advances in neural information processing systems*, 36:73943–73968, 2023.

Russell Tsuchida, Jiawei Liu, Cheng Soon Ong, and Dino Sejdinovic. Squared families: Searching beyond regular probability models. *arXiv preprint arXiv:2503.21128*, 2025.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

Jean Ville. *Étude critique de la notion de collectif*. Gauthier-Villars, 1939.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019.

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, and Bo Dai. Making linear mdps practical via contrastive representation learning. In *International Conference on Machine Learning*, pp. 26447–26466. PMLR, 2022.

Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. *Advances in Neural Information Processing Systems*, 33:1583–1594, 2020.

A Details of Squared Neural Families

A squared neural family (SNEFY) (Tsuchida et al., 2023) is obtained by squaring the 2-norm of a neural network and normalising it with respect to a base measure. Given a measure space $(\Omega, \mathcal{F}, \mu)$ with $\Omega \subseteq \mathbb{R}^d$. Let $Y \in \mathbb{R}^{m\times n}$ be the readout parameter of a neural network and $W \in \mathbb{R}^{n\times D}$ and $b \in \mathbb{R}^n$ be the weights and biases of a hidden layer of a neural network $f : \mathbb{R}^D \to \mathbb{R}^m$ with activation $\sigma(\cdot)$, then the squared neural family parameterised by Y, W and b over the support $X \subseteq \Omega$ is given as,

$$P(\mathrm{d}x; Y, W, b) := \frac{\mu(\mathrm{d}x)}{Z(Y, W, b)} ||f(t(x); Y, W, b)||_2^2,\ f(t; Y, W, b) = V\sigma(Wt + b)$$

and the normalisation constant is finite and non-zero,

$$z(Y, W, b) := \int_X ||f(t(x); Y, W, b)||_2^2 \mu(\mathrm{d}x).$$

As discussed in the main text, one can easily see that SNEFY is just a special case of the squared family. The original SNEFY is proposed without conditioning variables. In our work, since the next state transition density always depends on the current state and action, we choose to re-write the SNEFY density to include these conditionals. Specifically by noticing that Y can be generalised to a arbitrary neural network instead of a simple readout parameter, we choose to make $Y : S \times A \to \mathbb{R}^{m\times n}$ to absorb the conditionals. The rest of the properties of SNEFY still remains. In particular, the normalisation term is tractable to compute under many choices of activation, base measure and sufficient statistics, which we provide details next.

A.1 Tractable Squared Kernels

In this section, we brief some important squared kernels used in the paper for tractable MLE computations. Given a SNEFY distribution (Section 3.2) with identity sufficient statistics such that $\xi(s') = \sigma(Ws' + b)$. When the base measure is chosen as an isotropic Gaussian $\mu_\Phi(\cdot)$ and $\sigma(\cdot)$ is the identity function, we obtain

$$\kappa_{\xi,\mu_\Phi}(w_i, b_i, w_j, b_j) = w_i^\top w_j + b_i b_j.$$

When setting $\sigma(\cdot)$ to be the cosine activation functions, then we have the kernel,

$$\kappa_{\xi,\mu_\Phi}(w_i, b_i, w_j, b_j) = \frac{\cos|b_i - b_j|}{2} \exp\left(\frac{-||w_i - w_j||^2}{2}\right) + \frac{\cos|b_i + b_j|}{2} \exp\left(\frac{-||w_i + w_j||^2}{2}\right)$$

where κ above is the ij-th entry of $\mathcal{K}$ and w_i, b_i denotes the i-th row of weights W and bias b. If the activation function is the $\mathrm{Snake}_{a-\frac{1}{2a}}$ activation (Ziyin et al., 2020), then

$$\begin{aligned}
&\kappa_{\mathrm{Snake}_{a-\frac{1}{2a}},\mathrm{Id},\mu_\Phi}(w_i, b_i, w_j, b_j) = \\
&\frac{1}{4a^2}\kappa_{\xi,\mu_\Phi}(2a(w_i, b_i), 2a(w_j, b_j)) + w_j^\top w_j \left(\sin(2ab_j)\exp\{-2a^2\|w_j\|^2\} + \sin(2ab_i)\exp\{-2a^2\|w_i\|^2\}\right) \\
&\quad - \frac{b_i}{2a}\cos(2ab_j)\exp\{-2a^2\|w_j\|^2\} - \frac{b_j}{2a}\cos(2ab_i)\exp\{-2a^2\|w_i\|^2\} + \kappa_{Id,Id,\mu_\Phi}(w_i, b_i, w_j, b_j).
\end{aligned} \tag{13}$$

Full derivations of these kernels, as well as other settings allowing for ReLU, GELU, LReLU, exponential and (sigmoidal) error function activations are given in Tsuchida et al. (2023).

A.2 Connections to Energy-based Models (EBMs)

Without further constraints, densities of the form $P(s' \mid s, a) = \langle \phi(s, a), \nu(s') \rangle$ is not guaranteed to be non-negative, while not being rich enough to capture many non-linear densities (Chowdhury et al., 2021). To resolve the non-negative density issue, one can take the exponential of the density or square the density. The former results in the energy based models that have been studied by many recent works (Chowdhury et al., 2021; Ouhamma et al., 2023; Shribak et al., 2024). However, they are either theoretical or have to resolve to score matching to deal with parameter estimations. The latter gives the squared families, its subclass SNEFY permits tractable normalisations and hence MLE.

Here we show that SNEFY can be parameterised as a mixture of EBMs. For a SNEFY, consider using an activation function $\sigma(x) = \exp(x/2)$ and set $Y(s,a)^\top Y(s,a) = \exp f(s,a)\tilde{Y}^\top\tilde{Y}$ for some arbitrary function $f(s,a) : S \times A \to \mathbb{R}^n$ and matrices $\tilde{Y}$, then we obtain

$$P(s' \mid s,a) \propto \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{y}_{\cdot,i}^\top \tilde{y}_{\cdot,j} \exp\left(\frac{1}{2}(w_i + w_j)^\top t(s') + f(s,a)\right)\nu(\mathrm{d}s').$$

This density can be viewed as a mixture of EBMs.

B More about Linear MDPs

We provide more backgrounds of linear MDPs in this section and emphasis its role in this work. The linear decomposition in Eq. 2 may appear restrictive, however many works have shown that non-linear stochastic systems admits linear decompositions via kernel embeddings or spectral representations. A good example is given by Ren et al. (2022b). Suppose the following system dynamics,

$$s' = f(s,a) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$$

for some possibly non-linear dynamics $f(\cdot)$. Then equivalently, we can write the dynamics as,

$$P(s' \mid s,a) \propto \exp\left(\frac{||s' - f(s,a)||}{2\sigma^2}\right).$$

Then, by using the reproducing property of Gaussian kernels, the above transition density can be written as,

$$P(s' \mid s,a) = \left\langle k(f(s,a),\cdot), (2\pi\sigma^2)^{d/2} k(s',\cdot)\right\rangle_{\mathcal{H}}.$$

Here $k(\cdot)$ is the Gaussian kernel with bandwidth σ, $k(x,y) = \exp(\frac{||x-y||_2^2}{2\sigma^2})$ and $\mathcal{H}$ is the associated reproducing kernel Hilbert space.

From here, more general non-linear dynamics beyond Gaussian are identified (Ren et al., 2022a; Shribak et al., 2024) to satisfy the desired linear decompositions. In our work, we have shown in Proposition 5.2 that the linear decomposition can be achieved via tensorisation. That is by lifting vector features from SNEFY to matrix spaces. Essentially squaring the 2-norms of a linear statistics is equivalent to taking the trace of the matrix form of the linear statistics and trace is an inner product in the matrix space.

We want to emphasis that only the empirical section of this work leverage linear MDP as means for solving the RL problems. The theoretical result is much more general as the squared family in general satisfies universal approximation properties.

C PROOFS OF SQUARED FAMILY UCRL REGRET BOUNDS

We provide detailed proofs of Theorem 4.5 and Theorem 4.6 in this section. Firstly, we define some concise notations to improve the readability. We then define a martingale around the squared family and use this martingale to bound the MLE errors. This MLE error is then translated to the final regret bound via the simulation lemma and elliptical analysis of the squared family.

C.1 NOTATIONS

Let $T = KH$ be the total number of transition samples. To make notations more succinct, when specific k and h does not matter, we flatten the episodic index (k, h) into a single index $t \in [T]$, and write

$$x_t = (s_t, a_t), \qquad y_t = s_{t+1}.$$

Let $\mathcal{F}_{t-1}$ be the filtration (history) before observing y_t. Under the true parameter $\theta^\star \in \Theta$,

$$y_t \mid \mathcal{F}_{t-1} \sim {}^{\epsilon}P_{\theta^*,t}.$$

Recall ${}^{\epsilon}P_{\theta,t}$ is the mixture squared family probability measure (Definition 4.1).

Let ${}^{\epsilon}p_{\theta,t}$ denote the density of ${}^{\epsilon}P_{\theta,t}$ with respect to the base measure μ. For the smoothed squared-family model, under the normalisation $z_t(\theta) = 1$, we write

$${}^{\epsilon}p_{\theta,t}(y) = (1-\varepsilon)\big(\psi_t(y)^\top\theta\big)^2 + \varepsilon, \qquad \psi_t(y) := \psi(s_t, a_t, y),$$

where $\varepsilon \in (0, 1)$.

We use $\hat{\theta}_t$ to denote the estimated parameter at time t.

For any two densities p, q, define the squared Hellinger distance

$$SH(p:q) := \frac{1}{2}\int \Big(\sqrt{p(y)} - \sqrt{q(y)}\Big)^2 \, \mu(dy).$$

Equivalently,

$$SH(p:q) = 1 - \int \sqrt{p(y)q(y)}\,\mu(dy).$$

The total variation distance is given as,

$$\mathrm{TV}(p:q) = \frac{1}{2}\int |p(y) - q(y)|\mu(\mathrm{d}y).$$

We write $V^{\pi}_{\theta,t}$ to denote the value function under π with estimated transition parameter θ at step t. If an expectation is taking with respect to a squared family with parameter θ, we denote it as $\mathbb{E}_\theta$. We use $\lambda_{\min}(X_t)$ and $\lambda_{\max}(X_t)$ to denote the minimum and maximum eigenvalues of a given PSD matrix X_t. And if the index is not written, we mean the uniform minimum and maximum of the eigenvalue across all t, e.g. $\lambda_{\min}(X) := \inf_t \lambda_{\min}(X_t)$.

C.2 POINT-WISE UNIVERSAL APPROXIMATION PROPERTY

The original definition of squared families does not condition on any random variables and it is shown in Tsuchida et al. (2025) that it exhibits universal approximation properties. The unclear part in this paper is that whether our definition of conditional squared families with normalisations and ϵ smoothing retains this universal approximation property. The next proposition gives positive answer.

Proposition C.1. *Fix any $s, a \in S \times A$. Then restricting Definition 4.1 on the parameter set $\Omega_{s,a} := \{z_\theta(s, a) = 1\}$ does not reduce its point-wise approximation power. Specifically, under the conditions from Theorem 16 of Tsuchida et al. (2025), for any target transition ${}^{\epsilon}P^*(\cdot \mid s, a)$, there exists a normalised parameter $\bar{\theta}$ and a constant $C_u > 0$ such that*

$$\mathrm{TV}({}^{\epsilon}P_\theta(\cdot \mid s, a), P^*(\cdot \mid s, a)) \le (1-\epsilon)\sqrt{\frac{C_u}{2}}d^{-1/8}$$

Proof. When s, a is fixed and under the same conditions as Tsuchida et al. (2025), we know that

$$\mathrm{KL}(p^*(\cdot \mid s,a) : \tilde{p}_\theta(\cdot \mid s,a)) \le C_u d^{-1/4}$$

where p^* is the target density, $\tilde{p}$ is the conditional squared family defined in Definition. 4.1. Because squared family densities are invariant to nonzero rescaling, for fixed s, a we hence have

$$\bar{\theta} = \frac{\theta}{\sqrt{z_\theta(s,a)}},$$

so that $z_{\bar{\theta}}(s,a) = 1$ and $\tilde{p}_\theta = \tilde{p}_{\bar{\theta}}$. Hence the same KL bounds holds for $\bar{\theta}$ too. Hence,

$$\begin{aligned}\mathrm{KL}({}^{\epsilon}P^*(\cdot \mid s,a) : {}^{\epsilon}P_{\bar{\theta}}(\cdot \mid s,a)) &= \mathrm{KL}((1-\epsilon)\tilde{p}^*(\cdot \mid s,a) + \epsilon\mu : (1-\epsilon)\tilde{p}_{\bar{\theta}} + \epsilon\mu) \\ &\le (1-\epsilon)\,\mathrm{KL}(\tilde{p}^*(\cdot \mid s,a) : \tilde{p}_{\bar{\theta}}) + \epsilon\,\mathrm{KL}(\mu : \mu) \\ &\le (1-\epsilon)C_u d^{-1/4}.\end{aligned}$$

The second inequality comes from the joint convexity property of KL divergences. Then apply Pinsker to the above KL bound, we got the desired result. □

This result imply that restricting our parameter set Ω to focus on normalised densities does not destroy expressiveness of the underlying squared family model, since for any given transition density $P^*(\cdot \mid s,a)$, we can find a squared family to closely fit it.

C.3 MLE Concentrations

MLE. Since the base measure of the squared family does not depend on the parameter, we can more cleanly define the MLE in Eq. 8 to be with respect to the density only,

$$\widehat{\theta}_n \in \arg\max_{\theta\in\Omega} \sum_{t=1}^{n} \log {}^{\epsilon}p_{\theta,t}(y_t),$$

where $\Omega = \{\theta \in \mathbb{R}^d \mid \theta_1 \ge \eta > 0, ||\theta||_2 < \mathcal{R}, z(\theta,s,a) = 1\ \forall s,a \in S \times A\}$. The constant η here ensures that the parameter set Ω has an explicit boundary $\theta_1 \ge \eta$ such that the optimal point can lie on the boundary and the optimisation problem is well defined (i.e. remove the case where supreme is achieved but not maximal.) It is not hard to see that the above MLE is well defined as shown in the next lemma.

Lemma C.2 (Existence of the continuous MLE)**.** *Suppose $\Omega \subset \mathbb{R}^d$ is nonempty and compact, and suppose $\varepsilon \in (0,1)$. For each $n \le T$, define*

$$\mathcal{L}_n(\theta) := \sum_{t=1}^{n} \log {}^{\epsilon}p_{\theta,t}(y_t).$$

If ${}^{\epsilon}p_{\theta,t}(y) = (1-\varepsilon)(\psi_t(y)^\top\theta)^2 + \varepsilon$, then $\mathcal{L}_n$ is continuous on Ω. Consequently,

$$\arg\max_{\theta\in\Omega} \mathcal{L}_n(\theta) \neq \varnothing.$$

Proof. Since ${}^{\epsilon}p_{\theta,t} > \epsilon > 0$ and it is continuous, so is the log-likelihood and their summations. Combined with the assumptions, the MLE is well defined. □

We next state two continuity related lemmas that bound the change in log-likelihood and the squared Hellinger distances for squared family densities. These lemmas are used to later control our martingale which eventually control the MLE concentration.

Lemma C.3 (Uniform log-likelihood Lipschitzness)**.** *For any $\theta, \theta' \in \Omega$, under our assumptions*

$$\|\psi_t(y)\|_2 \le B, \qquad \|\theta\|_2 \le R, \qquad z_t(\theta) = 1,$$

and that $\varepsilon \in (0,1)$, the smoothed squared-family log-likelihood is uniformly Lipschitz in θ,

$$|\log {}^{\epsilon}p_{\theta,t}(y) - \log {}^{\epsilon}p_{\theta',t}(y)| \le L_{\log}\|\theta - \theta'\|_2,$$

where one may take

$$L_{\log} := \frac{2(1-\varepsilon)B^2R}{\varepsilon}.$$

Proof. Using

$$^{\epsilon}p_{\theta,t}(y) = (1-\varepsilon)\big(\psi_t(y)^\top\theta\big)^2 + \varepsilon,$$

we have

$$^{\epsilon}p_{\theta,t}(y) \geq \varepsilon.$$

Moreover,

$$\begin{aligned}
|^{\epsilon}p_{\theta,t}(y) - {}^{\epsilon}p_{\theta',t}(y)| &= (1-\varepsilon)\left|\big(\psi_t(y)^\top\theta\big)^2 - \big(\psi_t(y)^\top\theta'\big)^2\right| \\
&= (1-\varepsilon)\left|\psi_t(y)^\top(\theta-\theta')\right|\left|\psi_t(y)^\top(\theta+\theta')\right| \\
&\leq (1-\varepsilon)\|\psi_t(y)\|_2\|\theta-\theta'\|_2\|\psi_t(y)\|_2\|\theta+\theta'\|_2 \\
&\leq 2(1-\varepsilon)B^2R\|\theta-\theta'\|_2.
\end{aligned}$$

Since $u \mapsto \log u$ is $1/\varepsilon$-Lipschitz on $[\varepsilon,\infty)$, it follows that

$$\begin{aligned}
|\log {}^{\epsilon}p_{\theta,t}(y) - \log {}^{\epsilon}p_{\theta',t}(y)| &\leq \frac{1}{\varepsilon}\,|^{\epsilon}p_{\theta,t}(y) - {}^{\epsilon}p_{\theta',t}(y)| \\
&\leq \frac{2(1-\varepsilon)B^2R}{\varepsilon}\|\theta-\theta'\|_2.
\end{aligned}$$

This proves the claim. □

Lemma C.4 (Squared-Hellinger continuity for the smoothed squared family)**.** *For the smoothed squared-family density*

$$^{\epsilon}p_{\theta,t}(y) = (1-\varepsilon)\big(\psi_t(y)^\top\theta\big)^2 + \varepsilon,$$

we have

$$SH(^{\epsilon}p_{\theta,t} : {}^{\epsilon}p_{\theta',t}) \leq \frac{1-\varepsilon}{2}(\theta-\theta')^\top K_t(\theta-\theta').$$

Consequently, if $\lambda_{\max}(K_t) \leq \lambda_{\max}$ *and set* $C_H := \frac{1-\varepsilon}{2}\lambda_{\max}$*, we have*

$$SH(^{\epsilon}p_{\theta,t} : {}^{\epsilon}p_{\theta',t}) \leq C_H\|\theta-\theta'\|_2^2.$$

Proof. Define

$$g(u) := \sqrt{(1-\varepsilon)u^2+\varepsilon}.$$

Then

$$\sqrt{^{\epsilon}p_{\theta,t}(y)} = g\big(\psi_t(y)^\top\theta\big).$$

For any θ,θ', by the fundamental theorem of calculus,

$$\begin{aligned}
\sqrt{^{\epsilon}p_{\theta,t}(y)} - \sqrt{^{\epsilon}p_{\theta',t}(y)} &= g\big(\psi_t(y)^\top\theta\big) - g\big(\psi_t(y)^\top\theta'\big) \\
&= \int_0^1 g'\big(\psi_t(y)^\top\big(\theta'+\tau(\theta-\theta')\big)\big)\,\psi_t(y)^\top(\theta-\theta')\,d\tau.
\end{aligned}$$

Since

$$g'(u) = \frac{(1-\varepsilon)u}{\sqrt{(1-\varepsilon)u^2+\varepsilon}},$$

we have

$$|g'(u)| \leq \sqrt{1-\varepsilon}.$$

Therefore,

$$\left(\sqrt{^{\epsilon}p_{\theta,t}(y)} - \sqrt{^{\epsilon}p_{\theta',t}(y)}\right)^2 \leq (1-\varepsilon)\left(\psi_t(y)^\top(\theta-\theta')\right)^2.$$

Integrating and multiplying by $1/2$ gives

$$\begin{aligned}
SH(^{\epsilon}p_{\theta,t} : {}^{\epsilon}p_{\theta',t}) &= \frac{1}{2}\int\left(\sqrt{^{\epsilon}p_{\theta,t}(y)} - \sqrt{^{\epsilon}p_{\theta',t}(y)}\right)^2\mu(dy) \\
&\leq \frac{1-\varepsilon}{2}\int\left(\psi_t(y)^\top(\theta-\theta')\right)^2\mu(dy) \\
&= \frac{1-\varepsilon}{2}(\theta-\theta')^\top K_t(\theta-\theta').
\end{aligned}$$

The eigenvalue bound gives the final claim. □

Before diving into the MLE concentration result, we state the following martingale concentration theorem, which serves as the backbone to control the parameter concentration.

Theorem C.5 (Finite-time Ville inequality (Ville, 1939; Howard et al., 2020)). *Let $(M_n, \mathcal{F}_n)_{n=0}^T$ be a nonnegative supermartingale. Then, for every $a > 0$,*

$$\mathbb{P}\left(\max_{0\le n\le T} M_n \ge a\right) \le \frac{\mathbb{E}[M_0]}{a}.$$

In particular, if $M_0 = 1$, then

$$\mathbb{P}(\exists n \le T :\ M_n \ge a) \le \frac{1}{a}.$$

The proof of the MLE concentration bound starts by designing a likelihood-ratio martingale for a fixed parameter. Ville's inequality is used to control this point-wise martingale. Then, we use the standard discretisation argument with a union bound to translate the point-wise bound to a bound across the entire parameter set Ω. Here we recall the definition of the α-cover of a set.

Definition C.6 (α-cover (Wainwright, 2019)). *Given a metric space $(\mathbb{X}, d)$, an α-cover of the set $\mathbb{X}$ with respect to the metric d is a set $\{\theta^1, \dots, \theta^T\} \subset \mathbb{X}$ such that for each $\theta \in \mathbb{X}$, there exists*

$$\Pi_\alpha(\theta) = \theta^i \in \Omega_\alpha$$

such that

$$\|\Pi_\alpha(\theta) - \theta\|_2 \le \alpha.$$

The α-covering number is the cardinality of the smallest α-cover.

Theorem C.7 (MLE Concentration Bound). *Under the previous assumptions, let Ω_α be an α-cover of Ω. Then, with probability at least $1-\delta$, simultaneously for all $n \le T$,*

$$\sum_{t=1}^{n} SH({}^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) \le 2\log\frac{|\Omega_\alpha|}{\delta} + \frac{2n(1-\varepsilon)B^2R}{\varepsilon}\alpha + n(1-\varepsilon)\lambda_{\max}\alpha^2.$$

Proof. The proof uses the α-net only as a technical device. The MLE itself is continuous.

Step 1: fixed-parameter likelihood-ratio martingale.

Fix $\theta \in \Omega_\alpha$. Define

$$h_t(\theta) := SH({}^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\theta,t}),$$

and define the one-step likelihood ratio

$$L_t(\theta) := \frac{{}^{\epsilon}p_{\theta,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)}.$$

Because the model is ε-smoothed, all densities are strictly positive, so $L_t(\theta)$ is well-defined.

Define

$$Z_t(\theta) := \frac{\sqrt{L_t(\theta)}}{1-h_t(\theta)},$$

and

$$M_n(\theta) := \prod_{t=1}^{n} Z_t(\theta), \qquad M_0(\theta) := 1.$$

The value $M_0(\theta) = 1$ is the empty-product convention.

Remark. The random variable $Z_t(\theta)$ measures the square-root of the log-likelihood ratio normalised around its mean. As we will soon see that the expectation of the numerator is the denominator. This quantity cannot grow too fast since if θ starts to deviate away from θ^*, $1-h_t(\theta)$ becomes small. And the square-root log likelihood ratio on average is controlled by this balance.

We now verify that $(M_n(\theta))_{n\ge0}$ is a nonnegative martingale. Each $Z_t(\theta)$ is nonnegative because $L_t(\theta) \ge 0$ and

$$1 - h_t(\theta) = \int \sqrt{{}^{\epsilon}p_{\theta^*,t}(y)\,{}^{\epsilon}p_{\theta,t}(y)}\,\mu(dy) > 0.$$

Furthermore, $Z_t(\theta)$ is $\mathcal{F}_t$-measurable, since $x_t = (s_t, a_t)$ is $\mathcal{F}_{t-1}$-measurable and y_t is revealed at time t. It remains to prove the conditional expectation identity. Conditional on $\mathcal{F}_{t-1}$, we have $y_t \sim {}^{\epsilon}p_{\theta^*,t}$. Hence

$$\begin{aligned}\mathbb{E}_{\theta^\star}\left[\sqrt{L_t(\theta)} \mid \mathcal{F}_{t-1}\right] &= \int \sqrt{\frac{{}^{\epsilon}p_{\theta,t}(y)}{{}^{\epsilon}p_{\theta^*,t}(y)}}\, {}^{\epsilon}p_{\theta^*,t}(y)\, \mu(dy) \\ &= \int \sqrt{{}^{\epsilon}p_{\theta,t}(y)\, {}^{\epsilon}p_{\theta^*,t}(y)}\, \mu(dy) \\ &= 1 - h_t(\theta).\end{aligned}$$

Therefore,

$$\mathbb{E}_{\theta^\star}\left[Z_t(\theta) \mid \mathcal{F}_{t-1}\right] = 1.$$

Since

$$M_n(\theta) = M_{n-1}(\theta) Z_n(\theta)$$

and $M_{n-1}(\theta)$ is $\mathcal{F}_{n-1}$-measurable,

$$\begin{aligned}\mathbb{E}_{\theta^\star}\left[M_n(\theta) \mid \mathcal{F}_{n-1}\right] &= M_{n-1}(\theta)\mathbb{E}_{\theta^\star}\left[Z_n(\theta) \mid \mathcal{F}_{n-1}\right] \\ &= M_{n-1}(\theta).\end{aligned}$$

Thus $M_n(\theta)$ is a nonnegative martingale.

Step 2: uniform event over the net.

By Ville's inequality C.5, for fixed $\theta \in \Omega_\alpha$,

$$\mathbb{P}_{\theta^\star}\left(\exists n \le T : M_n(\theta) \ge \frac{|\Omega_\alpha|}{\delta}\right) \le \frac{\delta}{|\Omega_\alpha|}.$$

Taking a union bound over Ω_α, with probability at least $1-\delta$, the following event holds.

$$\mathcal{E} := \left\{\forall \theta \in \Omega_\alpha,\ \forall n \le T : M_n(\theta) \le \frac{|\Omega_\alpha|}{\delta}\right\}$$

On $\mathcal{E}$, for every $\theta \in \Omega_\alpha$ and every $n \le T$,

$$\log M_n(\theta) \le \log \frac{|\Omega_\alpha|}{\delta}.$$

Expanding the definition of $M_n(\theta)$ gives

$$\frac{1}{2}\sum_{t=1}^{n} \log \frac{{}^{\epsilon}p_{\theta,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)} - \sum_{t=1}^{n} \log(1 - h_t(\theta)) \le \log \frac{|\Omega_\alpha|}{\delta}.$$

Since $-\log(1-u) \ge u$ for $u \in [0,1)$,

$$\sum_{t=1}^{n} h_t(\theta) \le \log \frac{|\Omega_\alpha|}{\delta} - \frac{1}{2}\sum_{t=1}^{n} \log \frac{{}^{\epsilon}p_{\theta,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)}.$$

Equivalently,

$$\sum_{t=1}^{n} SH({}^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\theta,t}) \le \log \frac{|\Omega_\alpha|}{\delta} - \frac{1}{2}\sum_{t=1}^{n} \log \frac{{}^{\epsilon}p_{\theta,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)}.$$

Step 3: approximate MLE estimations by points in Ω_α.

For each $n \le T$, define the closest point in Ω_α for a given estimated $\hat{\theta}_n$ as

$$\bar{\theta}_n := \Pi_\alpha(\widehat{\theta}_n) \in \Omega_\alpha.$$

Then

$$\|\bar{\theta}_n - \widehat{\theta}_n\|_2 \le \alpha.$$

Because $\mathcal{E}$ holds uniformly over all net points, we may apply the previous inequality to $\theta = \bar{\theta}_n$:

$$\sum_{t=1}^{n} SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\bar{\theta}_n,t}) \le \log \frac{|\Omega_\alpha|}{\delta} - \frac{1}{2}\sum_{t=1}^{n} \log \frac{{}^{\epsilon}p_{\bar{\theta}_n,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)}.$$

Now use continuous-MLE optimality. Since $\theta^\star \in \Omega$ and $\widehat{\theta}_n$ maximises the likelihood over Ω,

$$\sum_{t=1}^{n} \log {}^{\epsilon}p_{\widehat{\theta}_n,t}(y_t) \ge \sum_{t=1}^{n} \log {}^{\epsilon}p_{\theta^*,t}(y_t).$$

Also, by log-likelihood Lipschitzness C.3,

$$\log {}^{\epsilon}p_{\bar{\theta}_n,t}(y_t) \ge \log {}^{\epsilon}p_{\widehat{\theta}_n,t}(y_t) - L_{\log}\|\bar{\theta}_n - \widehat{\theta}_n\|_2 \ge \log {}^{\epsilon}p_{\widehat{\theta}_n,t}(y_t) - L_{\log}\alpha.$$

Summing over $t = 1, \dots, n$ gives

$$\sum_{t=1}^{n} \log {}^{\epsilon}p_{\bar{\theta}_n,t}(y_t) \ge \sum_{t=1}^{n} \log {}^{\epsilon}p_{\widehat{\theta}_n,t}(y_t) - nL_{\log}\alpha.$$

Therefore,

$$\sum_{t=1}^{n} \log \frac{{}^{\epsilon}p_{\bar{\theta}_n,t}(y_t)}{{}^{\epsilon}p_{\theta^*,t}(y_t)} \ge -nL_{\log}\alpha.$$

Substituting into the net-point inequality yields

$$\sum_{t=1}^{n} SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\bar{\theta}_n,t}) \le \log \frac{|\Omega_\alpha|}{\delta} + \frac{1}{2}nL_{\log}\alpha.$$

Step 4: transfer the bound from Ω_α to Ω.

Let

$$H(p : q) := \sqrt{SH(p : q)}$$

be the Hellinger distance. It satisfies the triangle inequality. Hence, for each t,

$$H(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) \le H(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\bar{\theta}_n,t}) + H(p_{\bar{\theta}_n,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}).$$

Using $(a+b)^2 \le 2a^2 + 2b^2$, we obtain

$$SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) \le 2SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\bar{\theta}_n,t}) + 2SH(^{\epsilon}p_{\bar{\theta}_n,t} {}^{\epsilon}{:}p_{\widehat{\theta}_n,t}).$$

Summing over $t = 1, \dots, n$,

$$\begin{aligned}\sum_{t=1}^{n} SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) &\le 2\sum_{t=1}^{n} SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\bar{\theta}_n,t}) + 2\sum_{t=1}^{n} SH(p_{\bar{\theta}_n,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) \\ &\le 2\left(\log \frac{|\Omega_\alpha|}{\delta} + \frac{1}{2}nL_{\log}\alpha\right) + 2nC_H\alpha^2.\end{aligned}$$

Therefore,

$$\sum_{t=1}^{n} SH(^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\widehat{\theta}_n,t}) \le 2\log \frac{|\Omega_\alpha|}{\delta} + nL_{\log}\alpha + 2nC_H\alpha^2.$$

Choice of α. Taking $n = T$ gives the continuous-MLE confidence radius

$$\beta_T^{\text{cont}} := 2\log \frac{|\Omega_\alpha|}{\delta} + TL_{\log}\alpha + 2TC_H\alpha^2.$$

For the smoothed squared-family constants above,

$$\beta_T^{\text{cont}} = 2\log \frac{|\Omega_\alpha|}{\delta} + \frac{2T(1-\varepsilon)B^2R}{\varepsilon}\alpha + T(1-\varepsilon)\lambda_{\max}\alpha^2.$$

Choosing $\alpha = \frac{1}{T}$ yields

$$\beta_T^{\text{cont}} = 2\log\frac{|\Omega_{1/T}|}{\delta} + \frac{2(1-\varepsilon)B^2R}{\varepsilon} + \frac{(1-\varepsilon)\lambda_{\max}}{T}.$$

If $\Omega \subseteq \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq R\}$, then

$$|\Omega_\alpha| \leq \left(\frac{cR}{\alpha}\right)^d,$$

so

$$\log|\Omega_{1/T}| \leq d\log(cRT).$$

Thus

$$\beta_T = O\left(d\log(cRT) + \log\frac{1}{\delta} + \frac{(1-\varepsilon)B^2R}{\varepsilon}\right) = \widetilde{O}(d).$$

This prove our central claim. □

Corollary C.8. *Define the confidence set,*

$$C_n := \left\{\theta \in \Omega \mid \sum_{t=1}^{n} SH({}^{\epsilon}p_{\hat{\theta},t} : {}^{\epsilon}p_{\theta,t}) \leq \beta_n\right\}.$$

Then the following holds,

$$P(\theta^\star \in C_n, \forall n \leq T) \geq 1-\delta.$$

Proof. By the symmetry of the Squared Hellinger distance,

$$\{\theta^\star \in C_t\} = \{SH({}^{\epsilon}p_{\hat{\theta},t} : {}^{\epsilon}p_{\theta^\star,t}) \leq \beta_t\}.$$

Then from Theorem C.7 we know the intersection of events,

$$\bigcap_{n=1}^{T}\{\theta^\star \in C_n\} = \bigcap_{n=1}^{T}\{SH({}^{\epsilon}p_{\hat{\theta},n} : {}^{\epsilon}p_{\theta^\star,n}) \leq \beta_n\} := \mathcal{E}_T$$

happens with probability $1-\delta$. □

Up to this point, we have proven Theorem 4.5 stated in the main text. The confidence set with high probability retains the optimal parameter. However, the OFU mechanism (line 5) in Algorithm. 1 uses the most optimal parameter in the confidence set, not the MLE estimator, to solve for the optimal policy. We define this OFU parameter as,

$$\{\tilde{\theta}_t, \pi_t\} = \underset{\theta\in C_t, \pi\in\Pi}{\arg\max}\, V_{\theta,0}^{\pi}(s_0). \tag{14}$$

The next corollary bounds the squared Hellinger distance between the θ_t^* and $\tilde{\theta}_t$.

Corollary C.9. *For any $t < T$ and for fixed $\theta_t^*, \tilde{\theta}_t \in C_t$, the following bounds hold deterministically,*

$$\sum_{t<T} \text{SH}({}^{\epsilon}p_{\theta^*,t} : {}^{\epsilon}p_{\tilde{\theta},t}) \leq 4\beta_T$$

Proof. Since we know that H is also a metric, then for any $\theta_t^*, \tilde{\theta}_t$,

$$H({}^{\epsilon}p_{\theta^*,t}, {}^{\epsilon}p_{\tilde{\theta},t}) \leq H({}^{\epsilon}p_{\theta^*,t}, {}^{\epsilon}p_{\hat{\theta},t}) + H({}^{\epsilon}p_{\hat{\theta},t}, {}^{\epsilon}p_{\tilde{\theta},t}) \tag{15}$$

Using $(a+b)^2 \leq 2a^2 + 2b^2$ again and summing over past observations we get the desired result. □

We next see how this confidence set gets translated to the regret through simulation lemma and elliptical potential analysis designed around the squared family.

C.4 SIMULATION LEMMA

We use a path-wise simulation lemma approach (Kakade et al., 2020) and bound the difference in expectation of the value functions in terms of the squared Hellinger distance between two squared families.

Lemma C.10 (Squared Hellinger Distance Simulation Lemma)**.** *For episode k, let $\tilde{\theta}_k \in C_k$ be the OFU parameter and let $\pi \in \Pi$ be any policy. Define $\Delta_h = V^{\pi}_{\tilde{\theta}_k,h} - V^{\pi}_{\theta^*,h}$ and*

$$\zeta^k_h := \mathbb{E}_{s' \sim {}^{\epsilon}P_{\theta^*}(\cdot|s^k_h,a^k_h)}[\Delta_{h+1}(s')] - \Delta_{h+1}(s^k_{h+1}).$$

Then, for any initial state $s_0 \in S$,

$$V^{\pi}_{\tilde{\theta}_k,0} - V^{\pi}_{\theta^*,0} \le 2H \sum_{h=1}^{H} \sqrt{2\,\mathrm{SH}({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h})} + \sum_{h=1}^{H} \zeta^k_h$$

where the expectation is taken over the trajectory generated by π under the true parameter θ^*.

Moreover, on the good event $\theta^* \in C_k$, if

$$(\tilde{\theta}_k, \pi_k) \in \operatorname*{arg\,max}_{\theta \in C_k, \pi \in \Pi} V^{\pi}_{\theta,0}(s_0),$$

then the per-episode regret satisfies,

$$V^{\pi^*}_{\theta^*,0}(s_0) - V^{\pi_k}_{\theta^*,0}(s_0) \le 2H \sum_{h=1}^{H} \sqrt{2\,\mathrm{SH}({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h})} + \sum_{h=1}^{H} \zeta^k_h.$$

Proof. Since the reward function is known and stays the same, the Bellman equation gives,

$$V^{\pi}_{\tilde{\theta}_k,h}(s^k_h) = r(s^k_h, a^k_h) + \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\tilde{\theta}_k}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right],$$

and

$$V^{\pi}_{\theta^*_k,h}(s^k_h) = r(s^k_h, a^k_h) + \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\theta^*,h+1}(s^k_{h+1}) \right],$$

Subtracting the two gives,

$$V^{\pi}_{\tilde{\theta}_k,h} - V^{\pi}_{\theta^*_k,h}(s^k_h) = \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\tilde{\theta}_k}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right] - \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\theta^*,h+1}(s^k_{h+1}) \right]$$

Add and subtracting $\mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right]$ yeilds

$$\begin{aligned}
&V^{\pi}_{\tilde{\theta}_k,h}(s^k_h) - V^{\pi}_{\theta^*_k,h}(s^k_h) \\
&= \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\tilde{\theta}_k}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right] \\
&- \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right] \\
&+ \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) - V^{\pi}_{\theta^*,h+1}(s^k_{h+1}) \right]
\end{aligned}$$

To bound the first two terms, we note that the per-step reward is bounded in $[0, 1]$ and therefore the value is bounded by H, hence

$$\begin{aligned}
&\mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\tilde{\theta}_k}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right] - \mathbb{E}_{s^k_{h+1} \sim {}^{\epsilon}P_{\theta^*}} \left[V^{\pi}_{\tilde{\theta}_k,h+1}(s^k_{h+1}) \right] \\
&\le H \int_S \left| {}^{\epsilon}P_{\tilde{\theta}_k}(\mathrm{d}s' \mid s^k_h, a^k_h) - {}^{\epsilon}P_{\theta^*}(\mathrm{d}s' \mid s^k_h, a^k_h) \right| \\
&= 2H\,\mathrm{TV}\left({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h} \right) \\
&\le 2H \sqrt{2\,\mathrm{SH}\left({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h} \right)}.
\end{aligned}$$

The last inequality comes from basic divergence inequalities from Lemma C.16.

For the last term, we note that

$$\mathbb{E}_{s_{h+1}^k \sim {}^{\epsilon}P_{\theta^*}}\left[V^{\pi}_{\tilde{\theta}_k,h+1}(s_{h+1}^k) - V^{\pi}_{\theta^*,h+1}(s_{h+1}^k)\right] = \Delta_{h+1}(s_{h+1}^k) + \zeta_h^k.$$

Then, this gives us the following per step inequality,

$$\Delta_h(s_h^k) \leq 2H\sqrt{2\,\mathrm{SH}\left({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h}\right)} + \Delta_{h+1}(s_{h+1}^k) + \zeta_h^k.$$

Hence by recursively expanding on the next Δ_{h+1}, we arrive at

$$\begin{aligned}\Delta_1(s_0^k) &\leq 2H\sqrt{2\,\mathrm{SH}\left({}^{\epsilon}P^k_{\theta^*,0} : {}^{\epsilon}P^k_{\tilde{\theta}_k,0}\right)} + \zeta_1^k + \Delta_2(s_2^k)\\ &\vdots\\ &\leq 2H\sum_{h=1}^{H}\sqrt{2\,\mathrm{SH}\left({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h}\right)} + \sum_{h=1}^{H}\zeta_h^k + \Delta_{H+1}(s_{H+1}^k),\end{aligned}$$

By noting that $\Delta_{H+1} = 0$ by definition that any reward beyong H is zero, we obtain the first claim. For the second claim that is conditioned on the good event, optimism gives

$$V^{\pi_k}_{\tilde{\theta}_k,0}(s_0) \geq V^{\pi^*}_{\theta^*,0}(s_0)$$

Hence,

$$V^{\pi^*}_{\theta^*,0}(s_0) - V^{\pi_k}_{\theta^*,0}(s_0) \leq V^{\pi_k}_{\tilde{\theta}_k,0}(s_0) - V^{\pi_k}_{\theta^*,0}(s_0),$$

which proves the second claim. □

C.5 Regret Bounds

At this point, a clear path is given to provide a regret bound for our setting. We show that the regret induced by Algorithm 1 is recursively bounded by the estimation errors from the MLE and this error cannot grow too quickly. The core of bounding the regret is to bound the sum of squared Hellinger distances between two mixture squared families (Lemma C.10). This can be done using our carefully designed elliptical potential analysis provided in the next section of the proof together with the MLE concentration results from Theorem C.7. Here we combine everything together to firstly derive the final regret bound stated in Theorem 4.6 before diving into the details of elliptical potential analysis.

Proof. We condition on two events, $\mathcal{E}_1 = \{\theta^* \in C_k, \forall k \leq K\}$, and $\mathcal{E}_2 = \{\sum_{t=1}^{T}\zeta_t \leq 2H\sqrt{2T\log\frac{2}{\delta}}\}$. The first event ensures the true parameter is retained, it comes from our concentration result. Indeed $P(\mathcal{E}_1) \geq 1 - \frac{\delta}{2}$ by Theorem C.7. For the second event, since $|\zeta_t| \leq 2H$ and $\mathrm{E}[\zeta_t \mid \mathcal{F}_t] = 0$, apply Azuma-Hoeffding inequality to $(\zeta_t)_{t<T}$, we have $P(\mathcal{E}_2) \geq 1 - \frac{\delta}{2}$. Taking a union bound over the two events, we have with probability $1-\delta$, the following happen,

$$\begin{aligned}\mathrm{Regret}(K) &= \sum_{k=1}^{K} V^{\pi^*}_{\theta^*,0}(s_0^k) - V^{\pi_k}_{\theta^*,0}(s_0^k)\\ &\leq \sum_{k=1}^{K} V^{\pi_k}_{\tilde{\theta}_k}(s_0^k) - V^{\pi_k}_{\theta^*,0}(s_0^k)\\ &\leq 2H\sum_{k=1}^{K}\sum_{h=1}^{H}\sqrt{2\,\mathrm{SH}({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h})} + 2H\sqrt{2T\log\frac{2}{\delta}}\\ &\leq 2H\sqrt{KH\sum_{k=1}^{K}\sum_{h=1}^{H}2\,\mathrm{SH}({}^{\epsilon}P^k_{\theta^*,h} : {}^{\epsilon}P^k_{\tilde{\theta}_k,h})} + 2H\sqrt{2T\log\frac{2}{\delta}}\\ &\leq 4\sqrt{KH^3(1-\epsilon)d\log(1+T)\left(\frac{8\beta_T}{c_\epsilon} + 4H\lambda_{\max}\mathcal{R}^2\right)} + 3H\sqrt{T\log\frac{2}{\delta}}.\end{aligned}$$

The first inequality comes from OFU principle. Then we separate the good and bad events in the second inequality, where the regret is upper bounded by H for the bad events. Then, we invoked Lemma C.10 on the good events followed by CS inequality. For the last inequality, we invoke Lemma C.14 to conclude the proof. □

When ignoring constants and logarithms, the first term in the final regret bound dominates the growth. In particular, the term $\sqrt{KH^3d\beta_T}$ contributes to $\tilde{O}(dH\sqrt{T})$ and the term $\sqrt{KH^3dH}$ contributes to $\tilde{O}(\sqrt{d}H^{3/2}\sqrt{T})$. Taking the max of the two, we get

$$\tilde{O}(H\sqrt{T}\cdot\max(d,\sqrt{dH})) = \begin{cases}\tilde{O}(dH\sqrt{T}), \text{ if } d \geq H,\\ \tilde{O}(H^{3/2}\sqrt{dT}), \text{ otherwise.}\end{cases}$$

In most common RL settings, feature dimensions d are much larger than the horizon, making the first case above more standard.

C.6 Elliptical Potential Lemmas

Naively bounding the sums of SH distances by the width of the confidence set from Theorem 4.5 will result in a linear growth of regret with respect to episode number K. This is because the confidence set bounds the total SH differences for a single MLE estimator, whereas the MLE estimator varies across episodes when running Algorithm 1. To remove redundancies, we design an information matrix that summarises the cumulative information and use a form of elliptical potential lemma to control the growth of the regret.

Firstly, we show that the SH distance between two mixture squared families can be written in a quadratic form that permits elliptical potential style analysis.

For any parameter θ, let $\tilde{l}_{t,\theta} := \psi_t(s_t,a_t,s_{t+1})^\top\theta$ denote the linear statistic in step t. And let $g(x) = \sqrt{(1-\epsilon)x^2+\epsilon}$. Hence, we have $\sqrt{{}^{\epsilon}P^k_{\theta,h}} = g(\tilde{l}_{t,\theta})$ as we have $z(\theta)=1$. Apply the fundamental theorem of calculus along the parameter segment $\theta_\tau := \theta+\tau(\theta^*-\theta)$ with $\tau\in[0,1]$,

$$g(\tilde{l}_{t,\theta^*}) - g(\tilde{l}_{t,\theta}) = \int_0^1 \frac{\mathrm{d}}{d\tau} g(\tilde{l}_{t,\theta_\tau})\mathrm{d}\tau = \int_0^1 g'(\tilde{l}_{t,\theta_\tau})\psi_t^\top(\theta^*-\theta)\mathrm{d}\tau. \tag{16}$$

The second equality above comes from the chain rule.

Define $\iota_{\theta_t,\theta^*}(s_t,a_t,s_{t+1}) = \int_0^1 g'(\tilde{l}_{t,\theta_\tau})\mathrm{d}\tau$. Then we have the following lemma.

Lemma C.11. *For any step t, the squared Hellinger distance between two mixture squared families satisfies,*

$$\mathrm{SH}({}^{\epsilon}P_{\theta,t} : {}^{\epsilon}P_{\theta',t}) = \frac{1}{2}(\theta-\theta')^\top\mathcal{K}^\epsilon_t(\theta-\theta'), \tag{17}$$

where

$$\mathcal{K}^\epsilon_t(\theta,\theta') = \int \iota^2_{\theta_t,\theta'}(s_t,a_t,s')\psi_t(s_t,a_t,s')\psi_t(s_t,a_t,s')^\top\mu(\mathrm{d}s'). \tag{18}$$

Proof. From (16) and our definition of ι_t, we obtain

$$g(\tilde{l}_{t,\theta^*}) - g(\tilde{l}_{t,\theta}) = \iota_t\psi_t^\top(\theta^*-\theta).$$

Then, substituting this into the definition of SH (5) and expanding, we get the desired result. □

In the above lemma, we have defined the path averaged squared kernel $\mathcal{K}^\epsilon_t$. It governs the local geometry of two densities parameterised by θ,θ'. The next lemma relates the path averaged squared kernel with the usual squared kernel, so that in the latter proofs we can upper bound the path average squared kernel with the standard squared kernel that does not depend on the parameters.

Lemma C.12. *The path averaged squared kernel is upper bounded by the usual squared kernel,*

$$\mathcal{K}^\epsilon_t(\theta,\theta') \preceq (1-\epsilon)\mathcal{K}_t.$$

Proof. Computing the derivative of $g(x)$ with respect to some inputs x,

$$g'(x) = \frac{(1-\epsilon)x}{\sqrt{(1-\epsilon)x^2+\epsilon}}.$$

Squaring both sides,

$$g'(x)^2 = \frac{(1-\epsilon)^2x^2}{(1-\epsilon)x^2+\epsilon} \leq 1-\epsilon.$$

Hence, $|g'(x)| \leq \sqrt{1-\epsilon}$. This means,

$$|\iota_{\theta,\theta'}(s_t,a_t,s_{t+1})| = \left|\int_0^1 g'(\tilde{l}_{t,\theta_\tau})\mathrm{d}\tau\right| \leq \int_0^1 |g'(\tilde{l}_{t,\theta_\tau})|\mathrm{d}\tau \leq \sqrt{1-\epsilon}$$

By substitution,

$$K_t^\epsilon(\theta,\theta') \preceq (1-\epsilon)K_t$$

□

The above lemma together with Assumption 4.4 gives us the following uniform sandwich result on the path averaged kernel for any $t < T$,

$$c_\epsilon \mathcal{K}_t \leq \mathcal{K}_t^\epsilon(\theta,\theta') \leq (1-\epsilon)\mathcal{K}_t \tag{19}$$

This bounds allows us the remove the dependence on the specific parameters $\tilde{\theta}_t, \theta^*$ which is important for controlling the telescoping sums later in the proof.

Conditioned on the good event that our concentration result in Theorem 4.5 holds, we can now proceed to analyse the elliptical potential of the quadratic form in (17). The next lemma defines a norm bound with respect to the following information matrix. Define

$$J_0 = \lambda_{\max} I_d, \text{and } J_t = J_{t-1} + \mathcal{K}_t.$$

Here, J_t can be viewed as the empirical covariance matrix that controls the exploration of the algorithm.

Lemma C.13. *Fix an episode $k \in [K]$ and let $\tilde{\theta}_k \in C_k$ be the OFU parameter. Then very every step t in episode k with $t_k \leq t \leq t_k + H$ we have*

$$||\theta^* - \tilde{\theta}_t||^2_{J_{t-1}} \leq \frac{8\beta_{t_1}}{c_\epsilon} + 4H\lambda_{\max}\mathcal{R}^2.$$

Proof. We abuse the notation Δ and let $\Delta_t := \theta^* - \tilde{\theta}_k$, then

$$4\beta_{k-1} \geq \sum_{i=1}^{t_k} \mathrm{SH}({}^\epsilon P_{\theta^*,t} : {}^\epsilon P_{\tilde{\theta}_t,i}) \tag{20}$$

$$= \frac{1}{2}\Delta_t^\top \sum_{i=1}^{t_k} \mathcal{K}_t^\epsilon \Delta_t \tag{21}$$

$$\geq \frac{c_\epsilon}{2}\Delta_t^\top \sum_{i=1}^{t_k} \mathcal{K}_t \Delta_t. \tag{22}$$

The first inequality is our concentration result and the second inequality comes from Eq. 19. Hence,

$$\Delta_t^\top \sum_{i=1}^{t_k} \mathcal{K}_t \Delta_t \leq \frac{8\beta_{t-1}}{c_\epsilon}.$$

By the definition of J_t, we can split it according to the past trajectories and the steps within the current trajectory,

$$J_{t-1} = \lambda_{\max} I_d + \sum_{i=1}^{t_k} K_i + \sum_{i=t_k+1}^{t-1} K_i,$$

then

$$||\Delta_t||_{J_{t-1}} = \Delta_t^\top \sum_{i=1}^{t-1} \mathcal{K}_i \Delta_t + \Delta_t^\top \sum_{i=1}^{t_k} \mathcal{K}_i \Delta_t + \lambda_{\max} ||\Delta_t||_2^2 \tag{23}$$

$$\leq \frac{8\beta_{t-1}}{c_\epsilon} + 4H\lambda_{\max} ||\Delta_t||_2^2. \tag{24}$$

where the last inequality comes from $\Delta^T \mathcal{K}_i \Delta \leq \lambda_{\max} ||\Delta||_2^2 \leq 4\lambda_{\max} R^2$. □

As we vary the MLE estimator over time, the growth of the SH distance can be controlled via the next lemma.

Lemma C.14. *In general, for any total steps T and any $\{\hat{\theta}_1, \ldots, \hat{\theta}_T\} \in \Theta^T$, we have*

$$\sum_{t=1}^{T} \mathrm{SH}({}^{\epsilon}P_{\theta^*,t} : {}^{\epsilon}P_{\tilde{\theta},t}) \leq (1-\epsilon) d \log(1+T) \left(\frac{8\beta_T}{c_\epsilon} + 4\lambda_{\max} \mathcal{R}^2 \right).$$

Proof. Again denote $\Delta_t = \theta_t^* - \hat{\theta}_t$. Then, using the properties of trace, for a single squared Hellinger distance,

$$\begin{aligned}
\mathrm{SH}({}^{\epsilon}P_{\theta^*,t} : {}^{\epsilon}P_{\tilde{\theta},t}) &\leq \frac{1-\epsilon}{2} \Delta_t^\top \mathcal{K}_t \Delta_t \\
&= \frac{1-\epsilon}{2} \mathrm{Tr}(\mathcal{K}_t \Delta_t^\top \Delta_t) \\
&= \frac{1-\epsilon}{2} \mathrm{Tr}(J_{t-1}^{-1/2} \mathcal{K}_t J_{t-1}^{-1/2} J_{t-1}^{1/2} \Delta_t \Delta_t^\top J_{t-1}^{1/2}) \\
&\leq \frac{1-\epsilon}{2} \mathrm{Tr}(J_{t-1}^{-1} \mathcal{K}_t) ||\Delta_t||_{J_{t-1}}^2
\end{aligned}$$

where the first inequality comes from Eq. 19 and the last inequality comes from squaring $J_{t-1}^{1/2}$. Then consider the sums,

$$\begin{aligned}
\sum_{t=1}^{T} \mathrm{SH}({}^{\epsilon}P_{\theta^*,t} : {}^{\epsilon}P_{\tilde{\theta},t}) &\leq \frac{1-\epsilon}{2} \left(\frac{8\beta_T}{c_\epsilon} + 4\mathcal{R}^2 \right) \sum_{t=1}^{T} (\mathrm{Tr}(J_{t-1}^{-1} \mathcal{K}_t)) \\
&\leq (1-\epsilon) d \log(1+T) \left(\frac{8\beta_T}{c_\epsilon} + 4\lambda_{\max} \mathcal{R}^2 \right).
\end{aligned}$$

The first inequality comes from Lemma C.13 and realising that β_t is a non-decreasing sequence. The second inequality comes from the elliptical potential lemma (Lemma C.15) which gets us to the desired result. □

The final thing we need to bound from the previous lemma is the summation of traces. Define $\nabla_t = \mathrm{Tr}(J_{t-1}^{-1} \mathcal{K}_t)$, we next state and prove the adapted matrix elliptical potential lemma that is used to control ∇_t,

Lemma C.15. *For any PSD matrix $\mathcal{K}_t^\epsilon$ with rank d. We have*

$$\sum_{t=1}^{T} \nabla_t \leq 2d \log(1+T).$$

Proof. Since the trace of a matrix is the sum of its eigenvalues and by definition J_t is still a PSD matrix (and full rank), we can write $\nabla_t = \sum_{i=1}^{d} \lambda_i$, where d is the rank of $\mathcal{K}_t$.

Next, define $A_t = J_{t-1}^{-1/2} \mathcal{K}_t J_{t-1}^{-1/2}$, so $\nabla_t = \mathrm{Tr}(A_t)$. Furthermore, $\lambda_{\max}(A_t) = ||A_t||_{op} \leq ||J_{t-1}^{-1/2}||_{op}^2 ||\mathcal{K}_t||_{op}$ and hence $\lambda_{\max}(A_t) \leq \frac{||\mathcal{K}_t||_{op}}{\lambda_{\max}} \leq 1$. Meanwhile, by the inequality $\min\{x, 1\} \leq 2\log(x+1)$ for any $x \geq 0$. We know $\lambda_i \leq 2\log(\lambda_i + 1)$ for every i. And hence,

$$\nabla_t = \sum_{i=1}^{d} \lambda_i \leq 2 \sum_{i=1}^{d} \log(\lambda_i + 1) \tag{25}$$

Using the fact that the determinant of a matrix is the product of its eigenvalues, and when adding an identity matrix to a PSD matrix, its eigenvalues are added by 1, we have

$$\begin{aligned}\det(J_t) &= \det(J_{t-1}^{1/2}(I + A_t)J_{t-1}^{1/2}) \\ &= \det(J_{t-1})\det(I + A_t) \\ &= \det(J_{t-1})\prod_{i=1}^{d}(1+\lambda_i).\end{aligned}$$

This implies,

$$\log\frac{\det(J_t)}{\det(J_{t-1})} = \sum_{i=1}^{d}\log(1+\lambda_i). \tag{26}$$

Then following from Equation 25 and Equation 26 and observe the telescoping sum cancels the inner determinants, we have

$$\sum_{t=1}^{T}\nabla_t \le 2\sum_{t=1}^{T}\log\frac{\det(J_t)}{\det(J_{t-1})} = 2\log\frac{\det(J_T)}{\det(J_0)}.$$

But,

$$\begin{aligned}\frac{\det(J_T)}{\det(J_0)} &= \frac{\det(\lambda_{\max}I + \sum_{t=1}^{T}\mathcal{K}_t)}{\lambda_{\max}^d} \\ &= \det\left(I + \frac{1}{\lambda_{\max}}\sum_{t=1}^{T}\mathcal{K}_t\right) \\ &= \prod_{i=1}^{d}(1+\kappa_i) \\ &\le \left(1 + \frac{1}{d}\sum_{i=1}^{d}(\kappa_i)\right)^d \\ &= \left(1 + \frac{1}{\lambda_{\max}d}\mathrm{Tr}\left(\sum_{t=1}^{T}\mathcal{K}_t\right)\right)^d\end{aligned}$$

where we have used κ_i to denote the eigenvalues of the matrix $\frac{1}{\lambda_{\max}}\sum_{t=1}^{T}\mathcal{K}_t$ and the inequality comes from AM-GM. Since

$$\mathrm{Tr}(\mathcal{K}_t) \le d||\mathcal{K}_t||_{op} \le d\lambda_{\max},$$

where last inequality replaces the path averaged squared kernel with the standard kernel used throughout the algorithm by invoking Lemma C.12, we hence have,

$$\left(1 + \frac{1}{\lambda_{\max}d}\mathrm{Tr}\left(\sum_{t=1}^{T}\mathcal{K}_t\right)\right)^d \le \left(1 + \frac{T\lambda_{\max}}{\lambda_{\max}}\right)^d.$$

Putting everything together, we reach the intended conclusion. □

The lemma below provides us with bounds between different divergences that are used throughout the proofs.

Lemma C.16. *For two densities p, q, the following divergences inequalities hold,*

$$\mathrm{TV}(p:q) \le \sqrt{2}\sqrt{\mathrm{SH}(p:q)} \le \sqrt{\mathrm{KL}(p:q)}$$

where SH *is the squared Hellinger distance,* TV *is the total variation, and* KL *is the KL divergence.*

Proof. See Sason & Verdú (2016). □

	Squared-family MDPs	Exponential-family MDPs
Density	$p_\theta(x_t) = (1-\varepsilon)\,\frac{\left(\psi_t(x)^\top \theta\right)^2}{Z(\theta, s_t, a_t)} + \varepsilon$	$p_\theta(x_t) = h(x)\exp\Big(\psi(s_{t+1})^\top M_\theta\,\phi(s_t, a_t) - \bar{Z}(\theta, s_t, a_t)\Big)$
Normalisation	$Z(\theta, s_t, a_t) = \theta^\top K_t \theta$	$\bar{Z}(\theta, s_t, a_t) = \log \int_S \exp(\psi(s_{t+1})^\top M_\theta \phi(s_t, a_t)) h(x) ds_{t+1}$
Unknown parameters	$\theta \in \mathbb{R}^d$	$M_\theta = \sum_{i=1}^{d} \theta_i A_i,$ $\theta \in \mathbb{R}^d$ unknown
Parameter set	$\Theta = \{\theta \in \mathbb{R}^d : \theta_1 \geq \eta > 0,\ \|\theta\|_2 \leq R,$ $z_t(\theta) = 1\}$	$\theta \in \mathbb{R}^d$
MLE	$\arg\max_{\theta\in\Theta} \sum_{t=1}^{n} \log p_\theta(x_t)$	$\arg\max_{\theta\in\mathbb{R}^d} \sum_{t=1}^{n} -\log p_\theta(x_t) + \frac{\eta}{2}\|\theta\|_A^2$
Natural geometry	$\mathrm{SH}(p:q) = \frac{1}{2}\int \left(\sqrt{p} - \sqrt{q}\right)^2 \mathrm{d}\mu$	$\mathrm{KL}_{s,a}(\theta, \theta') = Z_{s,a}(\theta') - Z_{s,a}(\theta) - (\theta' - \theta)^\top \nabla Z_{s,a}(\theta)$
Core assumption	$C_\epsilon \mathcal{K}_t \preceq \mathcal{K}_t^\epsilon\ \forall t \in [T], s, a \in S \times A$	$Z_{s,a}(\theta)$ is strictly convex for all $s, a \in S \times A$.

Table 3: Comparison between squared-family MDPs and exponential-family MDPs.

D SQUARED FAMILY VS EXPONENTIAL FAMILY MDPS

We compare the proposed squared family MDP setting with the exponential family MDP framework from Chowdhury et al. (2021). Both settings study finite horizon model-based reinforcement learning with unknown parametric transition kernels, known rewards, and optimism-based exploration. The main distinction is the statistical geometry used to control model estimation: squared family MDPs are naturally controlled in squared Hellinger geometry, whereas exponential family MDPs are controlled in KL/Bregman geometry induced by the log-partition function. This fundamental difference induces different martingale formations to control the parameter estimation errors and therefore regret.

Problem setup and assumptions. The two models impose different parametric structures on the transition kernel. In the squared-family setting, the conditional transition density is obtained by squaring a linear statistic and adding a smoothing component. In contrast, Chowdhury et al. model the transition density as a bilinear exponential family (exponentiation of a linear statistics). Theoretically, both models exhibit universal approximation properties, making them attractive for modelling arbitrary transition densities in RL. A concise comparison is given in Table 3.

We use t to index the flattened transition samples, with $x_t = (s_t, a_t, s_{t+1})$ and $p_\theta(x_t) := p_\theta(s_{t+1} \mid s_t, a_t)$ is the density function and $P_\theta(x_t)$ is the probability measure. In the **squared family model**, $\psi(x_t)$ is a known feature map, θ is the unknown transition parameter, $\varepsilon \in (0, 1)$ is a smoothing constant, and $z_t(\theta) = 1$ denotes the imposed normalisation. The main assumptions are bounded features, $\|\psi(x_t)\|_2 \leq B$, bounded parameters, compactness of Θ, strict positivity from smoothing, bounded eigenvalues of the squared-family kernel $K_t = \int \psi(x_t)\psi(x_t)^\top \mu(ds_{t+1})$, and uniform local geometry bound from Assumption 4.4.

In the **exponential family setting**, $h(\cdot), \psi(\cdot), \phi(\cdot)$ are known feature functions and $A_1, \ldots, A_d$ are known matrices. $\bar{Z}(\theta, s_t, a_t)$ is the log-partition function, and A denotes the trace-norm matrix induced by the matrices A_i. The main assumptions are strict convexity of $Z_{s,a}$, bounded covariance curvature, bounded feature-matrix scale, and a bounded true parameter norm $\|\theta^\star\|_A \le B_A$.

Both settings study parameter estimations via MLE. The parameter space used in the squared family setting is constrained to a statistical manifold to remove their singular and non-identifiable nature and permit mathematical analysis. This is similar to the explicit regularisation term used by Chowdhury to control the parameter of their exponential setting. Assumption 4.4 also plays the role of ensuring the squared kernel does not collapse. Specifically, by enforcing

$$c_\epsilon \mathcal{K}_t \preceq \mathcal{K}_t^\epsilon,$$

we ensure

$$\mathrm{SH}({}^\epsilon P_{\theta,t} : {}^\epsilon P_{\theta',t}) \ge \frac{c_\epsilon}{2} ||\theta - \theta'||^2_{\mathcal{K}_t}.$$

It says that the statistical distance grows at least quadratically in the geometry induced by the squared kernel. This is akin to the strong convexity assumption used on the log partition function in the exponential family setting. Without these setups, analysis can be quite challenging in both settings. In the squared family setting, the MLE is defined over the parameter space without further regularisation whereas in the exponential family setting, the MLE is regularised by the matrix norm of the parameter. Finally, the parameter concentration results are given in terms of their natural geometries.

More on Assumption 4.4. Careful readers may ask whether this assumption is valid. Here we provide a straight forward example. For any s, a, s' and $\theta \in \mathrm{conv}\Omega$, when $|\psi^T \theta| > \rho$ for some $\rho > 0$, then

$$|g'(\psi^\top \theta)|^2 \ge \frac{(1-\epsilon)^2 \rho^2}{(1-\epsilon)\rho^2 + \epsilon} := c_\epsilon.$$

In other words, under the condition that the linear statistics does not vanish, we obtain the desired c_ϵ lower bound. This setting brings a even better advantage, since ${}^\epsilon p_{\theta,t} \ge (1-\epsilon)\rho^2 + \epsilon$, then Lemma C.3 gives a better Lipschitz constant $L_{\log} := \frac{2c_\epsilon B^2 R}{(1-\epsilon)\rho^2}$. This constant no longer explodes when ϵ gets infinitesimally close to zero, which makes our eventual regret bound even shaper.

Comparison of proof techniques. Both analyses follow the same high-level model-based RL template: estimate the transition parameter by MLE, construct a high-probability confidence set, choose an optimistic model-policy pair, and convert transition estimation error into regret. The key technical difference is the concentration mechanism. In the exponential family setting, the proof exploits the fact that the KL divergence is the Bregman divergence of the log-partition function. Chowdhury et. al. construct an exponential martingale for fixed directions and then integrate these martingales using a Gaussian prior, yielding a Laplace-mixture concentration inequality for conditional exponential families. A direct application of Markov's inequality and stopped martingale control yields the concentration bound. In the squared family setting, the proof uses a one-sided likelihood-ratio martingale based on Hellinger affinity so that $M_n(\theta)$ is a nonnegative martingale. Ville's inequality applied on $M_n(\theta)$ and a finite α-net argument then give a uniform Squared Hellinger confidence bound, which is transferred back to the MLE using log-likelihood Lipschitzness and Hellinger continuity properties of the squared family.

Since the concentration results are defined in their own natural geometry. Chowdhury et.al. show that the regret can be upper bounded by a sum of KL divergences using transportation inequalities. Each individual KL is then bounded by carrying out elliptical potentially analysis around the matrices A_i combined with the previous concentration results of the exponential family setting. On the other hand, the squared family settings use simulation lemma instead of transportation inequalities to decompose the regret into sum of squared Hellinger distances. These distances can be upper bounded via careful elliptical potential analysis around the squared kernel K_t and the concentration bound is substituted into the analysis to derive the final regret.

Comparison of regret bounds. The resulting regret rates are of the same general form: both are sublinear in the total number of interaction steps and polynomial in the horizon and dimension. For squared-family MDPs, one has

$$\mathrm{Regret}(K) = \widetilde{O}\Big(\max(d, \sqrt{dH})H\sqrt{T}\Big), \qquad T = KH.$$

By comparison, Chowdhury et al. obtain the exponential-family regret bound

$$R(N) = \widetilde{O}\Big(d\sqrt{H^3 N}\Big) = \widetilde{O}\Big(dH^{3/2}\sqrt{T}\Big)\,,$$

where N is the total number of steps. In either case of $H \leq d$ or $d > H$, the squared-family rate is mildly sharper than the exponential family setting. This however does not necessarily mean squared family is always better suited for RL than exponential family, because the restrictions on the model classes are not identical. The exponential-family result applies to a broad bilinear exponential model and relies on KL/Bregman geometry, while the squared-family result exploits the special quadratic structure of the smoothed squared density, which yields a direct squared-Hellinger quadratic form and enables a sharper Hellinger-based confidence radius. The implication is that squared-family MDPs can match the likelihood-based optimism philosophy of exponential-family MDPs while retaining tractable normalisation and a potentially sharper horizon dependence under the stated assumptions. For a comparison of their empirical aspects, see Appendix A.

E Extra Experimental Details

E.1 Implementation Details

We use Gaussian mixtures (with 8 mixture components) as the base measure $\mu(\cdot)$, Snake$_a$ activation (Eq.13) as the activation $\sigma(\cdot)$ of the ξ feature and the identity function for the sufficient statistics $t(\cdot)$. We parameterise feature $Y(\cdot,\cdot)$ with residual blocks. Specifically, we use two separate perceptron layers to process the incoming state and action inputs to obtain two 128 dimensional embeddings and then concatenate the embeddings as inputs for the residual blocks. The output is a flattened vector in $\mathbb{R}^{nm}$ and is then viewed as a matrix in $\mathbb{R}^{m\times n}$. The learnt feature ϕ is flattened and passed to the critic as an latent representation.

During training, we found it is helpful to allow gradients to flow through the critic back to feature ϕ, such that the parameters of ϕ is trained jointly by three objectives, the model MLE, the reward MSE and the critic TD-error. Interestingly, this is not the case in other representation methods, like DiffSR (Shribak et al., 2024), as when we allow gradient information to flow into diffusion objectives, the performance of DiffSR drops. This is likely because of the singularity property of the Squared family, that is the same density can be represented by multiple parameters. Allowing critic information to flow back to our model enable the algorithm to find a better set of parameters for optimisation purposes.

Similar to a target Q-network, we also found helpful to use a target SNEFY model to stabilise the training. Since only the features from the target SNEFY model is used for the target Q-network, we never needs to evaluate the Squared kernel of the target model, which is another benefit gained from the parameter integral factorisation property of the Squared family. We found setting $n = 512$ and $m = 8$ good for both Gym-Mujoco and DMC tasks, as they balance well between having rich feature representations and computational costs.

Many works (Agarwal et al., 2020; Uehara et al., 2021; Guo et al., 2023) that leverage low rank MDP properties often design an exploration bonus to explicitly maintain a good balance between exploration and exploitations. Since we build our practical algorithm based on the actor critic methods, the exploration is handled well by the method, therefore, dropping the requirement of designing such bonus terms.

We found many existing benchmarking results are outdated (Tassa et al., 2018; Wang et al., 2019), as Gym-Mujoco and DMC packages have gone through several major updates, hence we choose to run all baselines from scratching using the newest software instead of re-using the old results.

We use the official implementation provided by the baseline methods in all our experiments.

E.2 Extra Experiment Results

We discuss extra experimental details and results in this section, including wall clock training times, more training graphs, ablation studies and hyperparameters.

E.2.1 Wall Clock Training Time

The wall clock training times of all the methods in the half cheetah task over 10 steps are displayed in Fig. 3. We can see by freezing the θ_2 parameter and hence re-use the same squared kernel, RESNEF-RF saves 34% of training time from the original RESNEF. It however still performs bit slower when compared to DIFFSR, this is because of the matrix multiplications involved in operating the matrices Y_{θ_1} and $\tilde{\mathcal{K}}_\xi$. The training times also depicts a no free lunch phenomenon between computational complexity and sample efficiency. In order to use less sample to achieve better performances, extra computations are required to extract meaningful information from limited data.

E.2.2 Extra Training Graphs

More training graphs for Gymnasium and DMC are displayed in Figure 4 and Figure 5. Inverted Double Pendulum is the simplest task and all methods perform equally well. Representational methods all perform better than SAC in Reacher. For the two hardest control task, RESNEF-RF is the second best method after DiffSR that can achieve high performance in Ant and it is the

Figure 3: Wall Clock Training Time in Half Cheetah Across 10 Steps

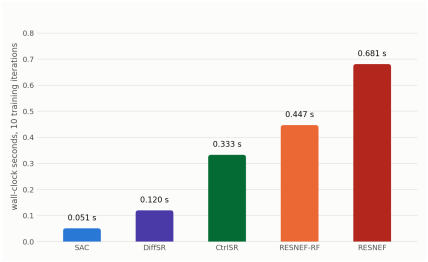


Figure 4: More training curves of Gym-Mujoco tasks. X-axis is the training steps and y-axis is the averaged evaluation rewards across 4 random seeds. Solid lines represent average performances and shaded areas represent standard errors.

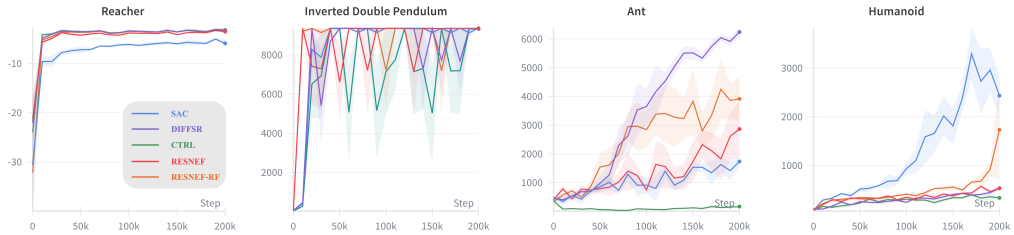
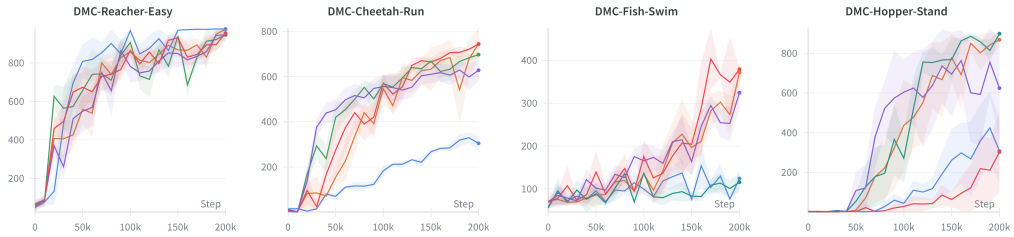


Figure 5: More training curves of Deepmind Control Suite tasks.



only representational method that can find good representations early to control Humaniod. It also performs really well in Fish-Swim and Hopper-Stand tasks from DMC.

E.3 Ablations

To isolate the effect of MLE and whether it indeed bring sample efficiency when compared to other estimation method, we compare standard RESNEF against a variant that uses score matching. The results across 3 tasks from Gymnasium with 4 random seeds are shown in Fig. 6. In all experiments, MLE outperforms score matching by a clear margin. In particular, we see that MLE brings larger gains from very early on in these cases, bridging the gap between our theory and practice.

We also conduct ablation studies on the feature dimensions n and m. The study uses RESNEF-RF as it is more efficient than RESNEF in terms of training time. We found out that setting $n = 512$ and $m = 4$ works well for more complex environments like hopper, walker and ant. For simpler dynamics like HalfCheetah, higher dimensions may lead to overfitting and a smaller dimensions works well (e.g. $n = 256$ and $m = 4$). In particular, we found that putting enough ranks into the model general improve performance, this is as the rank reduces, the model increasingly approach a mixture model (setting $m = 1$ recovers a standard mixture model), which do not have strong universal approximation properties as a higher ranked SNEFY model.

Figure 6: Ablation studies on MLE vs score matching (SM) for RESNEF.

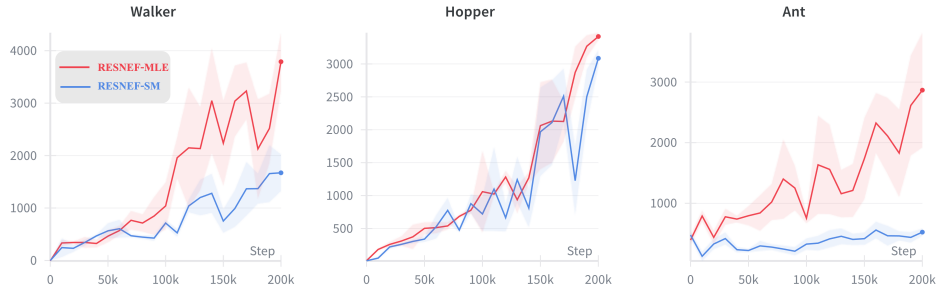
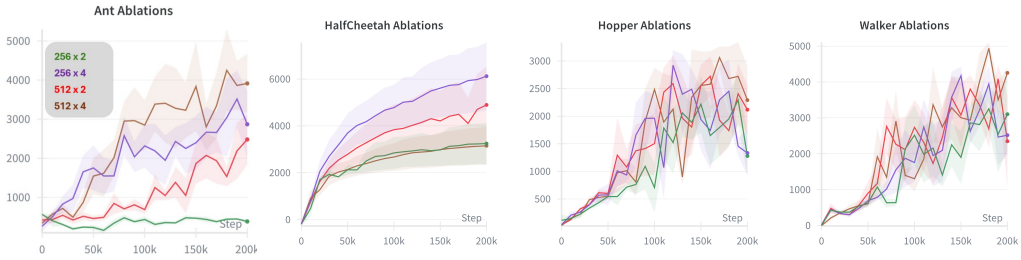


Figure 7: Ablation studies on the feature dimensions n and m of RESNEF-RF across 4 Gymnasium control tasks.



E.4 HYPERPARAMETERS

Hyperparameters used in the experiments are displayed in Table. 4.

Table 4: Hyperparameters of RESNEF.

Hyperparameters	**Values**
Model learning rate	0.0001
Actor learning rate	0.0003
Critic learning rate	0.0003
Actor hidden dimensions	(512, 512)
Critic hidden dimensions	(512,)
n	512
m	8 (4 for Gymnasium)
σ	Snake_a
$\tilde{\sigma}$	ReLU
t	Id
μ	Gaussian mixtures
Y hidden dimensions	(512,)
Y embed dimensions	256
Discount factor γ	0.99
Target critic update factor	0.005
Target model update factor	0.01