Mitigating Overgeneralization in RND via Spectral Target Design

Minseok Jeong[1,*] **Yechan Lee**[1,*] **Hyewon Choi**[1]
Jeongyong Yang[2] **SooJean Han**[1,†]
[1]KAIST, [2]University of Washington

Abstract

Random Network Distillation (RND) is a scalable novelty signal for RL based on a fixed random target network and a trained predictor. However, it can overgeneralize: the predictor extrapolates the random target off the data manifold, causing novelty scores to collapse on unfamiliar inputs. This paper analyzes this failure through a kernel-theoretic lens. Under a GP target model and an NTK-regime predictor, we show that the RND novelty score is a *cross-kernel residual variance*: it is governed jointly by the *target covariance kernel* and the *predictor NTK*, rather than by either one alone. This identifies overgeneralization as spectral under-excitation. Motivated by this view, we formulate spectral target design as a principle for shaping RND novelty geometry. As a concrete instance, we replace the implicit random-network target with a bandwidth-controlled Random Fourier Feature (RFF) target, redistributing residual signal toward modes less attenuated by the predictor. Experiments on offline D4RL, online Atari, and Fetch manipulation benchmarks show improved novelty discrimination and downstream RL performance. The code for the RL implementation is available at https://github.com/tuuktuc86/RFF-RND.

1 Introduction

Random Network Distillation (RND) [Burda et al., 2019b] is a simple yet effective novelty signal for reinforcement learning (RL), using the prediction error of a trained network against a fixed random target to measure familiarity. Its scalability and ease of integration have made it a standard component in modern RL pipelines, driving exploration in online settings [Burda et al., 2019a] and enforcing conservatism in offline ones [Rezaeifar et al., 2022, Nikulin et al., 2023].

Despite its widespread adoption, RND suffers from a fundamental failure mode known as *overgeneralization*: the predictor can extrapolate beyond the data manifold, assigning low novelty scores to inputs beyond the data coverage radius that should be flagged as novel [Belkin et al., 2019]. This undermines the intrinsic reward in both settings — misallocating exploration bonuses toward already-familiar regions and failing to penalize out-of-support actions sufficiently. Prior work has addressed this through distributional extensions [Yang et al., 2024] or architectural modifications [Nikulin et al., 2023], but these approaches do not explicitly characterize *why* overgeneralization occurs or provide principled control over the novelty signal's geometric structure. This raises a natural question:

What governs the discriminative power of RND, and how can it be reliably controlled?

In this work, we answer this question through a kernel-theoretic analysis [Rasmussen and Williams, 2006, Schölkopf and Smola, 2002]. We show that overgeneralization arises not from the predictor or target network in isolation, but from their interaction: when the predictor extrapolates the target accurately beyond the data manifold, the novelty score collapses off-manifold. Motivated by this analysis, we propose a spectral target-design principle for RND.

[*]Co-first authors. See Appendix A for individual contributions. [†]Corresponding author.

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

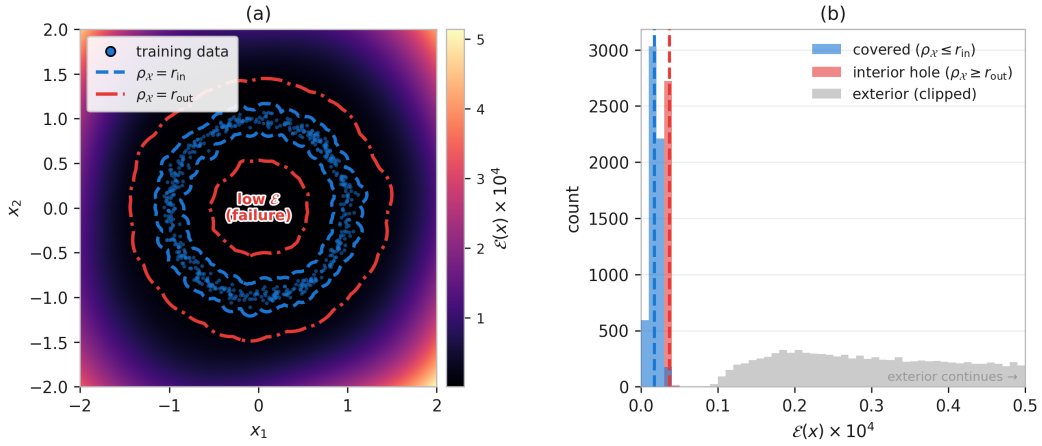


Figure 1: **Coverage-based diagnostic for standard RND on $\mathbb{S}^1$ data.** (a) RND energy heatmap with contours defined by $\rho_{\mathcal{D}}(x) = \min_{x_i \in \mathcal{D}} d_{\mathcal{X}}(x, x_i)$. The blue dashed contour denotes $\rho_{\mathcal{D}} = r_{\text{in}}$, and the red dash-dotted contour denotes $\rho_{\mathcal{D}} = r_{\text{out}}$. The interior hole ("low $\mathcal{E}$ (failure)") lies beyond the coverage radius but receives low RND energy. (b) Energy distributions by region. The covered band (blue, $\rho_{\mathcal{D}} \le r_{\text{in}}$) and the interior hole (red, $\rho_{\mathcal{D}} \ge r_{\text{out}}$) have overlapping low-energy distributions, illustrating a failure of coverage-based residual separation. The exterior region (gray) is shown only for reference. Energies are scaled by 10^4 for visualization since the trained RND energies are small.

Specifically, directly aligning the target with hard-to-predict modes is impractical, since the predictor spectrum is implicit, architecture-dependent, and generally unknown before training. We therefore seek predictor-agnostic target families whose spectra can explicitly control the kernel geometry, and are lightweight to sample. Random Fourier Features (RFF) [Rahimi and Recht, 2007, 2008, Le et al., 2013] provide one such concrete example, replacing the random MLP target with a target whose spectral density and coordinate-wise independence are directly specified. This method provides an explicit control knob for shaping the novelty scale without changing the downstream RL algorithm.

The rest of the paper is organized as follows. Section 2 defines the coverage-based failure mode we want RND to avoid. Section 3 analyzes why the residual can collapse by expressing RND energy as a cross-kernel residual variance. Section 4 turns this mechanism into a target-design principle, introduces `RFF-RND`, and provides a frequency-domain interpretation of the same mechanism. Section 5 diagnoses our theory and empirically evaluates our methods on downstream RL tasks.

2 Problem Setup and Motivation

2.1 RND for Novelty Detection

Standard RND consists of a fixed target network $f_t : \mathcal{X} \to \mathbb{R}^m$ and a trainable predictor network $f_p : \mathcal{X} \to \mathbb{R}^m$, both implemented as multi-layer perceptrons (MLPs) with random initialization. Let $x \in \mathcal{X} \subset \mathbb{R}^d$ denote an input datapoint, and $\mathcal{D} = \{x_i\}_{i=1}^n$ denote the dataset. The predictor $f_p(\cdot; \theta)$ is trained to match the output of the target network by minimizing the squared error on $\mathcal{D}$:

$$\min_{\theta} \sum_{i=1}^{n} \mathcal{E}(x_i; \theta), \tag{1}$$

where the RND energy $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ is defined as $\mathcal{E}(x_i; \theta) \triangleq \frac{1}{m} \left\| f_t(x_i) - f_p(x_i; \theta) \right\|_2^2$. When the context is clear, we write $\mathcal{E}(x) = \mathcal{E}(x; \theta)$, $f_p(x) = f_p(x; \theta)$ for brevity, i.e., the RND novelty score at a query point x is $\mathcal{E}(x) = \frac{1}{m} \| f_t(x) - f_p(x) \|_2^2$.

Although standard RND assigns high energy to some exterior regions, it assigns low energy to the region that is poorly covered by the observed data. Thus, reliable novelty detection requires a coverage-based criterion rather than merely assigning high energy somewhere off the data manifold.

2.2 A Coverage-Based Separation Criterion

A useful RND score should separate well-covered inputs from poorly-covered ones, i.e., the target should be predictable near visited samples but difficult to extrapolate beyond the data coverage radius. Let $d_{\mathcal{X}}$ be a task-relevant metric or representation-induced distance on $\mathcal{X}$. We formalize our desired novelty behavior as follows.

Definition 2.1 (Coverage-based residual separation)**.** Let $\rho_{\mathcal{D}}(x) \triangleq \min_{x_i \in \mathcal{D}} d_{\mathcal{X}}(x, x_i)$. We say that RND satisfies *coverage-based residual separation* if there exist $0 \le r_{\text{in}} < r_{\text{out}}$ and $0 \le \epsilon < \gamma$ such that

$$\rho_{\mathcal{D}}(x) \le r_{\text{in}} \;\Rightarrow\; \mathcal{E}(x) \le \epsilon, \qquad \rho_{\mathcal{D}}(x) \ge r_{\text{out}} \;\Rightarrow\; \mathcal{E}(x) \ge \gamma.$$

We use this criterion as a diagnostic notion rather than a universal guarantee: both the metric $d_{\mathcal{X}}$ and the relevant coverage parameters $r_{\text{in}}, r_{\text{out}}, \epsilon, \gamma$ are task-dependent. In Section 4, we establish a spectral analogue in an idealized setting. Figure 1 intuitively illustrates this coverage-based failure mode on an $\mathbb{S}^1$ toy dataset.

2.3 RND Variants through the Lens of Kernel Geometry

We next situate this coverage-based failure mode relative to existing RND variants.

Predictor- and target-focused RND variants. Many RND variants implicitly modify one side of the kernel geometry. Conditional RND [Nikulin et al., 2023] conditions the predictor via FiLM-style modulation [de Vries et al., 2017, Perez et al., 2018, Dumoulin et al., 2018], inducing a condition-dependent feature map that deforms the resulting geometry. In contrast, Distributional RND (DRND) [Yang et al., 2024] acts on the target side. It replaces the prediction error induced by a single target network with the ensemble variance induced by multiple target networks, thereby capturing uncertainty across random target draws. Although these approaches showed favorable performance improvements, they did not identify the underlying cause.

Why cross-kernel geometry matters. These variants suggest that *both* sides of the RND residual matter: the predictor geometry determines how target values are interpolated, while the target geometry determines what variation is available to be detected. In the following sections, we make this interaction explicit by characterizing RND energy as a cross-kernel residual variance.

3 RND Energy as a cross-kernel residual variance

We analyze RND under two idealizations: a GP model for the random target and an NTK approximation for the predictor.

Assumption 3.1 (GP target model)**.** We model each output coordinate of f_t as an independent GP:

$$f_t^{(j)} \sim \mathcal{GP}(0, k_t), \quad j = 1, \dots, m,$$

with output coordinates mutually independent.

Assumption 3.1 is a natural infinite-width idealization of standard RND: a randomly initialized MLP converges to a GP with the associated NNGP kernel [Neal, 1996, Lee et al., 2018, Matthews et al., 2018]. A key consequence of this is *joint Gaussianity*, that for any query x and dataset inputs X, the collection $\{f_t(x), f_t(x_1), \dots, f_t(x_n)\}$ is jointly Gaussian with cross-covariances

$$\mathbb{E}[f_t(x) f_t(x_i)^\top] = k_t(x, x_i)\, I_m.$$

Equivalently, for any inputs $x, x' \in \mathcal{X}$, $\mathbb{E}[f_t(x) f_t(x')^\top] = k_t(x, x')\, I_m$. The induced covariance on the dataset is $K_t = k_t(X, X) \in \mathbb{R}^{n \times n}$.

Assumption 3.2 (NTK predictor model)**.** The predictor is analyzed in the lazy-training regime [Chizat et al., 2019, Lee et al., 2019]. We use a decoupled multi-output NTK approximation, where the squared loss separates across the m output coordinates and each coordinate is governed by the same scalar NTK Θ_p [Jacot et al., 2018, Du et al., 2019, Allen-Zhu et al., 2019, Arora et al., 2019]. [0]

[0] A fully vector-valued NTK would replace Θ_p with an $nm \times nm$ block kernel over input–output coordinate pairs. We use the decoupled form only to avoid unnecessary notation: the same interpolation argument applies coordinate-wise, while the analysis below focuses on the induced dependence on the target matrix F_X.

Under gradient flow with ℓ_2 regularization, this yields kernel ridge regression with respect to Θ_p:

$$\Theta_p = \Theta_p(X, X) \in \mathbb{R}^{n\times n}, \quad \Theta_p(x, X) \in \mathbb{R}^{1\times n}.$$

Define the target matrix $F_X = [f_t(x_1), \ldots, f_t(x_n)]^\top \in \mathbb{R}^{n\times m}$. The limiting predictor at a query point x is $f_p^\infty(x) \triangleq F_X^\top(\Theta_p + \lambda I)^{-1}\Theta_p(x, X)^\top \in \mathbb{R}^m$.[1] For brevity, define

$$A_x \triangleq \Theta_p(x, X)(\Theta_p + \lambda I)^{-1} \in \mathbb{R}^{1\times n},$$

so that $f_p^\infty(x) = F_X^\top A_x^\top$.

Although Assumption 3.2 is idealized, it provides a natural approximation for RND. Because the predictor fits a fixed random target, the supervision signal contains no coherent task-level structure to be extracted through representation learning. So, that predictor only requires adapting the target kernel. A similar perspective has been studied in recent work connecting RND to ensemble and Bayesian interpretations in the infinite-width regime [Zanger et al., 2026]. However, our focus is complementary: we use the NTK view to diagnose why RND can fail as a novelty estimator. We begin by defining the cross-kernel residual variance.

Definition 3.3 (cross-kernel residual variance)**.** Given target kernel k_t and predictor NTK Θ_p, the *cross-kernel residual variance* at x is

$$\sigma^2_{\text{cross}}(x;\, k_t, \Theta_p) \;\triangleq\; k_t(x, x) - 2A_x\, k_t(X, x) + A_x K_t A_x^\top.$$

We next define the finite-output RND energy under the NTK predictor, which surrogates $\mathcal{E}(x)$, by

$$\mathcal{E}_m(x) \triangleq \frac{1}{m}\|f_t(x) - f_p^\infty(x)\|_2^2.$$

This leads to the following characterization of its infinite-output limit.

Theorem 3.4 (Infinite-dimensional GP target RND energy)**.** *Under the GP target model and NTK-regime predictor,*

$$\mathcal{E}_m(x) \xrightarrow[m\to\infty]{a.s.} \sigma^2_{\text{cross}}(x; k_t, \Theta_p).$$

The proof is provided in Appendix B.1. Note that, although our proof exploits the Gaussianity of the target GP, this assumption is stronger than necessary. In fact, the energy formula depends only on the second-order covariance structure k_t of the target process. In Section 4, we show that RFF targets implement the same covariance structure in expectation, despite not being GP sample paths.

Theorem 3.4 is the key point of our analysis. It shows that, in the infinite-output GP/NTK limit, the RND energy is cross-kernel residual variance, i.e., $\mathcal{E}_\infty(x) = \sigma^2_{\text{cross}}(x; k_t, \Theta_p)$. The target kernel k_t determines the covariance of the random target values, whereas the predictor NTK Θ_p determines the interpolation weights A_x. These weights specify how the predictor combines the observed target values F_X to estimate the target at a query point x. Hence the RND score is governed by the interaction between target variation and predictor interpolation, rather than by either kernel alone.

4 Spectral Target Design

We now use this view to derive a target-design principle. The key idea is to treat the target covariance as a spectral design object: rather than inheriting an uncontrolled spectrum from a random network, we seek a target family with an explicit knob for shaping RND energy landscapes.

4.1 Residual Collapse as Spectral Under-Excitation

We analyze the induced training residual $R_X = F_X - \widehat{F}_X, \widehat{F}_X = [f_p^\infty(x_1), \ldots, f_p^\infty(x_n)]^\top$ whose $\frac{1}{m}\operatorname{tr}(R_X R_X^\top)$ is the sum of RND energies over the training set.

Lemma 4.1 (Spectral residual decomposition)**.** *Consider the spectral decomposition $\Theta_p = U_p\Lambda_p U_p^\top$ with $\Lambda_p = \operatorname{diag}(\lambda_{p,1}, \ldots, \lambda_{p,n})$ and let u_i denote the i-th column of U_p. The training residual satisfies $R_X = \lambda(\Theta_p + \lambda I)^{-1}F_X$, and $\mathbb{E}\|u_i^\top R_X\|_2^2 = m\Big(\frac{\lambda}{\lambda_{p,i}+\lambda}\Big)^2 u_i^\top K_t\, u_i$.*

[1] Standard RND training does not include explicit ridge regularization. Here, $\lambda > 0$ is introduced to ensure that $\Theta_p + \lambda I$ is invertible; the limit $\lambda \to 0^+$ recovers the minimum-norm interpolant. We keep $\lambda > 0$ throughout for analytical clarity.

A detailed proof is provided in Appendix B.2. Lemma 4.1 shows that the regularized residual on the training set decomposes naturally in the eigenbasis of the predictor NTK Θ_p. In addition, since K_t governs the target covariance globally, the same spectral structure determines residual behavior at query points beyond the training set. We can further decompose the training-set residual energy along each predictor NTK eigenmode into a predictor-side and a target-side term.

$$\text{Predictor's mode-}u_i\text{ contribution to RND energy} \propto \underbrace{\left(\frac{\lambda}{\lambda_{p,i}+\lambda}\right)^2}_{\text{predictor residual filter}} \times \underbrace{u_i^\top K_t u_i}_{\text{target variance in mode } u_i}$$

If u_i is a predictor-easy mode ($\lambda_{p,i} \gg \lambda$), the residual filter nearly vanishes, so even a large value of $u_i^\top K_t u_i$ contributes little to the residual. Conversely, for predictor-hard modes ($\lambda_{p,i} \ll \lambda$), the filter remains large; however, when $u_i^\top K_t u_i$ is small, the target contains little variation along that mode, leaving little residual signal. In this terminology, overgeneralization arises from predictor-hard modes with insufficient target variation. This motivates framing target design as a mode-wise spectral allocation problem, which we formalize in the following definition.

Definition 4.2 (Target power and residual contribution)**.** Let $\Theta_p = U_p \Lambda_p U_p^\top$, with eigenvectors $\{u_i\}_{i=1}^n$ and eigenvalues $\{\lambda_{p,i}\}_{i=1}^n$. For a target Gram matrix K_t, define the *target power* assigned to predictor mode u_i by $q_i(K_t) \triangleq u_i^\top K_t u_i$. For ridge parameter $\lambda > 0$, define the *residual contribution* of mode u_i by $r_i(K_t) \triangleq \left(\frac{\lambda}{\lambda_{p,i}+\lambda}\right)^2 q_i(K_t)$.

In standard RND, the target and predictor networks typically share the same architecture, with their parameters independently initialized from the same distribution. Consequently, their associated kernels are expected to exhibit similar spectral densities.

4.2 RFF-RND: Target Design as Spectral Allocation

So far, we have analyzed how the RND energy, particularly its spectral properties, is characterized and determined by the interplay between the predictor and target networks. However, the preceding decomposition suggests an ideal but impractical allocation rule: choose K_t so that $q_i(K_t)$ is not concentrated only on predictor-easy modes, but also assigns power to modes where the residual filter is non-negligible. In general, directly optimizing K_t in the predictor eigenbasis is not practical in standard RND, since Θ_p, its eigenvectors, and its spectrum are implicit and architecture-dependent. Therefore, a natural extension is to make the target structure explicit in a *predictor-agnostic* form, with a tunable spectrum for generality rather than inheriting an uncontrolled spectrum from a randomly initialized MLP.

Moreover, the preceding analysis only depends on the *target covariance structure*. This motivates replacing the target family with a shift-invariant ***RFF target*** (which we call `RFF-RND`), whose spectral density can be controlled directly.

Corollary 4.3 (Infinite-output RFF target RND energy)**.** *For the m-dimensional RFF target $f_t^{(j)}(x) = \sqrt{2}\cos(\omega_j^\top x + b_j)$, $\omega_j \sim p_\alpha$[2], $b_j \sim \mathrm{Unif}[0, 2\pi]$, with independent draws across $j \in \{1, \ldots, m\}$, the target coordinates are zero mean and satisfy $\mathbb{E}_{\omega,b}\left[f_t^{(j)}(x) f_t^{(j)}(x')\right] = k_t^\alpha(x - x')$. Therefore, under the same NTK-regime predictor, $\mathcal{E}_m(x) \xrightarrow[m\to\infty]{a.s.} \sigma^2_{\mathrm{cross}}(x; k_t^\alpha, \Theta_p)$.*

The proof is provided in Appendix B.3. As briefly mentioned earlier, here we exploit the fact that it suffices to match only the second-order moments encoded by the target network's Gram matrix, without requiring Gaussianity of the target sample paths. For an RFF target, the corresponding covariance is K_t^α, whose bandwidth α controls the spectral spread of the target kernel.

Therefore, we can easily compare target designs by examining how K_t^α allocates power across the empirical predictor NTK eigenbasis, yielding a substantially more flexible target design.

[2]Here, we assume p_α is symmetric, i.e., $p_\alpha(\omega) = p_\alpha(-\omega)$. For the RBF target kernel, $p_\alpha(\omega) = \mathcal{N}(0, \alpha^{-2} I_d)$ and $k_t^\alpha(x - x') = \exp(-\|x - x'\|_2^2/(2\alpha^2))$. Other variants of shift-invariant kernels are summarized in Appendix C.

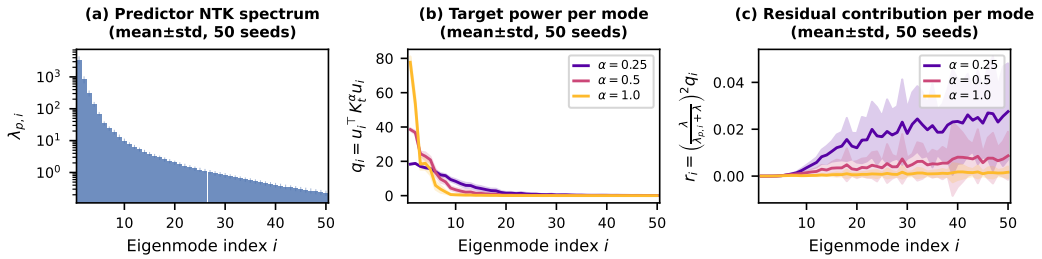


Figure 2: **Spectral allocation of RBF-RFF targets across predictor NTK modes.** (a) Predictor NTK eigenvalues $\lambda_{p,i}$, sorted in descending order. (b) Target power $q_i(K_t^\alpha) = u_i^\top K_t^\alpha u_i$ along predictor NTK eigenmodes. (c) Mode-wise residual contribution $r_i(K_t^\alpha) = (\lambda/(\lambda_{p,i} + \lambda))^2 \, q_i(K_t^\alpha)$.

Figure 2 illustrates this behavior empirically. Consider an RND setup with a large-width MLP predictor and an RFF target. As expected, the predictor NTK spectrum is concentrated on leading modes, so the residual filter strongly suppresses these modes while leaving higher-index modes less attenuated. The RFF bandwidth controls how target power, and consequently residual contribution, is allocated across this basis. Larger bandwidths concentrate power on easy leading modes ($\alpha = 1$), whereas smaller bandwidths spread power toward less-attenuated higher-index modes ($\alpha = 0.25$). Specifically, Figure 2-(c) shows that smaller RFF bandwidths concentrate residual contribution on high-index NTK modes, where small predictor eigenvalues imply weak interpolation.

Remark 4.4 (Pretrained feature map). Pretrained features can also define a target kernel $K_{\mathrm{pre}}(i,j) = \phi_{\mathrm{pre}}(x_i)^\top \phi_{\mathrm{pre}}(x_j)$. However, the mode-wise power $u_i^\top K_{\mathrm{pre}} u_i$ is inherited from pretraining and is not directly tunable for novelty detection. This is useful when the pretrained invariances match the desired coverage, but unlike RFF targets, it does not provide an explicit novelty-scale parameter.

4.3 Frequency-Domain Interpretation of Residual Separation

The finite-sample NTK decomposition above is general, but its modes are the empirical eigenvectors of Θ_p, which need not have an immediate frequency interpretation. To make the spectral mechanism explicit, we consider a clean idealized setting where this eigenbasis is known. On a uniform periodic grid, shift-invariant predictor and target kernels are diagonalized by the Fourier basis. In this case, the general predictor-mode decomposition reduces to a frequency-wise decomposition.

Proposition 4.5 (Frequency-wise residual decomposition)**.** *On a uniform periodic grid, suppose Θ_p and k_t^α are shift-invariant, with Fourier eigenvalues $\widehat{\theta}_p(\omega)$ and $\widehat{k}_t^\alpha(\omega)$. Then*

$$r_\alpha(\omega) \triangleq \frac{1}{m}\mathbb{E}\|u_\omega^* R_X\|_2^2 = \left(\frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda}\right)^2 \widehat{k}_t^\alpha(\omega).$$

The proof is given in Appendix B.4. Proposition 4.5 is the frequency-domain counterpart of the residual decomposition in Lemma 4.1. It shows that the residual at frequency ω factors into a predictor-side spectral residual filter, $(\lambda/(\widehat{\theta}_p(\omega) + \lambda))^2$, and a target-side spectral power term, $\widehat{k}_t^\alpha(\omega)$. This suggests that target should allocate power *broadly enough* to cover frequencies that are weakly interpolated by the predictor, rather than concentrating power on a narrow, easily interpolated band.

The same mechanism can also be understood intuitively in the spatial domain. For RBF-RFF targets, smaller α makes $k_t^\alpha(x,x') = \exp(-\|x - x'\|_2^2/(2\alpha)^2)$ decay more rapidly with distance, so target values far from $\mathcal{D}$ are harder to infer from observed dataset values. More explicitly, Corollary 4.1 gives

$$\mathcal{E}_\infty(x) = \sigma^2_{\mathrm{cross}}(x;\, k_t^\alpha, \Theta_p) = 1 - 2A_x k_t^\alpha(X,x) + A_x K_t^\alpha A_x^\top.$$

For out-of-coverage points, the cross-covariance term $A_x k_t^\alpha(X,x)$ is driven toward zero, while the quadratic term $A_x K_t^\alpha A_x^\top$ can remain non-negligible even when A_x is small. The resulting residual increase induces the coverage-based separation in Definition 2.1, consistent with the empirical diagnostic in Figure 3.

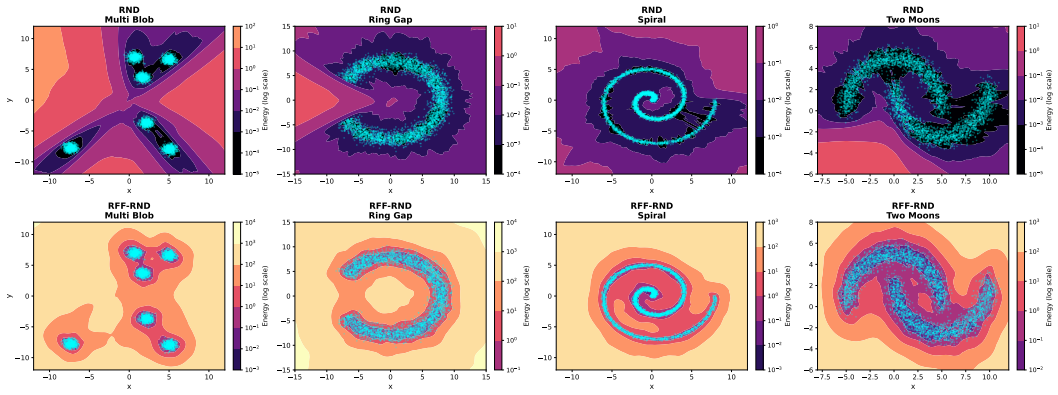


Figure 3: **Energy landscapes** ($\alpha = 0.75$**,** $m = 64$**).** Top: standard RND exhibits overly confident off-manifold energy patterns. Bottom: `RFF-RND` produces more consistent landscapes with clearer separation between the data manifold and off-manifold regions.

5 Experiments

5.1 Diagnostics and Ablations for `RFF-RND`

We provide diagnostics and ablations for `RFF-RND`, validating the kernel-level predictions and examining the effects of α, m, and predictor network width h on the induced novelty geometry. To keep the main text focused on the empirical narrative, we summarize these diagnostic results here and defer the full ablations to the appendices.

Finite-output concentration. Appendix D.1 verifies the finite-output concentration predicted by Corollary 4.3. We observed that as the RFF output dimension m increases, the empirical energy $\mathcal{E}_m(x)$ aligns more closely with the kernel-level prediction $\sigma^2_{\text{cross}}(x; k_t^\alpha, \Theta_p)$. On the Two Moons diagnostic, Pearson agreement increases from about $r = 0.65$ at $m = 4$ to about $r = 0.97$ at $m = 64$, suggesting that moderate output dimensions already provide reliable kernel-level estimates.

Frequency-domain diagnostic. Appendix D.2 visualizes the frequency-wise decomposition in Proposition 4.5. This diagnostic directly checks the product structure of the residual, rather than downstream performance. We observe the RFF bandwidth controls the target spectrum, while the predictor NTK induces a residual filter that suppresses frequencies it interpolates well. The observed residual power appears where target spectral power overlaps with frequencies that pass this filter, supporting bandwidth as a spectral knob for shaping the RND residual.

Energy-landscape ablations. Appendix D.3 studies how the RFF bandwidth α, output dimension m, and predictor width h affect the energy landscape. The bandwidth α controls the spatial scale of novelty: small values produce localized but noisy landscapes, while large values over-smooth the energy and blur nearby manifold boundaries. Increasing m stabilizes the finite-output energy, whereas varying h mainly changes the smoothness of the fitted contours without altering the overall manifold-aware separation. Together, these ablations clarify the roles of the main design parameters in our diagnostics: α changes the spatial scale of the energy landscape, m improves finite-output stability, and h mainly affects the smoothness of the fitted contours over the tested range.

5.2 Additional Comparisons with RND Variants

Appendix E compares `RFF-RND` with DRND and input-conditioned RND variants on synthetic manifolds, including computational and memory costs. DRND improves novelty contrast through a distributional correction over N frozen target networks, while input-conditioned RND reshapes the energy landscape through gating, bilinear, or FiLM-style modulation. However, their novelty geometry remains implicit and architecture-dependent, and both introduce additional overhead. In contrast, `RFF-RND` replaces the random target network with fixed RFF parameters $\Omega \in \mathbb{R}^{d\times m}$ and $\mathbf{b} \in \mathbb{R}^m$, providing explicit target-kernel control with comparatively small target-side overhead.

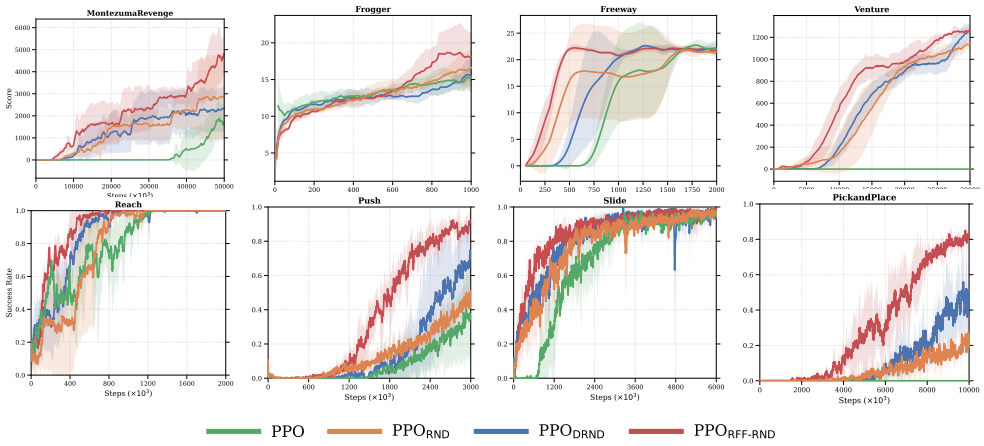


Figure 4: **Mean episode return on online RL tasks.** Top: Atari tasks (*MontezumaRevenge*, *Frogger*, *Freeway*, *Venture*); Bottom: Fetch Robotics tasks (*Reach*, *Push*, *Slide*, *PickAndPlace*). Results are averaged over 5 random seeds.

5.3 Downstream Performance on RL

We compare our method across a range of online and offline RL benchmarks to assess its effect on downstream task performance. We consider discounted MDPs [Sutton and Barto, 2018] $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with objective $J(\pi) = \mathbb{E}_\pi[\sum_{t \geq 0} \gamma^t r(s_t, a_t)]$. The RND energy $\mathcal{E}(x)$ is used as a unified uncertainty signal $b^+(x)$: in online RL, it is added as an exploration bonus, $\tilde{r}_t = r_t + \lambda_{\text{coeff}} b^+(s_t)$; in offline RL, it is used as an anti-exploration penalty in the critic target, $r_t + \gamma\, \mathbb{E}_{a'}[Q_{\theta'}(s_{t+1}, a') - \lambda_{\text{coeff}} b^+(s_{t+1}, a')]$. Here, $x = s_t$ for online exploration and $x = (s_t, a_t)$ for offline conservatism. Detailed RL configurations are provided in Appendix F.

We fix the output dimension to $m = 64$ for online RL and $m = 128$ for offline RL experiments, as performance stabilizes once m is sufficiently large. A detailed analysis on the effect of m is provided in Appendix G.1. As with other RND-based methods, we calibrate the scale of the intrinsic reward by tuning α and λ_{coeff} on a validation grid and selecting dataset-specific values (see Appendix G.2).

Online RL. We evaluate `RFF-RND` on four Atari exploration games (*MontezumaRevenge*, *Frogger*, *Freeway*, *Venture*) and four Fetch manipulation tasks (*Reach*, *Push*, *Slide*, *PickAndPlace*). We use PPO [Schulman et al., 2017] as the base algorithm and compare PPO, PPO_{RND} [Burda et al., 2019b], PPO_{DRND} [Yang et al., 2024], and the proposed $\text{PPO}_{\texttt{RFF-RND}}$, isolating the effect of the intrinsic reward.

As shown in Figure 4, $\text{PPO}_{\texttt{RFF-RND}}$ generally improves sample efficiency and achieves the best or competitive final performance. On sparse-reward Atari tasks, the gains are most visible on *MontezumaRevenge*, where $\text{PPO}_{\texttt{RFF-RND}}$ continues to improve while other methods plateau, and on *Venture* and *Freeway*, where it improves earlier during training. On *Frogger*, $\text{PPO}_{\texttt{RFF-RND}}$ remains consistently competitive with other RND-based variants. The Fetch tasks show a similar trend in goal-conditioned sparse-reward manipulation. $\text{PPO}_{\texttt{RFF-RND}}$ solves *Reach* fastest, is the only method to reach 80% success on *Push* within the training budget, and achieves the best final performance on *PickAndPlace*. On *Slide*, all methods approach comparable final performance, but $\text{PPO}_{\texttt{RFF-RND}}$ learns faster early in training. Overall, these results indicate that the proposed target-side design yields an intrinsic reward that is useful for exploration across various online RL tasks.

Offline RL. We evaluate our method on D4RL locomotion tasks (*HalfCheetah, Hopper, Walker2d*) [Fu et al., 2020], and additionally report results on *AntMaze* tasks to assess performance in goal-directed exploration settings. We compare against strong ensemble-free offline RL baselines, including SAC [Haarnoja et al., 2018], CQL [Kumar et al., 2020], IQL [Kostrikov et al., 2021], ReBRAC [Tarasov et al., 2023], and RND-based variants SAC_{RND} [Burda et al., 2019b] and SAC_{DRND} [Yang et al., 2024]. All RND-based variants are evaluated under the same SAC backbone and D4RL evaluation protocol to ensure fair comparisons.

Table 1: Normalized D4RL Locomotion scores are reported as mean ± std over 10 seeds, with 100 evaluation episodes per seed. Baselines are taken from prior work [An et al., 2021, Tarasov et al., 2023, Yang et al., 2024]. **Bold** and underline denote the best and second-best results.

Dataset	SAC	CQL	IQL	ReBRAC	SAC_{RND}	SAC_{DRND}	$SAC_{RFF\text{-}RND}$
halfcheetah-random	29.7 ± 1.4	$\mathbf{31.1 \pm 3.5}$	19.5 ± 0.8	29.5 ± 1.5	27.6 ± 2.1	30.4 ± 4.0	29.3 ± 1.7
halfcheetah-medium	55.2 ± 27.8	46.9 ± 0.4	50.0 ± 0.2	65.6 ± 1.0	66.4 ± 1.4	$\mathbf{68.3 \pm 0.2}$	67.0 ± 0.5
halfcheetah-expert	-0.8 ± 1.8	97.3 ± 1.1	95.5 ± 2.1	105.9 ± 1.7	102.6 ± 4.2	106.2 ± 3.7	$\mathbf{109.2 \pm 2.0}$
halfcheetah-medium-expert	28.4 ± 19.4	95.0 ± 1.4	92.7 ± 2.8	101.1 ± 5.2	107.6 ± 2.8	108.5 ± 1.1	$\mathbf{109.7 \pm 0.3}$
halfcheetah-medium-replay	0.8 ± 1.0	45.3 ± 0.3	42.1 ± 3.6	51.0 ± 0.8	51.2 ± 3.2	52.1 ± 4.8	$\mathbf{57.0 \pm 2.0}$
halfcheetah-full-replay	$\mathbf{86.8 \pm 1.0}$	76.9 ± 0.9	75.0 ± 0.7	82.1 ± 1.1	81.2 ± 1.3	81.4 ± 1.7	84.3 ± 1.1
hopper-random	9.9 ± 1.5	5.3 ± 0.6	10.1 ± 5.9	8.1 ± 2.4	19.6 ± 12.4	$\mathbf{32.7 \pm 0.4}$	31.4 ± 0.1
hopper-medium	0.8 ± 0.0	61.9 ± 6.4	65.2 ± 4.2	102.0 ± 1.0	91.1 ± 10.1	98.5 ± 1.1	$\mathbf{102.3 \pm 1.5}$
hopper-expert	0.7 ± 0.0	106.5 ± 9.1	108.8 ± 3.1	100.1 ± 8.3	109.7 ± 0.5	109.7 ± 0.3	$\mathbf{110.2 \pm 0.8}$
hopper-medium-expert	0.7 ± 0.0	96.9 ± 15.1	85.5 ± 29.7	107.0 ± 6.4	109.8 ± 0.6	108.7 ± 0.5	$\mathbf{112.0 \pm 0.2}$
hopper-medium-replay	7.4 ± 0.5	86.3 ± 7.3	89.6 ± 13.2	98.1 ± 5.3	97.2 ± 9.0	100.5 ± 1.0	$\mathbf{102.7 \pm 0.4}$
hopper-full-replay	41.1 ± 17.9	101.9 ± 0.6	104.4 ± 10.8	107.1 ± 0.4	107.4 ± 0.8	108.2 ± 0.7	$\mathbf{109.6 \pm 0.4}$
walker2d-random	0.9 ± 0.8	5.4 ± 1.7	11.3 ± 7.0	21.5 ± 0.1	18.7 ± 6.9	21.7 ± 0.1	$\mathbf{22.0 \pm 1.4}$
walker2d-medium	-0.3 ± 0.2	79.5 ± 3.2	80.7 ± 3.4	82.5 ± 3.6	91.6 ± 2.8	$\mathbf{95.2 \pm 0.7}$	94.3 ± 0.9
walker2d-expert	0.7 ± 0.3	109.3 ± 0.1	96.9 ± 32.3	112.3 ± 0.2	104.5 ± 22.8	$\mathbf{114.0 \pm 0.5}$	112.8 ± 0.5
walker2d-medium-expert	1.9 ± 3.9	109.1 ± 0.2	112.1 ± 0.5	111.6 ± 0.3	104.6 ± 11.2	109.6 ± 1.0	$\mathbf{113.2 \pm 0.6}$
walker2d-medium-replay	-0.4 ± 0.3	76.8 ± 10.0	75.4 ± 9.3	77.3 ± 7.9	88.7 ± 7.7	91.0 ± 2.9	$\mathbf{101.7 \pm 3.7}$
walker2d-full-replay	27.9 ± 47.3	94.2 ± 1.9	97.5 ± 1.4	102.2 ± 1.7	105.3 ± 3.2	109.6 ± 0.7	$\mathbf{111.7 \pm 2.8}$
average score	16.2	73.6	72.9	81.4	82.5	85.9	**87.8**

Table 2: Normalized D4RL AntMaze scores are reported as mean ± std over 10 seeds, with 100 evaluation episodes per seed. Baselines are from prior work [Nikulin et al., 2023, Tarasov et al., 2023, Yang et al., 2024]. **Bold** and underline denote the best and second-best results.

Dataset	SAC	CQL	IQL	ReBRAC	SAC_{RND}	SAC_{DRND}	$SAC_{RFF\text{-}RND}$
antmaze-umaze	0.0	74.0	83.3 ± 4.5	$\mathbf{97.8 \pm 1.0}$	97.0 ± 1.5	95.8 ± 2.4	97.0 ± 0.9
antmaze-umaze-diverse	0.0	84.0	70.6 ± 3.7	88.3 ± 13.0	66.0 ± 25.0	87.2 ± 3.2	$\mathbf{88.4 \pm 3.9}$
antmaze-medium-play	0.0	61.2	64.6 ± 4.9	84.0 ± 4.2	38.5 ± 29.4	$\mathbf{86.2 \pm 5.4}$	85.2 ± 6.1
antmaze-medium-diverse	0.0	53.7	61.7 ± 6.1	76.3 ± 13.5	74.7 ± 10.7	$\mathbf{83.0 \pm 3.8}$	80.5 ± 5.5
antmaze-large-play	0.0	15.8	42.5 ± 6.5	$\mathbf{60.4 \pm 26.1}$	43.9 ± 29.2	53.2 ± 4.1	57.1 ± 8.2
antmaze-large-diverse	0.0	14.9	27.6 ± 7.8	$\mathbf{54.4 \pm 25.1}$	45.7 ± 28.5	50.8 ± 10.5	51.6 ± 9.8
average score	0.0	50.6	58.4	**76.9**	61.0	76.0	76.6

As shown in Table 1, $SAC_{RFF\text{-}RND}$ achieves the best average score on D4RL locomotion among ensemble-free baselines, with an average score of 87.8. It obtains the best or second-best performance on 17 out of 18 datasets, with particularly strong gains on mixed-quality datasets such as *medium-replay*, *full-replay*, and *medium-expert*. These results suggest that RFF-RND provides a support-aware uncertainty penalty that helps exploit high-return behavior while discouraging poorly supported actions, letting the actor more reliably exploit high-quality behavior already in the offline dataset.

Table 2 reports additional results on AntMaze. $SAC_{RFF\text{-}RND}$ remains competitive on these goal-reaching tasks, achieving an average score of 76.6, close to the best ensemble-free baseline. This suggests that the proposed uncertainty penalty can extend beyond locomotion domains, although further adaptation to sparse-reward, long-horizon navigation remains an interesting direction.

6 Conclusion and Future Direction

In this paper, we rigorously analyzed random network distillation (RND) overgeneralization under a GP target model and an NTK-regime predictor, showing that the RND score is a cross-kernel residual variance determined by the interaction between target covariance and predictor interpolation. This reveals residual collapse as a spectral under-excitation problem, where the target fails to excite modes in which predictor residuals can survive. Motivated by this mechanism, we proposed RFF-RND, a bandwidth-controlled target design that makes the target spectrum and spatial novelty scale explicit. Diagnostics and RL experiments show improved novelty discrimination and downstream performance, highlighting target geometry as a central design choice for reliable RND.

The theoretical framework proposed in this paper imposes no specific restrictions on the input domain. Specializing it to multiple input modalities, such as images, text, and time series, and analyzing the additional structures that arise in each case is therefore an interesting direction for future research.

7 Practical Considerations for Spectral Target Design

This section summarizes several practical issues discussed during the rebuttal process that further clarify the scope of our spectral target design framework. In particular, we consider data-normalized bandwidth selection, the interaction between spectral design and stochastic nuisance variation, and the potential role of anisotropic target spectra.

Data-normalized bandwidth selection. The RBF bandwidth α should be interpreted relative to the characteristic scale of the input distribution. To obtain a simple data-dependent normalization, we standardize each state and action dimension, concatenate them into an input vector x_i, and compute the median pairwise distance

$$r_{\mathrm{med}} = \mathrm{median}_{i \neq j} \, \|x_i - x_j\|_2.$$

We then parameterize the bandwidth as

$$\alpha = c\, r_{\mathrm{med}}, \qquad c > 0.$$

For the RBF kernel

$$k_\alpha(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\alpha^2}\right),$$

two samples separated by r_{med} have similarity $\exp(-1/(2c^2))$. Thus, c directly controls the decay of target correlation relative to the characteristic geometric scale of the dataset, rather than its raw coordinate scale.

We evaluated this normalization on the 17 offline locomotion datasets used in our experiments. The single choice $c = 1/2$ yielded performance comparable to the best grid-searched bandwidth on 10 datasets, while the candidate sets $c \in \{1/2, 1/3\}$ and $c \in \{1/2, 1/3, 1/5\}$ covered 14 and all 17 datasets, respectively. These results suggest the compact data-normalized candidate set

$$\alpha \in \{r_{\mathrm{med}}/2,\ r_{\mathrm{med}}/3,\ r_{\mathrm{med}}/5\},$$

which substantially reduces the need for extensive task-specific bandwidth search.

Stochastic nuisance variation and the noisy-TV problem. Spectral target design does not by itself eliminate the classical noisy-TV failure mode. Our analysis predicts persistent novelty whenever target variation overlaps with directions that remain difficult for the predictor, regardless of whether such variation is task-relevant or arises from irreducible stochasticity. Hence, allocating more target power to predictor-hard directions should not be interpreted as a monotone preference for arbitrarily high-frequency target spectra.

To make this trade-off explicit, consider an observation $x = (z, \xi)$, where z denotes the task-relevant state and $\xi \sim \mathcal{N}(0, \sigma^2 I_q)$ is an independent nuisance variable. For an isotropic RBF target kernel, two independent observations of the same underlying state, $z = z'$, satisfy

$$\mathbb{E}_{\xi,\xi'}\left[k_\alpha\big((z,\xi),(z,\xi')\big)\right] = \left(1 + \frac{2\sigma^2}{\alpha^2}\right)^{-q/2}.$$

Therefore, when α is small relative to the nuisance scale, repeated observations of the same underlying state can become nearly uncorrelated in the target representation. If these nuisance directions are also difficult for the predictor, their residual contribution persists and can be mistaken for novelty. Conversely, increasing α suppresses stochastic variation but may also smooth over task-relevant differences between states.

This conflict arises because a single isotropic bandwidth controls both types of variation. When task-relevant and nuisance directions can be distinguished, the same framework naturally admits an anisotropic target kernel,

$$k_{\alpha_z,\alpha_\xi} = \exp\left(-\frac{\|z - z'\|_2^2}{2\alpha_z^2} - \frac{\|\xi - \xi'\|_2^2}{2\alpha_\xi^2}\right).$$

Here, α_ξ can be chosen large enough to suppress nuisance variation, while α_z preserves sensitivity to task-relevant novelty. More broadly, our analysis therefore suggests allocating target power to *task-relevant predictor-hard directions*, rather than indiscriminately increasing target frequency. Identifying such directions from side information, repeated observations, controlled augmentations, or invariant representations provides a natural extension of the present isotropic formulation.

Acknowledgement

This work was partly supported by the IITP grant funded by Korean government(MSIT) in 2026 (RS-2022-II220951, LBA, 50%), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25504174, 50%).

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, 2019.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116.

Yuri Burda, Harrison Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019a.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019b.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2019.

Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 2017.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 2018. doi: 10.23915/distill.00011.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Robert M. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, 2(3):155–239, 2006.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1861–1870. PMLR, 2018.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 5774–5783. PMLR, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020.

Quoc V. Le, Tamás Sarlós, and Alex J. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 244–252. PMLR, 2013.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, 1996. ISBN 978-0-387-94724-2.

Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 26228–26244. PMLR, 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8106–8114, 2022.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 11592–11620, 2023.

Kai Yang, Jian Tao, Jiafei Lyu, and Xiu Li. Exploration and anti-exploration with distributional random network distillation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 56397–56421. PMLR, 2024.

Moritz A. Zanger, Yijun Wu, Pascal R. Van der Vaart, Wendelin Böhmer, and Matthijs T. J. Spaan. On the equivalence of random network distillation, deep ensembles, and bayesian inference, 2026. URL `https://arxiv.org/abs/2602.19964`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction state the paper's main contributions: a GP/NTK-based analysis of RND overgeneralization, the resulting spectral interpretation, and the proposed `RFF-RND` target design. The claims are scoped to RND-style novelty estimation and are supported by the theoretical results and experiments reported in the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes].

   Justification: The paper discusses limitations of the present work, including the idealized GP/NTK-regime analysis, the restriction of the theoretical model to target-prediction RND mechanisms, and the fact that the proposed bandwidth-based control is evaluated on a finite set of RL benchmarks. We also discuss practical limitations such as sensitivity to bandwidth selection and the need to further study predictor optimization dynamics and higher-dimensional visual inputs.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes].

   Justification: The paper states the assumptions for each theoretical result, including the GP target model, predictor interpolation kernel, regularization, and bandwidth-controlled kernel conditions. Complete proofs are provided in the appendix, with the main text giving the corresponding intuition and implications.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes].

   Justification: The paper describes the experimental tasks, baselines, model classes, evaluation protocol, random seeds, and hyperparameter ranges used for the main experiments. Additional implementation details are provided in the appendix and in the accompanying anonymized code package.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes].

   Justification: We provide an anonymized code package with scripts and instructions for reproducing the main experimental results. The experiments use standard public RL benchmarks, and the supplemental material describes the required dependencies, data access procedures, and commands for running the proposed method and baselines.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes].

   Justification: The paper specifies the online and offline RL benchmarks, baselines, evaluation metrics, random seeds, training configurations, optimizers, and hyperparameter sweeps used for the proposed method and comparisons. Further details are included in the appendix and accompanying code package.

Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes].

   Justification: The experimental figures and tables report performance over multiple random seeds, with variability shown using standard deviations or shaded regions as specified in the corresponding captions. The reported uncertainty reflects variability across random initialization and environment-level randomness.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes].

   Justification: The appendix reports the compute resources used for the experiments, including GPU type, memory, approximate runtime per run, and estimated total GPU hours. We also indicate that the reported compute corresponds to the experiments included in the paper and does not include all preliminary or failed experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes].

   Justification: The research uses standard reinforcement learning benchmarks and synthetic or publicly available datasets, and does not involve human subjects, private data, or sensitive personal information. We have reviewed the NeurIPS Code of Ethics and believe the work conforms to it.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes].

    Justification: The paper discusses the potential positive impact of more reliable novelty estimation for exploration, offline conservatism, and safety-aware decision making. It also notes that improved exploration and autonomy methods should be carefully evaluated before deployment in safety-critical systems such as robotics or autonomous driving.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A].

    Justification: The paper does not release high-risk pretrained models, image generators, language models, or scraped datasets. The experiments use standard RL benchmarks and do not introduce assets with a high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes].

Justification: The paper cites the original sources for the benchmarks, datasets, and baseline algorithms used in the experiments. The appendix documents the public benchmark environments and software packages used, including their versions and license information where available.

Guidelines:

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

13. **New assets**

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

Answer: [Yes].

Justification: The paper introduces an implementation of `RFF-RND` and provides it as part of the anonymized code package. The accompanying documentation describes the method implementation, dependencies, command-line usage, hyperparameters, and scripts for reproducing the main results.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, human-subject experiments, user studies, or human annotation.

Guidelines:

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

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

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

Answer: [N/A].

Justification: The paper does not involve human subjects, crowdsourcing, or user studies, so IRB approval or equivalent review is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A].

    Justification: The core method, analysis, and experiments do not use LLMs as an important, original, or non-standard component. Any use of general writing or editing assistance, if any, does not affect the scientific methodology, rigor, or originality of the work.

    Guidelines:

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

A Contributions

- **Minseok Jeong:** Project lead. Developed the core algorithms, conducted the mathematical analysis, and led manuscript preparation.
- **Yechan Lee:** Led the downstream offline and online RL experiments.
- **Hyewon Choi:** Supported the downstream offline and online RL experiments.
- **Jeongyong Yang:** Contributed to manuscript writing and refinement of the technical exposition.
- **SooJean Han:** Advised on the project and contributed to manuscript writing.

B Proofs

B.1 Proof of Theorem 3.4

Fix an arbitrary query point $x \in \mathcal{X}$. By Assumption 3.2, the limiting NTK-regime predictor is

$$f_p^\infty(x) = F_X^\top A_x^\top, \qquad A_x = \Theta_p(x, X)(\Theta_p + \lambda I)^{-1}.$$

Let $f_t^{(j)}$ denote the j-th coordinate of the target function, and let

$$F_X^{(j)} = [f_t^{(j)}(x_1), \ldots, f_t^{(j)}(x_n)]^\top$$

be the j-th column of F_X. Under the GP target model, the output coordinates are independent and identically distributed as

$$f_t^{(j)} \sim \mathcal{GP}(0, k_t), \qquad j = 1, \ldots, m.$$

Therefore, for each coordinate j,

$$\begin{bmatrix} f_t^{(j)}(x) \\ F_X^{(j)} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k_t(x,x) & k_t(x,X) \\ k_t(X,x) & K_t \end{bmatrix}\right),$$

where $K_t = k_t(X, X)$.

The j-th coordinate of the residual is

$$r^{(j)}(x) \triangleq f_t^{(j)}(x) - f_p^{\infty,(j)}(x) = f_t^{(j)}(x) - A_x F_X^{(j)}.$$

Thus, conditional on X and the predictor kernel Θ_p, the same deterministic interpolation weights A_x are applied to each target coordinate. Since $r^{(j)}(x)$ is a linear transformation of a jointly Gaussian vector, it is Gaussian. Since the vector is centered, $r^{(j)}(x)$ is also centered, and its second moment is its variance:

$$\mathbb{E}_{f_t}\left[\left(r^{(j)}(x)\right)^2\right] = \mathrm{Var}\left(f_t^{(j)}(x) - A_x F_X^{(j)}\right).$$

Expanding this variance gives

$$\mathrm{Var}\left(f_t^{(j)}(x) - A_x F_X^{(j)}\right) = k_t(x,x) - 2A_x k_t(X,x) + A_x K_t A_x^\top = \sigma^2_{\mathrm{cross}}(x;\, k_t, \Theta_p),$$

where the last equality follows from Definition 3.3. Hence, under the GP target model,

$$r^{(j)}(x) \sim \mathcal{N}\big(0, \sigma^2_{\mathrm{cross}}(x;\, k_t, \Theta_p)\big).$$

For finite output dimension m, the ideal NTK-regime RND energy is

$$\mathcal{E}_m(x) = \frac{1}{m}\|f_t(x) - f_p^\infty(x)\|_2^2 = \frac{1}{m}\sum_{j=1}^m \big(r^{(j)}(x)\big)^2.$$

Because the target coordinates $\{f_t^{(j)}\}_{j=1}^m$ are independent GP samples and the same deterministic weights A_x are applied coordinate-wise, the residual coordinates $\{r^{(j)}(x)\}_{j=1}^m$ are i.i.d. centered Gaussian random variables with variance $\sigma^2_{\mathrm{cross}}(x;\, k_t, \Theta_p)$. Therefore, $\{(r^{(j)}(x))^2\}_{j=1}^m$ are i.i.d. with finite mean $\sigma^2_{\mathrm{cross}}(x;\, k_t, \Theta_p)$. By the strong law of large numbers,

$$\mathcal{E}_m(x) = \frac{1}{m}\sum_{j=1}^m \big(r^{(j)}(x)\big)^2 \xrightarrow[m\to\infty]{\text{a.s.}} \sigma^2_{\mathrm{cross}}(x;\, k_t, \Theta_p).$$

B.2 Proof of Lemma 4.1

By definition,

$$\widehat{F}_X = [f_p^\infty(x_1), \dots, f_p^\infty(x_n)]^\top = \Theta_p(\Theta_p + \lambda I)^{-1} F_X.$$

Hence the training-set residual is

$$R_X = F_X - \widehat{F}_X = \left(I - \Theta_p(\Theta_p + \lambda I)^{-1}\right) F_X.$$

Using

$$I - \Theta_p(\Theta_p + \lambda I)^{-1} = (\Theta_p + \lambda I)(\Theta_p + \lambda I)^{-1} - \Theta_p(\Theta_p + \lambda I)^{-1} = \lambda(\Theta_p + \lambda I)^{-1},$$

we obtain

$$R_X = \lambda(\Theta_p + \lambda I)^{-1} F_X.$$

Now let $\Theta_p = U_p \Lambda_p U_p^\top$. Then

$$\lambda(\Theta_p + \lambda I)^{-1} = U_p \operatorname{diag}\left(\frac{\lambda}{\lambda_{p,1} + \lambda}, \dots, \frac{\lambda}{\lambda_{p,n} + \lambda}\right) U_p^\top.$$

Therefore, for the i-th eigenvector u_i,

$$u_i^\top R_X = u_i^\top \lambda(\Theta_p + \lambda I)^{-1} F_X = \frac{\lambda}{\lambda_{p,i} + \lambda} u_i^\top F_X.$$

Taking squared Euclidean norm gives

$$\|u_i^\top R_X\|_2^2 = \left(\frac{\lambda}{\lambda_{p,i} + \lambda}\right)^2 \|u_i^\top F_X\|_2^2.$$

Under the GP target model, the output coordinates are independent and each column $F_{X,j}$ satisfies

$$F_X^{(j)} \sim \mathcal{N}(0, K_t).$$

Thus, for each coordinate j,

$$\mathbb{E}\left[(u_i^\top F_X^{(j)})^2\right] = u_i^\top K_t u_i.$$

Since $F_X \in \mathbb{R}^{n \times m}$, summing over the m output coordinates yields

$$\mathbb{E}\|u_i^\top F_X\|_2^2 = \sum_{j=1}^m \mathbb{E}\left[(u_i^\top F_X^{(j)})^2\right] = m\, u_i^\top K_t u_i.$$

Combining the two displays,

$$\mathbb{E}\|u_i^\top R_X\|_2^2 = m\left(\frac{\lambda}{\lambda_{p,i} + \lambda}\right)^2 u_i^\top K_t\, u_i.$$

B.3 Proof of Corollary 4.3

Fix $x \in \mathcal{X}$. Under the NTK-regime predictor, the prediction at x is obtained by applying the same interpolation weights to each target coordinate:

$$f_p^\infty(x) = F_X^\top A_x^\top, \qquad A_x = \Theta_p(x, X)(\Theta_p + \lambda I)^{-1}.$$

Let $F_X^{(j)} = [f_t^{(j)}(x_1), \dots, f_t^{(j)}(x_n)]^\top$ denote the j-th target coordinate on the dataset. The corresponding residual coordinate is

$$r^{(j)}(x) = f_t^{(j)}(x) - A_x F_X^{(j)}.$$

By the zero-mean and covariance assumptions,

$$\mathbb{E}[f_t^{(j)}(x)^2] = k_t(x, x), \qquad \mathbb{E}[F_X^{(j)} f_t^{(j)}(x)] = k_t(X, x), \qquad \mathbb{E}[F_X^{(j)}(F_X^{(j)})^\top] = K_t.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[(r^{(j)}(x))^2\right] &= \mathbb{E}\left[\left(f_t^{(j)}(x) - A_x F_X^{(j)}\right)^2\right] \\ &= k_t(x, x) - 2A_x k_t(X, x) + A_x K_t A_x^\top \\ &= \sigma^2_{\text{cross}}(x; k_t, \Theta_p).\end{aligned}$$

Thus, for finite m,

$$\mathbb{E}[\mathcal{E}_m(x)] = \mathbb{E}\left[\frac{1}{m}\sum_{j=1}^m (r^{(j)}(x))^2\right] = \sigma^2_{\text{cross}}(x; k_t, \Theta_p).$$

If the target coordinates are i.i.d., then the random variables $\{(r^{(j)}(x))^2\}_{j=1}^m$ are also i.i.d. with finite mean $\sigma^2_{\text{cross}}(x; k_t, \Theta_p)$. Hence, by the strong law of large numbers,

$$\mathcal{E}_m(x) = \frac{1}{m}\sum_{j=1}^m (r^{(j)}(x))^2 \xrightarrow[m\to\infty]{a.s.} \sigma^2_{\text{cross}}(x; k_t, \Theta_p).$$

B.4 Proof of Proposition 4.5

Let $\{u_\omega\}_\omega$ denote the discrete Fourier basis on the periodic grid. Since X is a uniform grid on a periodic domain and both kernels are shift-invariant, the corresponding Gram matrices are circulant. Hence they are simultaneously diagonalized by the discrete Fourier basis [Gray, 2006].

$$\Theta_p u_\omega = \widehat{\theta}_p(\omega) u_\omega, \qquad K_t^\alpha u_\omega = \widehat{k}_t^\alpha(\omega) u_\omega.$$

Equivalently,

$$\Theta_p = U_\mathcal{F} \widehat{\Theta}_p U_\mathcal{F}^*, \qquad K_t^\alpha = U_\mathcal{F} \widehat{K}_t^\alpha U_\mathcal{F}^*,$$

where $U_\mathcal{F}$ is the discrete Fourier matrix and $\widehat{\Theta}_p$, $\widehat{K}_t^\alpha$ are diagonal matrices whose diagonal entries are $\widehat{\theta}_p(\omega)$ and $\widehat{k}_t^\alpha(\omega)$, respectively.

From Lemma 4.1, the training-set residual satisfies

$$R_X = \lambda(\Theta_p + \lambda I)^{-1} F_X.$$

Projecting this residual onto the Fourier mode u_ω gives

$$u_\omega^* R_X = u_\omega^* \lambda(\Theta_p + \lambda I)^{-1} F_X.$$

Since u_ω is an eigenvector of Θ_p,

$$u_\omega^* \lambda(\Theta_p + \lambda I)^{-1} = \frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda} u_\omega^*.$$

Therefore,

$$u_\omega^* R_X = \frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda} u_\omega^* F_X.$$

Taking squared Euclidean norm over the m output coordinates,

$$\|u_\omega^* R_X\|_2^2 = \left(\frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda} \right)^2 \|u_\omega^* F_X\|_2^2.$$

Under the GP target model, each output coordinate satisfies

$$F_X^{(j)} \sim \mathcal{N}(0, K_t^\alpha), \qquad j = 1, \ldots, m,$$

independently. Hence, for each coordinate,

$$\mathbb{E}\left[\left| u_\omega^* F_X^{(j)} \right|^2 \right] = u_\omega^* K_t^\alpha u_\omega.$$

Since u_ω is also an eigenvector of K_t^α,

$$u_\omega^* K_t^\alpha u_\omega = \widehat{k}_t^\alpha(\omega),$$

where we use the normalization $\|u_\omega\|_2 = 1$. Summing over the m output coordinates gives

$$\mathbb{E}\|u_\omega^* F_X\|_2^2 = m\, \widehat{k}_t^\alpha(\omega).$$

Combining the above displays,

$$\mathbb{E}\|u_\omega^* R_X\|_2^2 = m \left(\frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda} \right)^2 \widehat{k}_t^\alpha(\omega).$$

Therefore, the output-normalized residual contribution is

$$r(\omega) \triangleq \frac{1}{m} \mathbb{E}\|u_\omega^* R_X\|_2^2 = \left(\frac{\lambda}{\widehat{\theta}_p(\omega) + \lambda} \right)^2 \widehat{k}_t^\alpha(\omega).$$

C Ablation on Target Kernel Structure

Our construction is not tied to a particular target kernel. By Bochner's theorem, any shift-invariant kernel with spectral density p_α can be instantiated through random Fourier features, and is therefore compatible with both our analysis and the RFF-RND construction. To test the sensitivity to kernel choice, we compare four representative shift-invariant kernels: RBF, Laplace, Matérn-$\frac{3}{2}$, and Matérn-$\frac{5}{2}$, all with the same bandwidth parameter $\alpha = 0.5$. Writing $r \triangleq \|x - x'\|_2/\alpha$, these kernels are

$$k_{\mathrm{RBF}}(x, x') = \exp\left(-\frac{r^2}{4}\right),$$

$$k_{\mathrm{Lap}}(x, x') = \exp(-r),$$

$$k_{\text{Matérn-3/2}}(x, x') = \left(1 + \sqrt{3}\, r\right) \exp(-\sqrt{3}\, r),$$

$$k_{\text{Matérn-5/2}}(x, x') = \left(1 + \sqrt{5}\, r + \frac{5}{3} r^2\right) \exp(-\sqrt{5}\, r).$$

Empirically, the four kernel choices are difficult to distinguish in any systematic or significant way. They produce broadly similar energy scales and substantially sharper coverage discrimination than standard RND. This suggests that the main benefit comes from enforcing a localized, distance-decaying target covariance, rather than from a delicate choice of the specific shift-invariant kernel family.

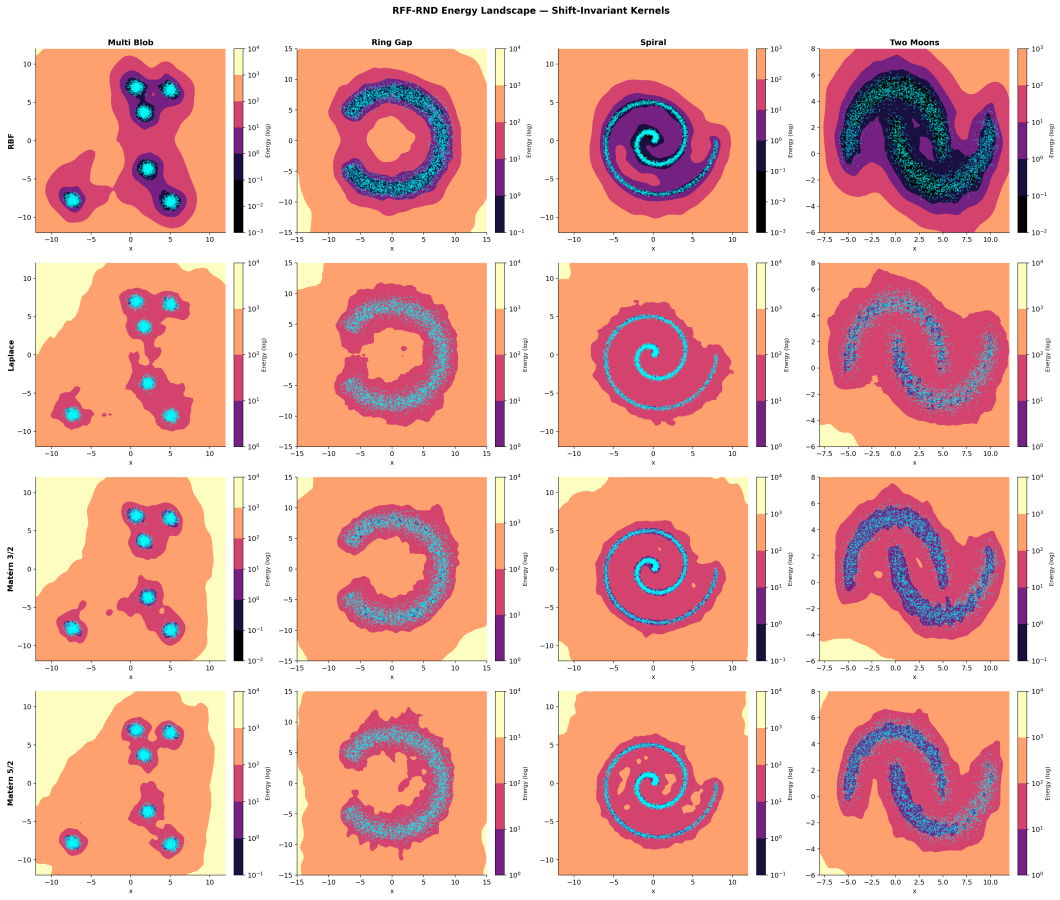


Figure 5: Comparison of RFF-RND using four shift-invariant target kernels—RBF, Laplace, Matérn-$\frac{3}{2}$, and Matérn-$\frac{5}{2}$—all with the same bandwidth $\alpha = 0.5$. The four kernel choices show very similar behavior, with no significant qualitative difference in discrimination performance or overall energy scale. In contrast, all kernelized variants provide substantially clearer ID/OOD separation than standard RND.

D Diagnostics and Ablations for RFF-RND

D.1 Diagnostics on Target Output Dimension m and Predictor Layer Width h

Corollary 4.3 states that, under the RFF target and NTK-regime predictor, the ideal infinite-output RND energy satisfies

$$\mathcal{E}_\infty(x) = \sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p), \quad \text{a.s.},$$

for each fixed $x \in \mathcal{X}$, where $\mathcal{E}_\infty(x) = \lim_{m\to\infty} \mathcal{E}_m(x)$. In practice, however, RND uses a single realization of the target with finite output dimension m. For a fixed query point x, the residual coordinates are i.i.d. under the RFF target construction, and hence

$$\mathcal{E}_m(x) = \frac{1}{m}\|f_t(x) - f_p^\infty(x)\|_2^2 \xrightarrow[m\to\infty]{\text{a.s.}} \mathcal{E}_\infty(x) = \sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p), \quad \text{pointwise in } x.$$

Thus, the finite-output empirical energy $\mathcal{E}_m(x)$ should approach the theoretical prediction as m increases. We verify this concentration empirically by measuring the agreement between $\mathcal{E}_m(x)$ and $\sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p)$ as a function of m.

Protocol. We use the Two Moons dataset ($n = 150$, $d = 2$). The predictor is a 2-hidden-layer MLP with hidden width $h \in \{16, 32, 64\}$ and output dimension $m \in \{4, 8, 16, 32, 64, 128, 256\}$. The empirical NTK Θ_p is computed via full Jacobians at initialization. The theoretical value $\sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p)$ is computed in closed form using Definition 3.3. The RFF target uses bandwidth $\alpha = 0.5$ and $\lambda = 0.1$. Agreement is measured by Pearson r between $\sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p)$ and $\mathcal{E}_m(x)$, evaluated on a 22×22 grid covering the input space and averaged over 5 random seeds.

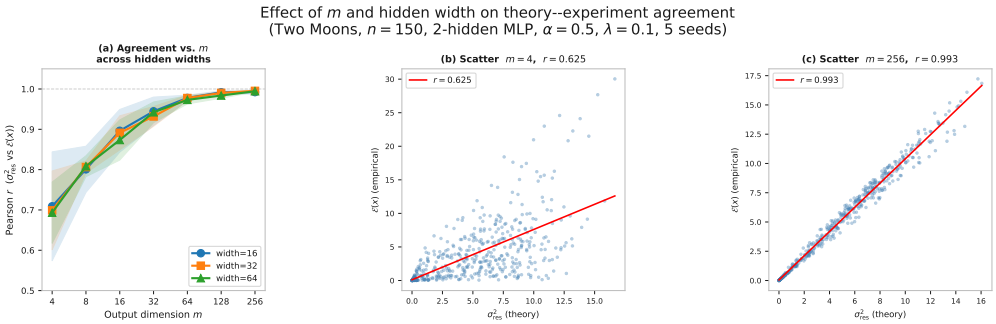


Figure 6: As the RFF target output dimension m increases, the finite-output empirical energy $\mathcal{E}_m(x)$ concentrates around its infinite-output limit $\mathcal{E}_\infty(x) = \sigma^2_{\text{cross}}(x;\, k_t^\alpha, \Theta_p)$. Since the output coordinates are i.i.d., this concentration follows from the law of large numbers and yields higher empirical agreement with the theoretical prediction.

Interpretation. Figure 6-(a) shows Pearson r as a function of the RFF target output dimension m across three hidden widths. The three curves are nearly indistinguishable across $h \in \{16, 32, 64\}$. This suggests that the observed agreement is robust to predictor width. On this low-dimensional two-moons dataset, even the narrow predictors induce sufficiently similar empirical NTK geometry for the residual-variance prediction to remain stable across widths. We note that in higher-dimensional settings, wider networks may be needed to obtain a reliable empirical NTK estimate; we leave this verification to future work.

Figure 6-(a) also shows the agreement improves consistently as m increases: starting from a relatively noisy finite-output estimate at $m = 4$, the correlation rapidly approaches one and reaches nearly perfect agreement by $m = 256$. This trend is consistent with the concentration argument above, since $\mathcal{E}_m(x)$ is an empirical average over m independent target coordinates. The scatter plots make this effect explicit. At $h = 64, m = 4$, substantial finite-output fluctuations remain and the empirical energy only coarsely tracks $\sigma^2_{\text{cross}}(x;\, k_t, \Theta_p)$ (Figure 6(b)). In contrast, at $h = 64, m = 256$, the scatter is almost linear with Pearson $r = 0.993$, showing that the finite-output realization closely matches the infinite-output prediction (Figure 6-(c)).

The key takeaway is that Corollary 4.3 characterizes the infinite-output limit of NTK predictor RND energy, while practical RFF-RND uses a single finite-dimensional target realization. For small m, $\mathcal{E}_m(x)$ has noticeable finite-output variance and therefore only approximately follows its limit. Increasing m reduces this variance and makes a single RFF-RND realization track $\sigma^2_{\text{cross}}(x;\, k_t, \Theta_p)$ more reliably. In practice, the agreement already becomes strong at moderate dimensions, with $m = 64$ yielding high correlations across all hidden widths, and becomes nearly exact by $m = 256$.

D.2 Frequency-domain Diagnostic of Proposition 4.5

Proposition 4.5 states that, on a uniform periodic grid with shift-invariant predictor and target kernels, the residual power at frequency ω decomposes as

$$r_\alpha(\omega) = \frac{1}{m}\mathbb{E}\|u_\omega^* R_X\|_2^2 = \left(\frac{\lambda}{\widehat{\Theta}_p(\omega)+\lambda}\right)^2 \widehat{k}_t^\alpha(\omega).$$

This experiment verifies the product-form decomposition in the idealized setting where the proposition applies exactly. The goal is not to evaluate downstream performance, but to visualize how the RFF target spectrum is filtered by the predictor residual filter.

Protocol. We use a one-dimensional uniform periodic grid with $N = 512$ points on $[0, 1)$. The predictor kernel is a periodic RBF kernel with bandwidth $\alpha_p = 0.08$, and the ridge parameter is $\lambda = 0.05$. We vary the target RBF bandwidth over

$$\alpha_t \in \{0.02, 0.04, 0.08, 0.16, 0.32\}.$$

For each α_t, we compute the target spectrum $\widehat{k}_t^{\alpha_t}(\omega)$, the predictor residual filter

$$\left(\frac{\lambda}{\widehat{\Theta}_p(\omega)+\lambda}\right)^2,$$

and the theoretical residual spectrum

$$r_{\alpha_t}^{\text{theory}}(\omega) = \left(\frac{\lambda}{\widehat{\Theta}_p(\omega)+\lambda}\right)^2 \widehat{k}_t^{\alpha_t}(\omega).$$

To obtain the empirical residual spectrum, we sample target functions from the periodic GP with covariance $k_t^{\alpha_t}$, apply the NTK residual operator, and measure the residual power in each Fourier mode. The empirical curves are averaged over Monte Carlo target samples.

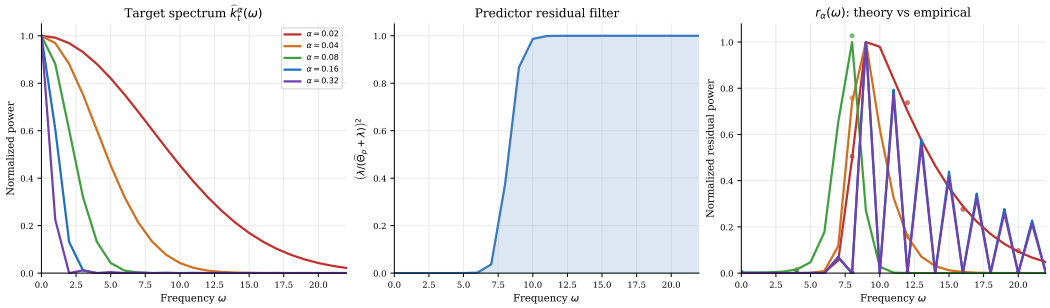


Figure 7: **Frequency-domain residual decomposition on a periodic grid.** Left: changing the target bandwidth α_t reshapes the target spectrum $\widehat{k}_t^{\alpha_t}(\omega)$. Middle: the predictor kernel induces a residual filter $(\lambda/(\widehat{\Theta}_p(\omega)+\lambda))^2$, which suppresses frequencies with large predictor spectral mass. Right: the residual spectrum is the product of these two terms. Empirical residuals obtained by sampling target functions and applying the NTK residual operator follow the predicted frequency-wise structure.

Interpretation. Figure 7 illustrates that residual signal does not appear wherever the target has large spectral power. Low frequencies have large target power for smooth targets, but they are strongly suppressed by the predictor residual filter. Conversely, very high frequencies pass the filter but may receive little target power. Residual power is therefore largest where the target spectrum overlaps with frequencies that are not interpolated away by the predictor. This supports the interpretation of the RFF bandwidth as a spectral knob: changing the target bandwidth moves target power across frequencies and thereby controls where RND residual signal can survive.

D.3 Ablation on Energy Landscapes Across α, m, and h

We further examine how the RFF-RND energy landscape changes with the target bandwidth α, target output dimension m, and predictor width h. These ablations are intended to visualize the geometric effect of each design parameter, rather than to tune downstream RL performance.

Effect of bandwidth α and output dimension m. Figure 8 shows energy landscapes on the Two Moons dataset. Varying α changes the spatial scale of the target kernel: smaller α produces more localized energy around the sampled manifold, while larger α over-smooths the landscape and blurs the separation between the manifold and nearby off-manifold regions. Varying m primarily affects finite-output stability. Small m yields noisier energy landscapes, whereas larger m gives smoother and more consistent separation, consistent with the concentration behavior in Corollary 4.3.

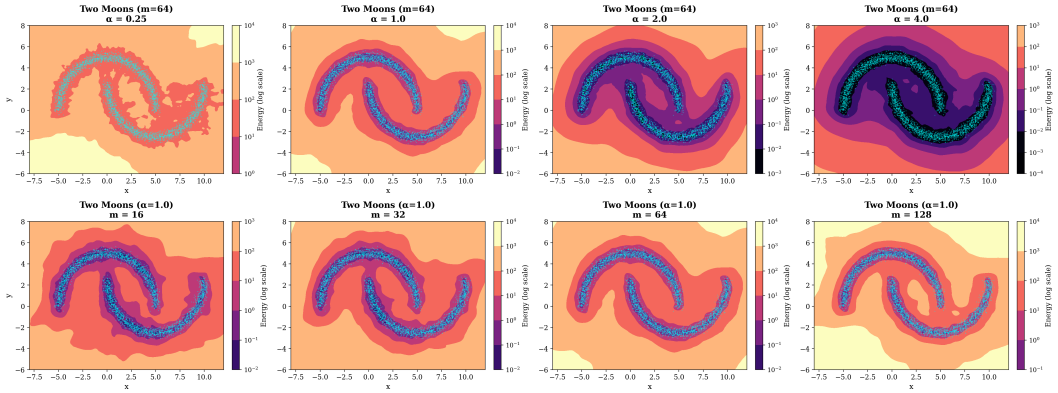


Figure 8: Energy landscapes of RFF-RND on the Two Moons dataset over varying α and m. **(Top Row)** Varying the RBF length-scale α with $m = 64$ fixed. **(Bottom Row)** Varying the RFF feature dimension m with $\alpha = 1.0$ fixed. Overall, smaller α yields more localized (but noisier) energy bands around the manifold, whereas larger α over-smooths the landscape and blurs manifold boundaries. Increasing m improves the kernel approximation and stabilizes the energy, producing smoother and more consistent manifold separation.

Effect of predictor width h. Figure 9 varies the predictor hidden width while keeping the RFF target fixed. Across widths, RFF-RND preserves a localized energy structure around the data manifold. Increasing h slightly smooths the energy contours near the sampled manifold, but the overall manifold-aware separation remains stable. This suggests that the qualitative geometry is primarily controlled by the target kernel, while predictor width mainly affects the smoothness of the fitted interpolation.

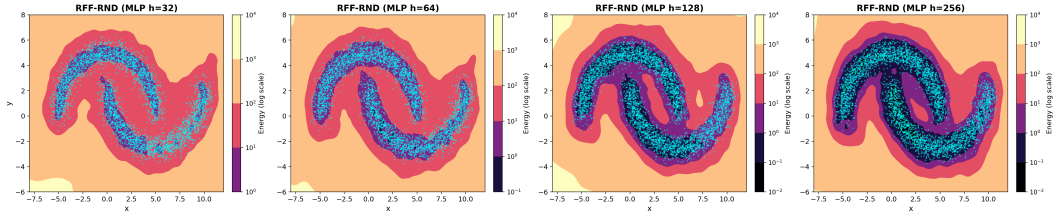


Figure 9: **Effect of predictor width on RFF-RND energy landscapes.** We vary the MLP hidden width $h \in \{32, 64, 128, 256\}$ while keeping the RFF target fixed. Across widths, RFF-RND produces a localized energy landscape that separates the data manifold from off-manifold regions. Wider predictors sharpen the low-energy region around the sampled manifold, but the overall manifold-aware structure remains consistent.

E Additional Ablations on RND variants

E.1 RND Variants

Distributional RND Yang et al. [2024]. We first describe the structure of DRND in Algorithm 1.

Algorithm 1 Distributional Random Network Distillation (DRND)

Input: Sample x; frozen random target networks $\{\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_N\}$; trainable predictor f_θ; step size η; aggregation weight $\beta \in [0, 1]$.
Output: Novelty score $b(x)$ and updated predictor parameters θ.

// Training Phase
Sample target index uniformly: $u(x) \sim \text{Unif}\{\tilde{f}_1(x), \tilde{f}_2(x), \ldots, \tilde{f}_N(x)\}$
Compute prediction: $f_\theta(x)$
Compute MSE loss: $L(\theta) = \|f_\theta(x) - u(x)\|_2^2$
Update predictor: $\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$

// Novelty Computation
Compute ensemble statistics:
$\mu(x) = \mathbb{E}[X] = \frac{1}{N} \sum_{i=1}^{N} \tilde{f}_i(x)$
$B_2(\mathbf{x}) = \mathbb{E}[X^2] = \frac{1}{N} \sum_{i=1}^{N} \|\tilde{f}_i(x)\|^2$

Compute normalized deviation:
$b_2(x) = \sqrt{\frac{\|f_\theta(x)\|_2^2 - \|\mu(x)\|_2^2}{B_2(x) - \|\mu(x)\|_2^2}}$

Compute final novelty score:
$b(x) = \beta \|f_\theta(x) - \mu(x)\|^2 + (1 - \beta) b_2(x)$
Return: $b(x)$

The hyperparameters used for DRND are reported in Table 3. Figure 10 visualizes the decomposition of the DRND novelty signal into its base and distributional components. Due to a scale mismatch, we do not apply logarithmic scaling to DRND in this visualization. The base term $\beta \|f_\theta(x) - \mu(x)\|^2$ closely resembles vanilla RND, yielding a smooth energy landscape that separates data-supported regions from off-manifold areas but fails to resolve fine-grained geometric structure. The distributional correction term $(1 - \beta) b_2(x)$ partially compensates for this behavior by amplifying certain regions of high prediction uncertainty. This effect is most apparent in the Multi Blob setting, where global density differences align with the correction term and lead to improved separation between clusters.

However, for manifolds with pronounced local geometry, such as Ring Gap, Spiral, and Two Moons, the correction term does not reliably capture interior holes, winding structures, or near-boundary regions. As a result, the combined novelty signal remains dominated by global reweighting rather than locality-aware discrimination. These observations indicate that while DRND improves calibration over vanilla RND, it does not explicitly alter

Table 3: DRND Hyperparameters

Parameter	**Value**
Input dimension	2
Feature dimension (m)	32
Number of target networks (N)	10
Hidden dimension	64
Network depth	2
Mixing parameter (β)	0.9
Learning rate	1e-4
Batch size	128
Training epochs	5,000
Optimizer	Adam

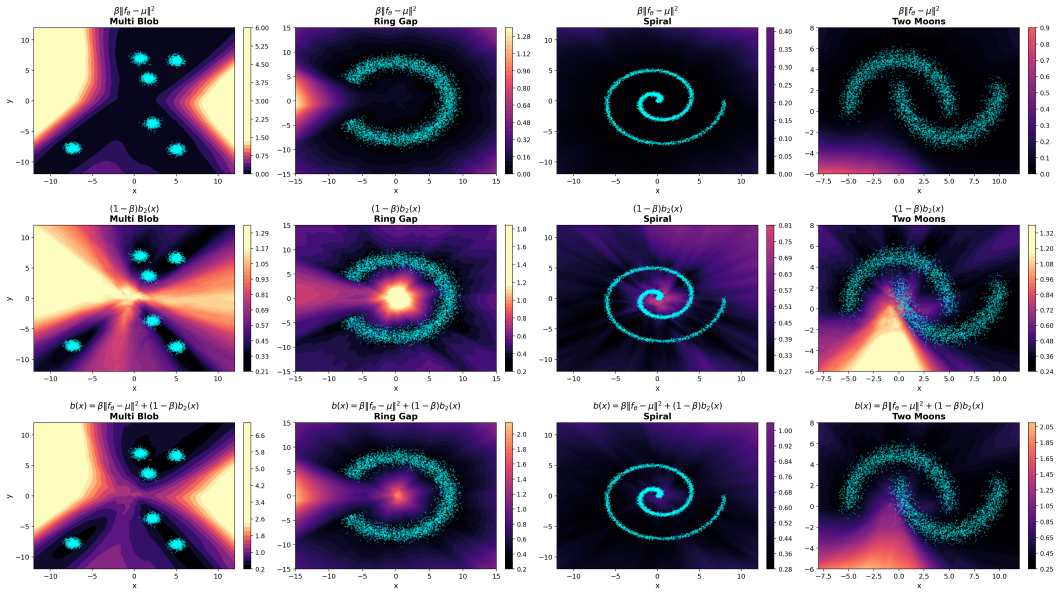


Figure 10: Energy decomposition of DRND on synthetic manifolds. The first row shows the base RND-like component $\beta\|f_\theta(x) - \mu(x)\|^2$, which provides a coarse separation between data and off-manifold regions but remains overly smooth. The second row shows the distributional correction term $(1-\beta)b_2(x)$, which partially sharpens the energy landscape by reweighting prediction errors. The third row shows the combined novelty signal $b(x) = \beta\|f_\theta(x) - \mu(x)\|^2 + (1-\beta)b_2(x)$. While the correction term improves contrast in simple settings such as Multi-Blob, DRND remains insufficiently sensitive to local manifold geometry in more structured datasets, including Ring, Spiral, and Two-Moons.

the underlying kernel geometry, motivating approaches that directly control locality through the construction of the induced kernel.

Input-conditioned RND Nikulin et al. [2023]. There are three main types of neural architectures used in this variant of RND.

Gating. Elementwise multiplicative modulation of two learned feature maps:

$$f(x) = \tanh(\mathbf{W}_1 x + \mathbf{b}_1) \odot (\mathbf{W}_2 x + \mathbf{b}_2).$$

This form induces axis-aligned, locally smooth variations in the energy landscape, but remains limited in expressing curved or anisotropic structures.

Bilinear. Second-order interaction between input coordinates:

$$f(x) = x^\top \mathbf{W} x \; + \; \mathbf{V} x \; + \; \mathbf{b},$$

where $\mathbf{W}$ is a third-order tensor. This allows explicit coupling between $x = (x, y)$ and can induce sharp, direction-dependent distortions of the novelty signal.

FiLM. Input-dependent affine modulation of hidden activations $\mathbf{h}(x)$:

$$f(\mathbf{h}(x)) = \gamma(x) \odot \mathbf{h}(x) \; + \; \beta(x).$$

Essentially, FiLM acts as a low-rank bilinear modulation, yielding smoother, globally consistent deformations of the energy landscape.

In Figure 11, we visualize the RND energy over a two-dimensional input space. Since the conditioning signal is derived from the input itself, each method effectively induces a different input-dependent transformation of the space. Gating-conditioned RND produces relatively smooth energy variations, but the modulation remains largely axis-aligned and struggles to adapt to strongly curved manifolds such as spirals. Bilinear conditioning introduces explicit second-order interactions between input coordinates, leading to sharp and highly directional distortions. While expressive, this often results in unstable or overly aggressive deformation of the energy landscape. In contrast, FiLM-conditioned RND yields smoother and more globally coherent energy landscapes. The induced modulation better follows the intrinsic geometry of curved manifolds, preserving low energy regions along the data support while maintaining separation across low-density gaps.

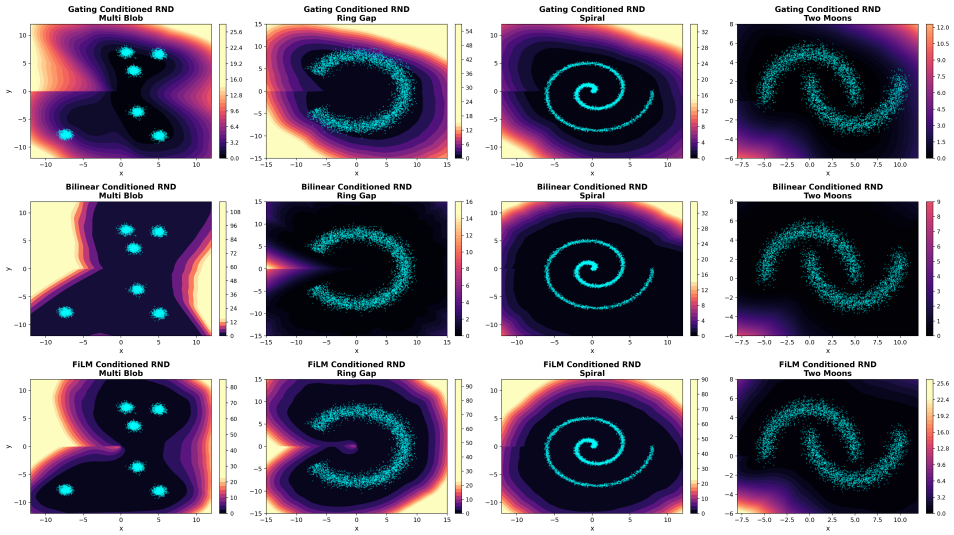


Figure 11: Comparison of input-dependent conditioning mechanisms in RND on two-dimensional synthetic manifolds. Rows correspond to Gating-, Bilinear-, and FiLM-conditioned RND, respectively, while columns show Multi-Blob, Ring Gap, Spiral, and Two-Moons datasets. Background contours visualize the RND energy over the input space, and cyan points denote training samples. Gating induces locally smooth but axis-aligned modulation, Bilinear conditioning yields sharp, direction-dependent distortions, and FiLM produces smoother, globally consistent energy landscapes that better respect curved manifold geometry.

E.2 Target-Side Control versus Predictor-Side Modulation

DRND maintains N frozen target MLPs $\{f_i\}_{i=1}^{N}$ and trains a single predictor f_θ to track a uniformly sampled target at each step; novelty is computed as a deviation statistic $b(x)$ relative to the target ensemble mean $\mu(x)$. The target-induced kernel is therefore determined by the MLP function class at initialization—an NTK-regime quantity that is not analytically characterized—and evaluating $b(x)$ requires a full forward pass through all N networks.

Input-conditioned RND variants (gating, bilinear, FiLM) modulate the *predictor*'s function space via a conditioning signal while keeping the target network unconditional; the resulting novelty geometry is shaped jointly by the modulation architecture and the choice of conditioning representation (e.g., hand-crafted polar coordinates), neither of which admits a closed-form kernel characterization.

In contrast, `RFF-RND` replaces the random target network with two fixed buffers $\Omega \in \mathbb{R}^{d\times m}$ and $\mathbf{b} \in \mathbb{R}^m$, producing features $\varphi(x) = \sqrt{2}\cos(\Omega^\top x + \mathbf{b})$ that are a Bochner-consistent approximation of a Gaussian kernel with bandwidth α [Rahimi and Recht, 2007]. The target-side geometry is therefore explicit and analytically specified: adjusting α directly controls the length scale of the induced kernel, without additional ensemble members or modulation modules.

E.3 RND Variants: Architecture and Cost

Table 4 summarizes the target and predictor architecture of each method. RND and all input-conditioned variants (Gating, Bilinear, FiLM) maintain a single frozen target MLP, whereas DRND stores N frozen target networks $\{\tilde{f}_i\}_{i=1}^{N}$. `RFF-RND` replaces the target network entirely with two fixed buffers $\Omega \in \mathbb{R}^{d\times m}$ and $\mathbf{b} \in \mathbb{R}^m$, with no gradient or optimizer state.

Table 5 reports parameter counts and per-sample inference complexity under the default configuration (d=2, h=64, m=64, N=10, L=2). DRND incurs the largest memory footprint (70,752 parameters), with target-side inference scaling as $\mathcal{O}(NLh^2)$ due to the full forward pass required through all N networks. Among the input-conditioned variants, Bilinear is the most expensive predictor-side at $\mathcal{O}(d^2m)$ from the third-order tensor contraction, while Gating and FiLM remain at $\mathcal{O}(hm)$. `RFF-RND` matches standard RND in total parameter count (8,704 vs. 17,024) while reducing target-side cost to $\mathcal{O}(dm)$—a single matrix multiply followed by an elementwise cosine.

Table 4: Architecture summary of compared methods. d: input dim, h: hidden dim, m: feature dim, N: number of targets (DRND only), L: number of hidden layers.

Method	**Target (frozen)**	**Predictor**
RND	MLP($d \to h^L \to m$)	MLP($d \to h^L \to m$)
DRND	N frozen MLPs $\{\tilde{f}_i\}_{i=1}^{N}$	MLP($d \to h^L \to m$)
Gating	MLP($d \to h^L \to m$)	$\tanh(\mathbf{W}_1 x + \mathbf{b}_1) \odot (\mathbf{W}_2 x + \mathbf{b}_2)$
Bilinear	MLP($d \to h^L \to m$)	$x^\top \mathbf{W} x + \mathbf{V} x + \mathbf{b},\ \mathbf{W} \in \mathbb{R}^{d\times d\times m}$
FiLM	MLP($d \to h^L \to m$)	$\gamma(x) \odot h(x) + \beta(x)$
`RFF-RND`	$\varphi(x) = \sqrt{2}\cos(\Omega^\top x + \mathbf{b}),\ \Omega \in \mathbb{R}^{d\times m}, \mathbf{b} \in \mathbb{R}^m$	MLP($d \to h^L \to m$)

Table 5: Parameter counts and per-sample inference complexity. Defaults: d=2, h=64, m=64, N=10, L=2.

Method	**Trainable**	**Frozen**	**Total**	**Target** $\mathcal{O}(\cdot)$	**Predictor** $\mathcal{O}(\cdot)$
RND	8,512	8,512	17,024	$\mathcal{O}(Lh^2)$	$\mathcal{O}(Lh^2)$
DRND	6,432	64,320	70,752	$\mathcal{O}(NLh^2)$	$\mathcal{O}(Lh^2)$
Gating	2,368	6,432	8,800	$\mathcal{O}(Lh^2)$	$\mathcal{O}(hm)$
Bilinear	6,432	6,432	12,864	$\mathcal{O}(Lh^2)$	$\mathcal{O}(d^2m)$
FiLM	6,816	6,432	13,248	$\mathcal{O}(Lh^2)$	$\mathcal{O}(hm)$
`RFF-RND`	8,512	192†	8,704	$\mathcal{O}(dm)$	$\mathcal{O}(Lh^2)$

† Fixed buffers (Ω, b); no gradient, no optimizer state.
DRND target cost scales as $\mathcal{O}(NLh^2)$; all others maintain a single frozen target. Bilinear predictor incurs $\mathcal{O}(d^2m)$ from the third-order tensor contraction $x^\top \mathbf{W} x$. `RFF-RND` target reduces to a single matmul and elementwise cosine, $\mathcal{O}(dm)$.

F Detailed Hyperparameters for RL Experiments

For all RL experiments, we use the log-scaled RND signal

$$b^+(\cdot) = \log\bigl(1 + \mathcal{E}(\cdot)\bigr)$$

as the intrinsic reward in exploration settings and as the intrinsic penalty signal in anti-exploration settings. The constant 1 ensures numerical stability and keeps the logarithm well-defined for nonnegative energies. We adopt this scaling because raw RND energies can vary substantially across domains and input modalities, making the corresponding reward or penalty coefficient difficult to tune. Log-scaling is a common practical remedy in RND-style methods and is applied uniformly across all RND variants in our experiments.

F.1 Online RL Setting

Proximal Policy Optimization (PPO). PPO is an on-policy actor–critic method that alternates between collecting rollouts with a stochastic policy (actor) and improving both the policy and a state-value function (critic) using minibatch stochastic gradient descent. Let the actor be a parameterized policy $\pi_{\theta_{\text{actor}}}(a \mid s)$ and the critic be a value function $V_{\phi_{\text{critic}}}(s)$. Given trajectories $\{(s_t, a_t, r_t)\}_{t=0}^{T-1}$ sampled from the current policy $\pi_{\theta^-_{\text{actor}}}$, PPO forms advantage estimates $\hat{A}_t$ (e.g., via GAE) using temporal-difference residuals $\delta_t = r_t + \gamma V_{\phi_{\text{critic}}}(s_{t+1}) - V_{\phi_{\text{critic}}}(s_t)$. The actor is updated by maximizing the clipped surrogate objective

$$\mathcal{L}_{\text{PPO}}(\theta_{\text{actor}}) \;=\; \mathbb{E}_t\Bigl[\min\Bigl(\rho_t(\theta_{\text{actor}})\hat{A}_t,\; \text{clip}(\rho_t(\theta_{\text{actor}}), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\hat{A}_t\Bigr)\Bigr] + \beta_{\text{ent}}\,\mathbb{E}_t[\mathcal{H}(\pi_{\theta_{\text{actor}}}(\cdot \mid s_t))]\,, \tag{2}$$

where $\rho_t(\theta_{\text{actor}}) = \pi_{\theta_{\text{actor}}}(a_t \mid s_t)/\pi_{\theta^-_{\text{actor}}}(a_t \mid s_t)$, ϵ_{clip} is the clip parameter, and the entropy bonus (weight β_{ent}) promotes exploration. The critic is trained by regression to a bootstrap target

$$\min_{\phi_{\text{critic}}} \; \mathbb{E}_t\left[\bigl(V_{\phi_{\text{critic}}}(s_t) - \hat{V}_t\bigr)^2\right], \qquad \hat{V}_t = \hat{A}_t + V_{\phi^-_{\text{critic}}}(s_t), \tag{3}$$

In practice, we collect a batch of on-policy rollouts, compute $\hat{A}_t$ and $\hat{V}_t$, and perform multiple epochs of minibatch updates of (2)–(3).

Online RL hyperparameters. For online RL, Table 6 reports the common PPO and training hyperparameters used across the Atari and Fetch Robotics experiments. These include learning rates, discount factor, GAE coefficient, rollout length, clip coefficient, number of environments, network architecture, and the intrinsic reward coefficient. Task-specific `RFF-RND` hyperparameters for Atari and Fetch Robotics, including the output dimension m and bandwidth α, are discussed in Appendix G.

Table 6: Hyperparameters of online experiments (Section 5.3)

Parameters	**Value**
Learning rate of the actor network	1e-4
Learning rate of the critic network	1e-4
Learning rate of the `RFF-RND` network	3e-4
Optimizer	`Adam`
Discount factor (γ)	0.99
GAE coefficient (λ_{GAE})	0.95
Rollout length (T)	256
clip coefficient (ϵ_{clip})	0.1
Number of environments	32
Actor–Critic network ($\pi_{\theta_{\text{actor}}}$, $V_{\phi_{\text{critic}}}$)	3-layer CNN encoder + shared MLP
Activation function	`ReLU`
Number of training epochs	4
Intrinsic reward coefficient (λ_{coeff})	1

F.2 Offline RL Setting

Soft Actor–Critic (SAC). SAC is an off-policy actor–critic method that learns a stochastic policy by maximizing both expected return and entropy, using replayed transitions. The actor is a stochastic policy $\pi_{\theta_{\text{actor}}}(a \mid s)$, and the critic consists of one or more state–action value functions $Q_\phi(s, a)$ (often two critics to

mitigate overestimation), together with slowly-updated target networks $\bar{\phi}$. Given transitions (s, a, r, s') sampled from a replay buffer, the critic is trained to satisfy the soft Bellman equation

$$\min_{\phi} \mathbb{E}\Big[\big(Q_\phi(s,a) - y(s,a,r,s')\big)^2\Big], \quad y = r + \gamma\, \mathbb{E}_{a' \sim \pi_{\theta_{\text{actor}}}(\cdot|s')}\Big[\min_i Q_{\bar{\phi}_i}(s', a') - \alpha_{\text{ent}} \log \pi_{\theta_{\text{actor}}}(a' \mid s')\Big], \tag{4}$$

where $\alpha_{\text{ent}} > 0$ is the temperature controlling the entropy term. The actor is updated to maximize the soft objective, equivalently minimizing

$$\min_{\theta_{\text{actor}}} \mathbb{E}_{s \sim \mathcal{B},\, a \sim \pi_{\theta_{\text{actor}}}(\cdot|s)}\Big[\alpha_{\text{ent}} \log \pi_{\theta_{\text{actor}}}(a \mid s) - \min_i Q_{\phi_i}(s,a)\Big]. \tag{5}$$

Optionally, α_{ent} is tuned by minimizing a dual objective to match a target entropy. Training alternates between sampling minibatches from the replay buffer, updating critics via (4), updating the actor via (5), and updating target networks with Polyak averaging.

Offline RL hyperparameters. For offline RL, Table 7 reports the common optimizer, architecture, and training hyperparameters used across the D4RL experiments. These include learning rates, discount factors, batch size, network dimensions, and the number of training updates. Dataset-specific `RFF-RND` coefficients are discussed separately in Appendix G. In particular, that section reports the selected values of λ_{coeff} and α for D4RL locomotion tasks, as well as λ_a, λ_c, and α for AntMaze tasks.

Table 7: Hyperparameters of D4RL offline experiments (Section 5.3).

Parameters	**Value**
Learning rate of the actor network	1e-3 (1e-4 on AntMaze)
Learning rate of the critic network	1e-3 (1e-4 on AntMaze)
Learning rate of the `RFF-RND` network	1e-6 (1e-5 on AntMaze)
Optimizer	`Adam`
Activation function	`ReLU`
Target entropy of the actor	`-action_dim`
Soft update rate	0.005
Discount factor (γ)	0.99 (0.999 on AntMaze)
Batch size	1,024
Actor, Critic Hidden dimension	256
Actor, Critic Network depth	3
`RFF-RND` hidden dimension	256
`RFF-RND` predictor network depth	4
`RFF-RND` output dimensions (m)	128
`RFF-RND` training epochs	100
Maximum iteration number of SAC	3,000
Gradient updates per iteration	1,000

G Ablation on RFF-RND Hyperparameter Tuning for RL

G.1 Concentration Effect of m

We first study the effect of the RFF output dimension m on downstream offline RL performance. As discussed in Section 5.1, m controls the finite-output stability of the RND energy. In particular, increasing m reduces finite-output variance and makes the empirical energy $\mathcal{E}_m(x)$ more closely match the kernel-level prediction. The energy-landscape diagnostics also show that larger m produces smoother and more stable empirical novelty estimates, without changing the underlying target kernel itself.

To verify whether this diagnostic observation translates to downstream performance, we compare $m \in \{32, 64, 128\}$ on offline D4RL medium datasets. We focus on medium datasets because they provide a balanced setting where both data coverage and policy suboptimality exist, making the effect of representation stability more observable. As shown in Table 8, the performance variation across m is relatively small, but $m = 128$ achieves the best average performance across the three offline medium datasets. This suggests that increasing m improves the stability of the random feature approximation, although the downstream gain is marginal once m is sufficiently large.

Table 8: Performance comparison across different output dimensions m on offline D4RL medium datasets.

Dataset	$m = 32$	$m = 64$	$m = 128$
halfcheetah-medium	66.3 ± 1.1	66.9 ± 2.3	67.0 ± 0.5
hopper-medium	101.5 ± 2.3	99.5 ± 6.7	102.3 ± 3.5
walker2d-medium	92.3 ± 4.1	91.5 ± 4.6	93.7 ± 2.8
Average	86.7	86.0	87.7

We also evaluate the effect of m in online RL using Atari Freeway and Frogger. For this validation, each run is trained for 1M environment steps. As shown in Table 9, the performance variation across m is relatively small once m is moderately large, and $m = 64$ achieves the best average performance.

Table 9: Performance comparison across different output dimensions m on online Atari tasks trained for 1M update steps.

Environment	$m = 32$	$m = 64$	$m = 128$
Freeway	20.3 ± 2.5	22.8 ± 1.9	21.5 ± 2.3
Frogger	16.4 ± 3.7	17.1 ± 2.5	16.8 ± 3.1
Average	18.4	20.0	19.2

Overall, changing m has a marginal effect on downstream performance compared with the effects of α and λ_{coeff}. Based on these validation results, we use $m = 128$ for the offline RL experiments and $m = 64$ for the online RL experiments.

G.2 Performance Across λ_{coeff} and α

Hyperparameter search protocol. We describe the hyperparameter search protocol used for the RL experiments. Following common practice in prior RND-based offline RL studies [Nikulin et al., 2023, Tarasov et al., 2023, Yang et al., 2024], we use a grid search to select task-specific hyperparameters.

For the online RL Atari tasks, we fix $\lambda_{\text{coeff}} = 1$ across all Atari and Fetch experiments to isolate the effect of the intrinsic reward geometry. We tune $\alpha \in \{0.5, 1, 2, 4\}$. The final online configuration, including the selected environment-specific values, is reported in Table 10.

For the offline D4RL locomotion tasks, we tune $\lambda_{\text{coeff}} \in \{1, 2, 3, 5, 10\}$ and $\alpha \in \{1, 2, 3\}$. The selected dataset-specific values are reported in Table 11. For AntMaze, we decouple the actor and critic coefficients and tune $\lambda_a \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$, $\lambda_c \in \{0.0002, 0.001, 0.005, 0.01, 0.05\}$, and $\alpha \in \{0.001, 0.01, 0.02, 0.1\}$. The selected values for AntMaze are reported in Table 12.

Table 10: Hyperparameters for online Atari and Fetch Robotics experiments (Section 5.3).

Environment	α
Atari MontezumaRevenge	1
Atari Venture	2
Atari Frogger	2
Atari Freeway	1
Fetch Reach	1
Fetch Push	1
Fetch Slide	1
Fetch Pick and Place	1

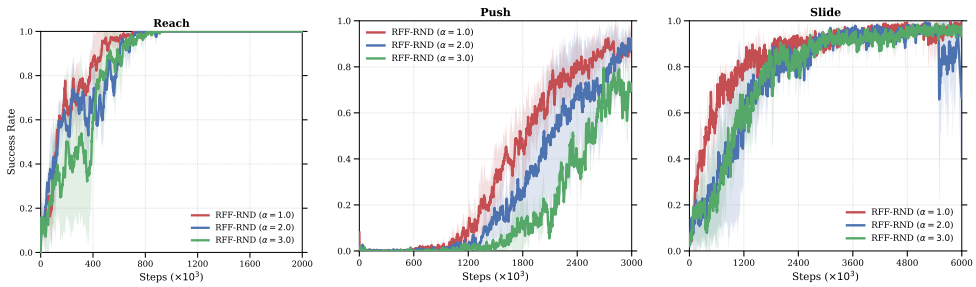
Table 11: Hyperparameters for D4RL Locomotion offline datasets (Section 5.3).

Dataset Name	λ_{coeff}	α
halfcheetah-random	5	1
halfcheetah-medium	2	3
halfcheetah-expert	10	3
halfcheetah-medium-expert	10	3
halfcheetah-medium-replay	1	2
halfcheetah-full-replay	1	3
hopper-random	1	2
hopper-medium	5	3
hopper-expert	5	1
hopper-medium-expert	1	1
hopper-medium-replay	1	1
hopper-full-replay	3	3
walker2d-random	5	3
walker2d-medium	5	3
walker2d-expert	5	2
walker2d-medium-expert	5	2
walker2d-medium-replay	3	3
walker2d-full-replay	1	3

Table 12: Hyperparameters for AntMaze offline datasets (Section 5.3).

Dataset Name	λ_a	λ_c	α
antmaze-umaze	0.005	0.005	0.02
antmaze-umaze-diverse	0.01	0.001	0.1
antmaze-medium-play	2.0	0.05	0.01
antmaze-medium-diverse	0.001	0.0002	0.001
antmaze-large-play	0.05	0.001	0.001
antmaze-large-diverse	0.005	0.001	0.001

Role of α in exploration and anti-exploration. In online sparse-reward exploration, the preferred behavior can be different. Smaller α induces a sharper novelty boundary, which helps the agent distinguish nearby but unexplored regions more quickly. We validate this on Fetch Robotics tasks in Figure 12. In particular, $\alpha = 2$ and $\alpha = 3$ reach the 50% success threshold 76.0% and 77.5% later than $\alpha = 1$, respectively. These results show that α affects both offline anti-exploration and online exploration, but the desirable scale depends on the task setting. Overall, the results justify using a small grid search for α and λ_{coeff}, followed by dataset-specific hyperparameter selection.



(a) Reach (b) Push (c) Slide

Figure 12: Effect of α on exploration performance in Fetch Robotics tasks.

We also examine the role of α more directly because it controls the spatial scale of the kernel-induced novelty estimate. In offline RL, larger α produces smoother uncertainty estimates, which can be useful when the dataset has broad support. Table 13 supports this trend on D4RL random datasets, where $\alpha = 2$ achieves the highest average score. However, the best value is not identical across all environments, as halfcheetah-random performs best at $\alpha = 1$. This again indicates that α affects anti-exploration behavior but does not follow a universal monotonic pattern.

Table 13: Performance comparison across different α values on D4RL random datasets.

Dataset	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
halfcheetah-random	24.6 ± 1.6	$\mathbf{29.3 \pm 1.7}$	29.0 ± 1.2
hopper-random	26.5 ± 0.2	30.8 ± 0.3	$\mathbf{31.4 \pm 0.1}$
walker2d-random	18.9 ± 0.7	17.3 ± 0.8	$\mathbf{21.7 \pm 1.6}$
Average	23.3	25.8	**27.4**