Mitigating Overgeneralization in RND via Spectral Target Design

Minseok Jeong[1], Yechan Lee[1], Hyewon Choi[1], Jeongyong Yang[2], SooJean Han[1]

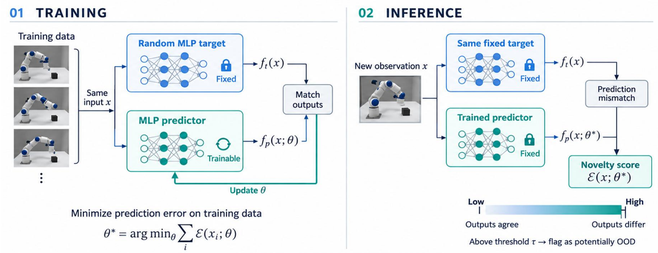
[1] KAIST, [2]University of Washington

Code | Paper | LinkedIn

Background: random network distillation

Given an input domain $\mathcal{X}$, and data points $\mathcal{D} = \{x_i\}_{i=1,\dots,n} \subset \mathcal{X}$,
Random network distillation (RND) consists of

- a **fixed target network** $f_t : \mathcal{X} \to \mathbb{R}^m$ and
- a **trainable predictor network** $f_p : \mathcal{X} \to \mathbb{R}^m$ parameterized by θ.

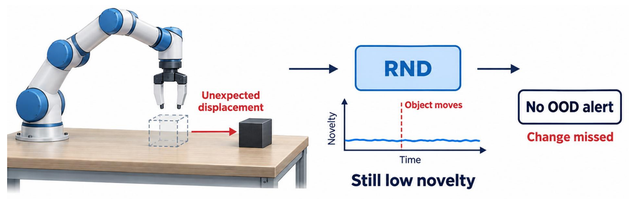


$$\mathcal{E}(x;\theta) = \frac{1}{m}\left\|f_t(x) - f_p(x;\theta)\right\|_2^2$$

It can be used for **novelty** estimation, **out-of-distribution (OOD)** detection, and exploration or anti-exploration in **online and offline RL**.

Motivation: overgeneralization

The predictor can easily predict the target in an unfamiliar situation



Why does RND overgeneralize?

RND novelty is **jointly** determined by the target and predictor **kernels**.

MAIN RESULT: Under infinite-width target, slow-training predictor,

$$\mathbb{E}[\mathcal{E}(x;\theta)] = k_t(x,x) - 2A_x^\top k_t(\mathcal{D},x) + A_x^\top K_t A_x$$

Target kernel
What variation does the random target emphasize?

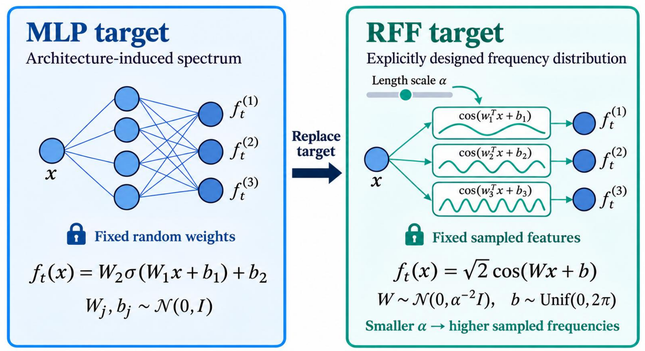
k_t, K_t
target NNGP kernel and Gram matrix

Predictor kernel
How does the predictor generalize from data?

$A_x = \left(\Theta_p(\mathcal{D},\mathcal{D}) + \lambda I\right)^{-1}\Theta_p(\mathcal{D},x)$
NTK-induced prediction weights

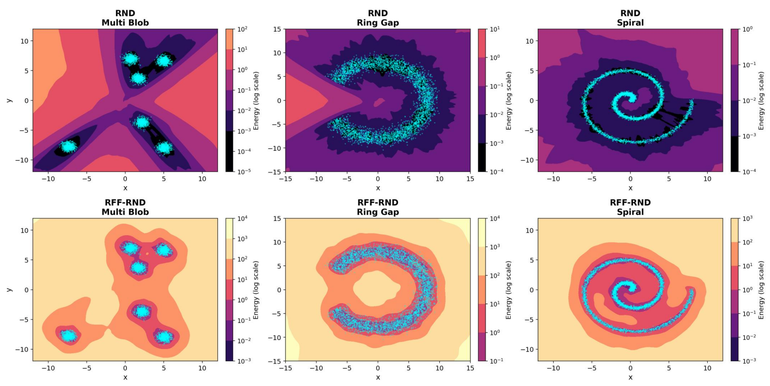
Proposed method: RFF-RND

Replace the fixed MLP target with **random Fourier features** (RFF).

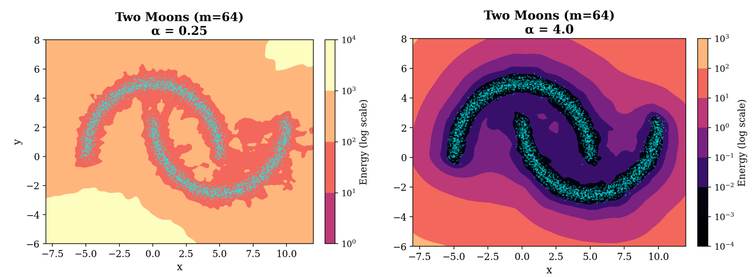


Synthetic validation

This **simple modification** effectively mitigates overgeneralization by enabling the model to distinguish **off-manifold regions.**

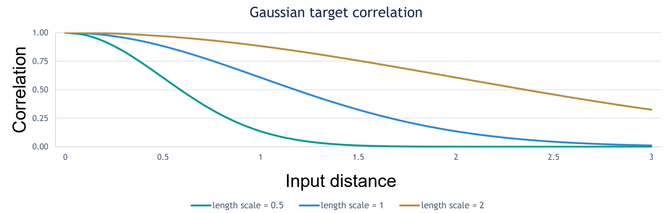


Effect of length scale α



Why does RFF-RND work?

The RFF target allows us to directly control its **sensitivity** to novelty through the length scale parameter α.



Application to downstream RL

The learned RFF-RND can be leveraged in two complementary ways:

- Online RL — **Encouraging** Exploration

Rewarding novel behavior to promote exploration.

$$r_{\text{online}}(x) = r_{\text{task}}(x) + \beta\mathcal{E}(x), \qquad x = (s,a).$$

- Offline RL — **Discouraging** OOD Behaviors

Penalizing unfamiliar behaviors to prevent OOD actions (**anti-exploration**)

$$r_{\text{offline}}(x) = r_{\text{task}}(x) - \beta\mathcal{E}(x), \qquad x = (s,a).$$

The same novelty detector $\mathcal{E}$ promotes exploration in online RL while discouraging OOD behavior in offline RL.

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}	$\text{SAC}_{\text{RFF-RND}}$
halfcheetah-random	29.7 ± 1.4	**31.1 ± 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	**68.3 ± 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	**109.2 ± 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	**109.7 ± 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	**57.0 ± 2.0**
halfcheetah-full-replay	**86.8 ± 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	**32.7 ± 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	**102.3 ± 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	**110.2 ± 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	**112.0 ± 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	**102.7 ± 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	**109.6 ± 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	**22.0 ± 1.4**
walker2d-medium	−0.3 ± 0.2	79.5 ± 3.2	80.7 ± 3.4	82.5 ± 3.6	91.6 ± 2.8	**95.2 ± 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	**114.0 ± 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	**113.2 ± 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	**101.7 ± 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	**111.7 ± 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}	$\text{SAC}_{\text{RFF-RND}}$
antmaze-umaze	0.0	74.0	83.3 ± 4.5	**97.8 ± 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	**88.4 ± 3.9**
antmaze-medium-play	0.0	61.2	64.6 ± 4.9	84.0 ± 4.2	38.5 ± 29.4	**86.2 ± 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	**83.0 ± 3.8**	80.5 ± 5.5
antmaze-large-play	0.0	15.8	42.5 ± 6.5	**60.4 ± 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	**54.4 ± 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