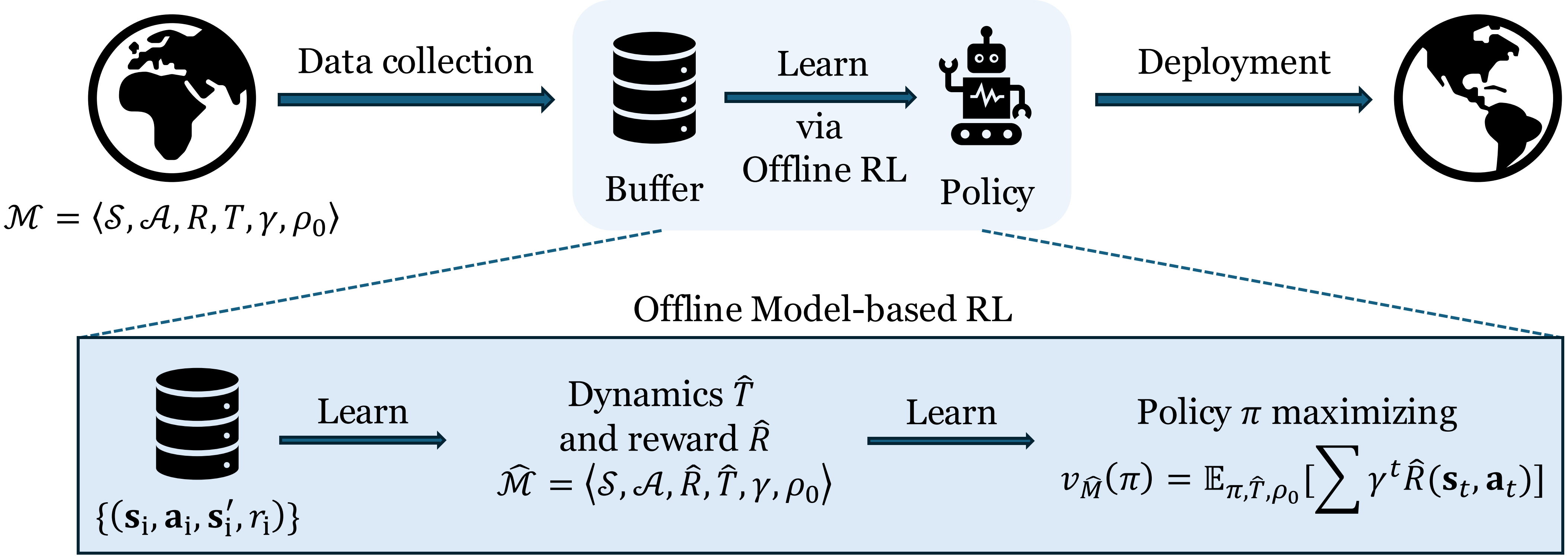
Neural Stochastic Differential Equations for Uncertainty-Aware Offline RL

Cevahir Koprulu[1], Franck Djeumou[2], Ufuk Topcu[1]

[1] The University of Texas at Austin, [2]Rensselaer Polytechnic Institute

TLDR: Neural SDEs for offline model-based RL outperforms SOTA in continuous control benchmarks, particularly in low-quality data regimes.

Offline Model-based RL



Model Exploitation

A pitfall of offline MBRL:

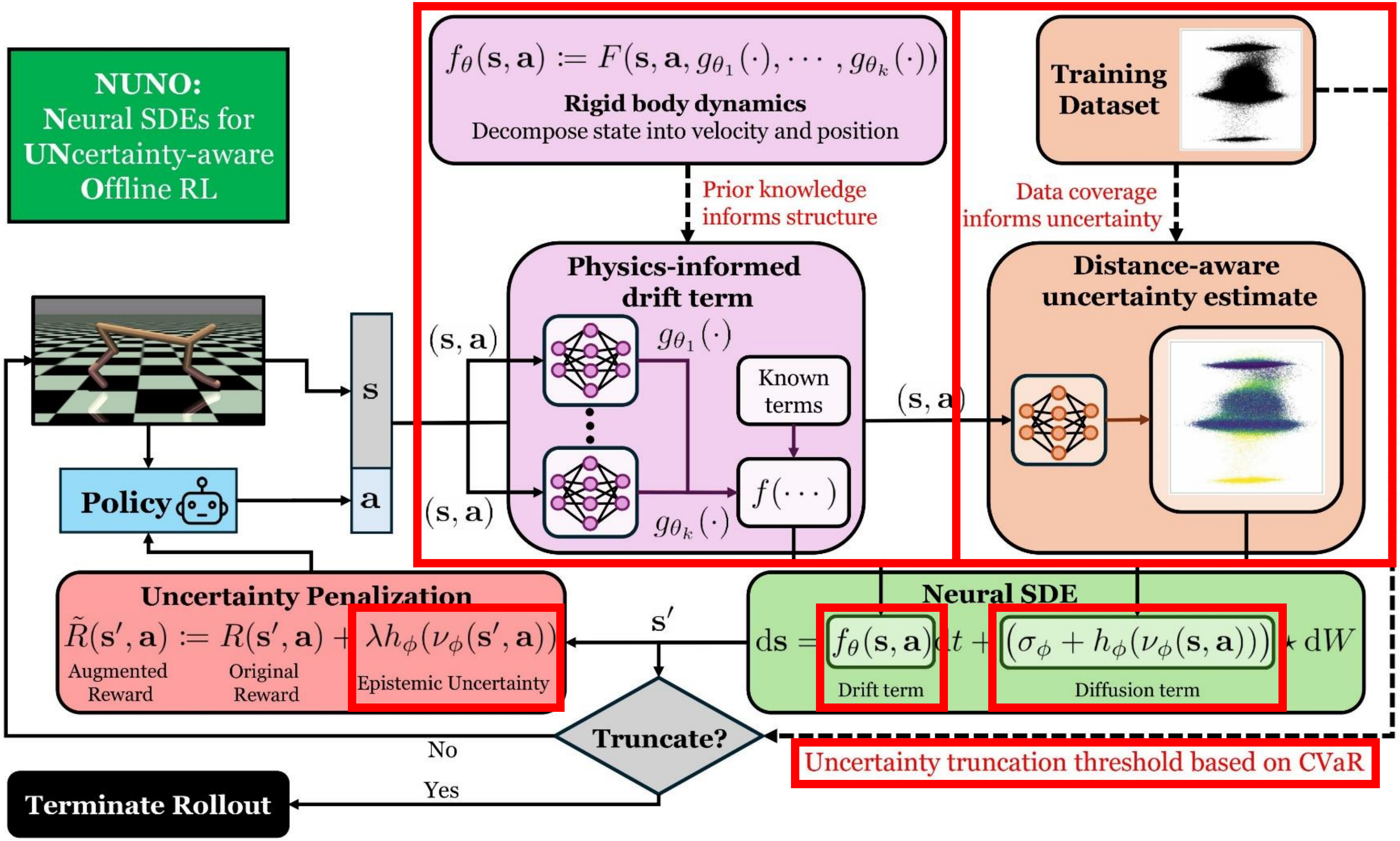
- Estimated return exceeds the true return $\nu_{\widehat{\mathcal{M}}}(\pi) - \nu_{\mathcal{M}}(\pi) > 0$.
- Policy π learns to exploit the regions of state-action space with high model uncertainty as well as high estimated return.

Popular remedies:

- Penalize policy with respect to model uncertainty [1]
- Truncate generated rollouts with high model uncertainty [2]

[1] Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., ... & Ma, T. (2020). Mopo: Model-based offline policy optimization. Advances in Neural Information Processing Systems, 33, 14129-14142.
[2] Zhang, J., Lyu, J., Ma, X., Yan, J., Yang, J., Wan, L., & Li, X. (2023). Uncertainty-driven trajectory truncation for data augmentation in offline reinforcement learning. In ECAI 2023 (pp. 3018-3025). IOS Press.

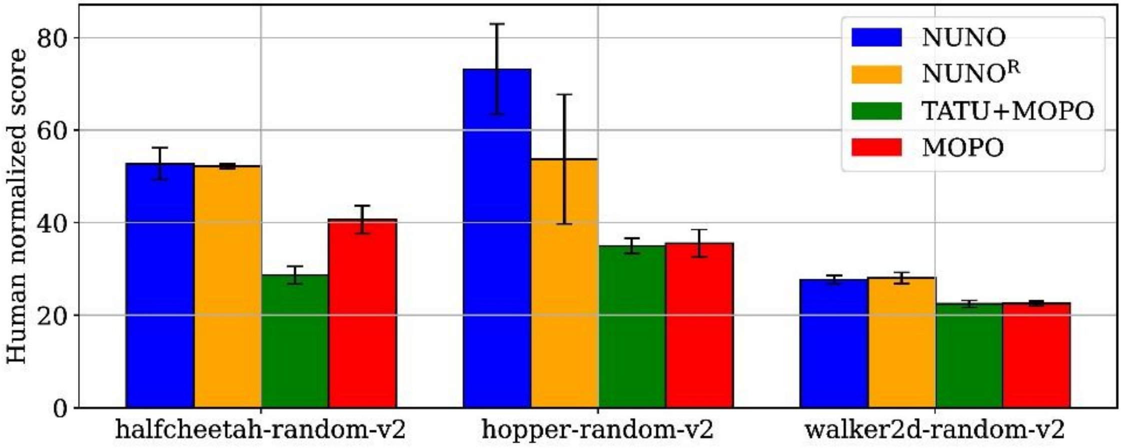
Can neural stochastic differential equations address model exploitation?



NUNO:
Neural SDEs for
UNcertainty-aware
Offline RL
$f_\theta(\mathbf{s},\mathbf{a}) := F(\mathbf{s},\mathbf{a},g_{\theta_1}(\cdot),\cdots,g_{\theta_k}(\cdot))$
Rigid body dynamics
Decompose state into velocity and position
Prior knowledge
informs structure
Training
Dataset
Data coverage
informs uncertainty
Physics-informed
drift term
$(\mathbf{s},\mathbf{a})$
$g_{\theta_1}(\cdot)$
Known
terms
$f(\cdots)$
$g_{\theta_k}(\cdot)$
Distance-aware
uncertainty estimate
s
a
Policy
Uncertainty Penalization
$\tilde{R}(\mathbf{s}',\mathbf{a}) := R(\mathbf{s}',\mathbf{a}) + \lambda h_\phi(\nu_\phi(\mathbf{s}',\mathbf{a}))$
Augmented
Reward
Original
Reward
Epistemic Uncertainty
$\mathbf{s}'$
Neural SDE
$\mathrm{d}\mathbf{s} = f_\theta(\mathbf{s},\mathbf{a})\mathrm{d}t + (\sigma_\phi + h_\phi(\nu_\phi(\mathbf{s},\mathbf{a}))) \star \mathrm{d}W$
Drift term
Diffusion term
Truncate?
No
Yes
Terminate Rollout
Uncertainty truncation threshold based on CVaR

Empirical Results

TLDR 1: NUNO outperforms SOTA in low-quality datasets by up to 93%.



Low quality datasets in D4RL: MOPO and TATU+MOPO penalize and truncate, rollouts based on uncertainty estimates from Gaussian ensembles, whereas NUNO achieves SOTA results in all environments via distance-aware uncertainty estimates of learned NSDEs.

TLDR 2: NUNO matches or surpasses their performance by up to 55% in high-quality ones.

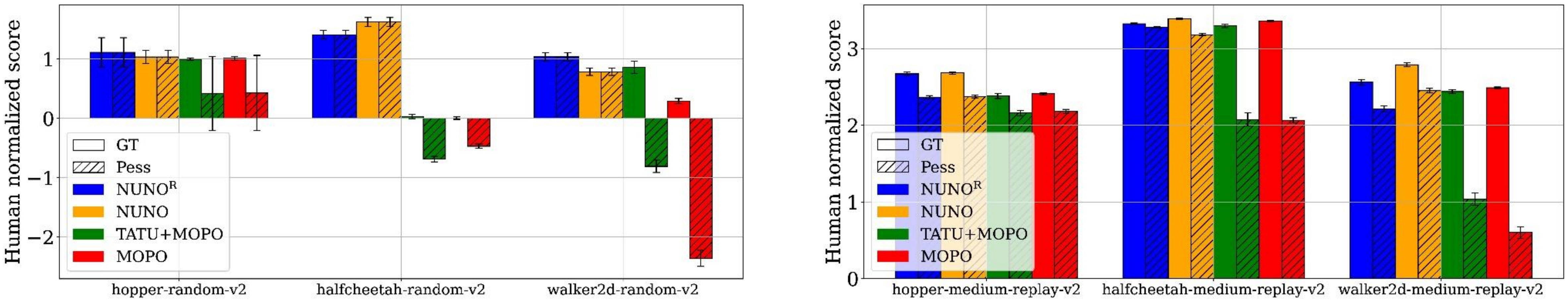
Benchmark 1: D4RL

Task	NUNO (Ours)	NUNO^R (Ours)	MOBILE	MOPO^T	MOPO	COMBO	MOReL	RAMBO	EDAC
hc-r	**52.7±3.4**	**52.2±0.5**	39.3±3.0	33.3	35.9	38.8	38.9	39.5	28.4
hp-r	**73.2±9.8**	**53.7±13.9**	31.9±0.6	31.9	16.7	17.9	38.1	25.4	25.3
wk-r	**27.7±0.9**	**28.1±1.2**	17.9±6.6	10.4	4.2	7.0	16.0	0.0	16.6
hc-m	68.8±0.4	64.7±0.5	74.6±1.2	61.9	73.1	54.2	60.7	**77.9**	65.9
hp-m	104.6±0.2	104.4±0.3	**106.6±0.6**	104.3	38.3	97.2	84.0	87.0	101.6
wk-m	85.4±0.9	**92.6±1.3**	87.7±1.1	77.9	41.2	81.9	72.8	84.9	**92.5**
hc-mr	66.5±0.2	64.6±0.3	**71.7±1.2**	67.2	69.2	55.1	44.5	68.7	61.3
hp-mr	**107.8±1.2**	**106.6±1.9**	103.9±1.0	104.4	32.7	89.5	81.8	99.5	101.0
wk-mr	**97.0±1.4**	**101.1±3.9**	89.9±1.5	75.3	73.7	56.0	40.8	89.2	87.1
hc-me	97.0±0.5	95.8±1.2	**108.2±2.5**	74.1	70.3	90.0	80.4	95.4	**106.3**
hp-me	**112.2±0.3**	**111.9±0.5**	**112.6±0.2**	107.0	60.6	111.1	105.6	88.2	110.7
wk-me	113.2±0.5	112.6±0.6	**115.2±0.7**	107.9	77.4	103.3	107.5	56.7	**114.7**
Average	**83.8**	82.4	80.0	71.3	49.4	66.8	64.3	67.7	76.0

Benchmark 2: NeoRL

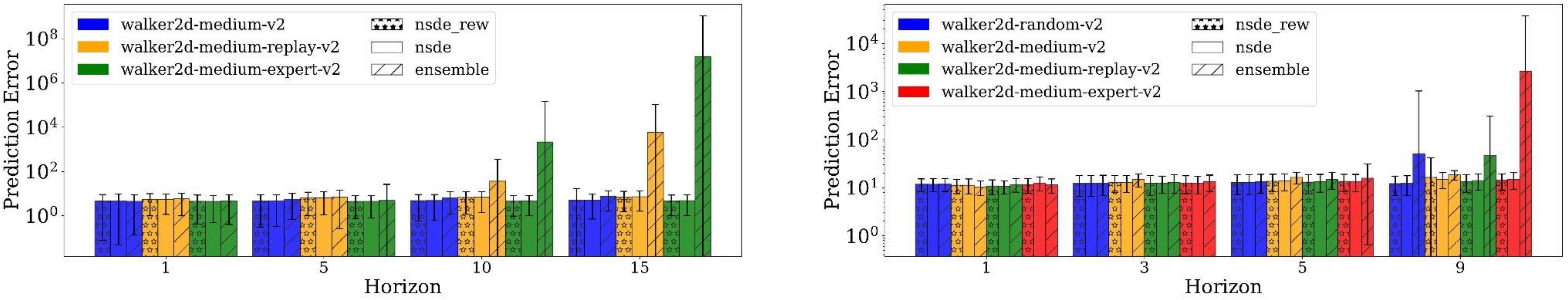
Task	NUNO (Ours)	NUNO^R (Ours)	MOBILE	MOPO	BC	CQL	TD3+BC	EDAC
hc-L	52.5±0.6	**58.4±0.5**	54.7±3.0	40.1	29.1	38.2	30.0	31.3
hp-L	**26.9±3.8**	**26.4±6.8**	17.4±3.9	6.2	15.1	16.0	15.8	18.3
wk-L	**52.5±2.4**	**49.4±1.9**	37.6±2.0	11.6	28.5	44.7	43.0	40.2
hc-M	73.4±0.6	**78.8±0.8**	**77.8±1.4**	62.3	49.0	54.6	52.3	54.9
hp-M	**103.3±2.2**	92.3±1.7	51.1±13.3	1.0	51.3	64.5	70.3	44.9
wk-M	**65.8±0.4**	**49.4±16.9**	62.2±1.6	39.9	48.7	57.3	58.5	57.6
hc-H	**85.2±0.6**	**84.9±0.4**	**83.0±4.6**	65.9	71.3	77.4	75.3	81.4
hp-H	**103.0±3.1**	**97.9±5.5**	**87.8±26.0**	11.5	43.1	76.6	75.3	52.5
wk-H	**72.9±1.6**	**74.5±1.6**	**74.9±3.4**	18.0	72.6	75.3	69.6	**75.5**
Average	**70.6**	68	60.7	28.5	45.4	56.1	54.5	50.7

TLDR 3: NUNO constructs pessimistic learned MDPs that are less conservative.



Model exploitation: Evaluation in rollouts from learned dynamics models in (a) random and (b) medium-replay tasks. We report the average score per step with (pessimistic, Pess) and without (groundtruth, GT) uncertainty penalization.

TLDR 4: Neural SDEs are more accurate than Gaussian ensembles over longer horizons.



Model analysis: We illustrate the evolution of model prediction error in different datasets for D4RL Walker2d. (a) In-distribution: Evaluation of the datasets in which the models are trained. (b) Out-of-distribution: Evaluation of models, trained via random, in trajectories from other datasets.

Thank you!

Cevahir Koprulu
Email: cevahir.koprulu@utexas.edu
Website: https://cevahir-koprulu.github.io/