NEURAL INFORMATION PROCESSING SYSTEMS NeurIPS 2026

Distributed, Intelligent, and Scalable COmputing DISC Lab

Robust Hopfield Decision Transformer

Andreia Podasca and Anup Das

Drexel University, Philadelphia

AMBITION CAN'T WAIT

Executive Summary

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.
- **Problem:** Softmax attention forms each representation in a single step, with no explicit mechanism to limit corrupted observations (sensor noise); robust variants (RDT, ARDT) change only the training.
- **Solution (RHDT):** Replace softmax attention with causally masked, multi-head Hopfield attention; regularize training with a gradient penalty and a key-orthogonality penalty.
- **Key Contribution:** First Hopfield attention in Decision Transformers (to our knowledge); Theorems 1 and 2 motivate the two regularizers; the mechanism is measured directly.
- **Key Findings:** At noise σ = 0.3 on six D4RL tasks: +14% mean score vs RDT (47.3 vs 41.4) and +62% CVaR-10 vs ARDT (18.5 vs 11.4); best mean score in 24/24 settings, best CVaR-10 in 22/24. Cost vs RDT: 2.8× time per training step, 1.5× per action.

CVaR-10 = mean normalized return of the worst 10% of episodes · σ = noise std as a fraction of each state dimension's dataset std
RDT = Robust Decision Transformer · ARDT = Adversarially Robust Decision Transformer · D4RL = offline RL benchmark

DREXEL UNIVERSITY 1891
Distributed, Intelligent, and Scalable COmputing
DISC Lab

Outline

- Background and Motivation
- RHDT: Robust Hopfield Decision Transformer
  - Hopfield Attention for Decision Making
  - Regularizers and Theory
- Key Contributions
- Evaluation
- Conclusion

Decision Transformers Under Observation Noise

At test time, sensor noise corrupts every observed state:

$$\tilde{s}_t = s_t + \delta, \quad \delta \sim \mathcal{N}(0, (\sigma \cdot \mathrm{std}(s))^2)$$

$\sigma \in \{0, 0.1, 0.2, 0.3\}$: noise std as a fraction of each state dimension's dataset std (§5.1)

Decision Transformer reads its context in one softmax step:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(QK^\top / \sqrt{d_k})\, V$$

a weighted average of values in a single step: no mechanism to correct a corrupted query (Eq. 1)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Decision Transformers Under Observation Noise

At test time, sensor noise corrupts every observed state:

$$\tilde{s}_t = s_t + \delta, \quad \delta \sim \mathcal{N}(0, (\sigma \cdot \mathrm{std}(s))^2)$$

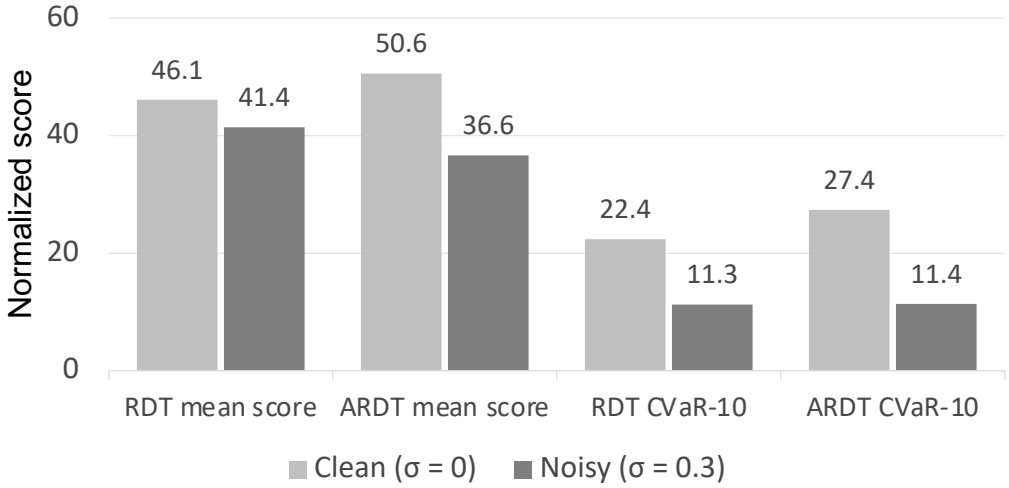
$\sigma \in \{0, 0.1, 0.2, 0.3\}$: noise std as a fraction of each state dimension's dataset std (§5.1)

Decision Transformer reads its context in one softmax step:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(QK^\top / \sqrt{d_k})\, V$$

a weighted average of values in a single step: no mechanism to correct a corrupted query (Eq. 1)

Existing robust DTs lose their tail under noise



Task-averaged score ↑ of the robust baselines, six D4RL tasks (paper Table 12a). CVaR-10 = mean of the worst 10% of episodes.

RDT and ARDT change only the training: attention stays single-step softmax

They keep only 50% and 42% of their CVaR-10 at σ = 0.3 (paper Table 11). Goal: correct queries inside attention

Distributed, Intelligent, and Scalable COmputing DISC Lab

Background: Modern Hopfield Networks

An MHN stores patterns $X = [x_1, \ldots, x_N]$ and defines an energy over a query ξ:

$$E(\xi) = -\,\mathrm{lse}(\beta, X^T\xi) + \tfrac{1}{2}\,\xi^T\xi + \mathrm{const}$$

lse = log-sum-exp · β = inverse temperature (Eq. 2)

Each update lowers the energy and moves the query toward a stored pattern (an attractor):

$$\xi^{(t+1)} = X\,\mathrm{softmax}(\,\beta\,X^T\,\xi^{(t)}\,) \qquad (\mathrm{Eq.\ 3})$$

One update, $\beta = 1/\sqrt{d_k}$

recovers the softmax attention weights of DT: keys are the stored patterns, queries are the probes

Iterated updates

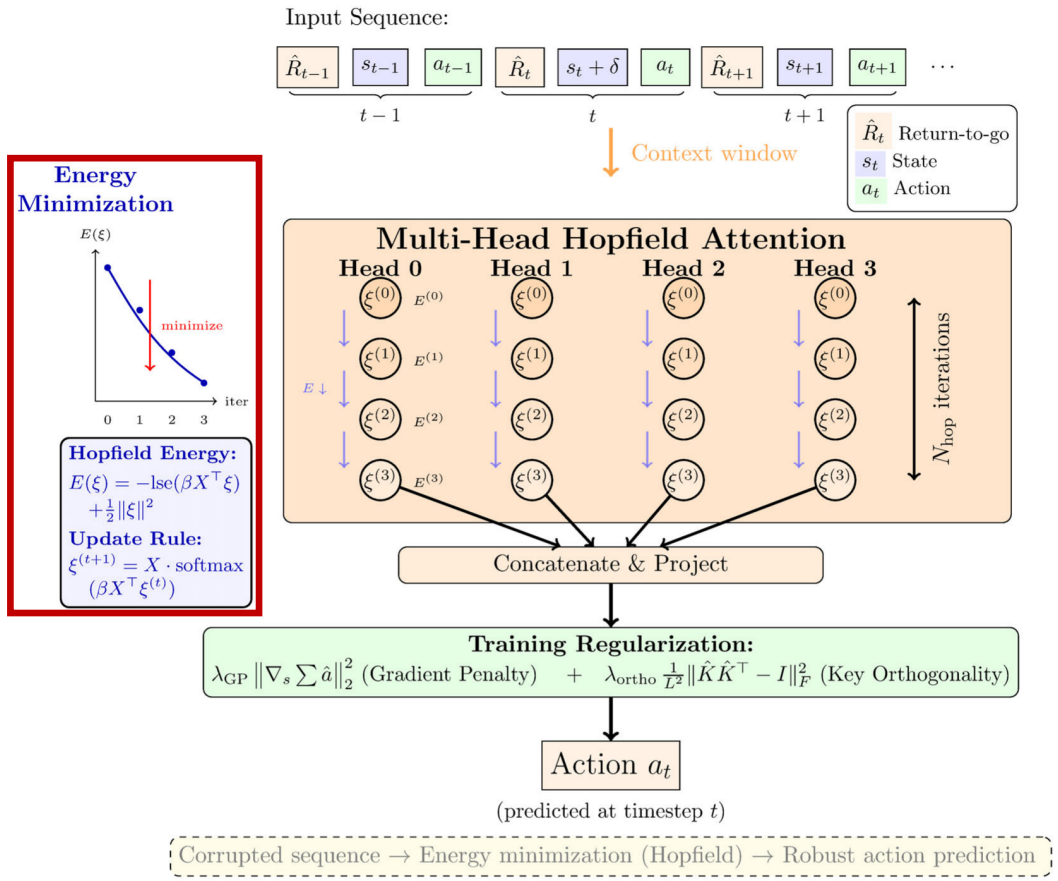
refine the query toward a fixed point near a stored pattern: a way to pull a corrupted query back

DREXEL UNIVERSITY 1891
Distributed, Intelligent, and Scalable COmputing
DISC Lab

Outline

- Background and Motivation
- RHDT: Robust Hopfield Decision Transformer
  - Hopfield Attention for Decision Making
  - Regularizers and Theory
- Key Contributions
- Evaluation
- Conclusion

RHDT: Robust Hopfield Decision Transformer



Paper Fig. 1 (it draws 3 updates; all main results use $N_{hop} = 1$)

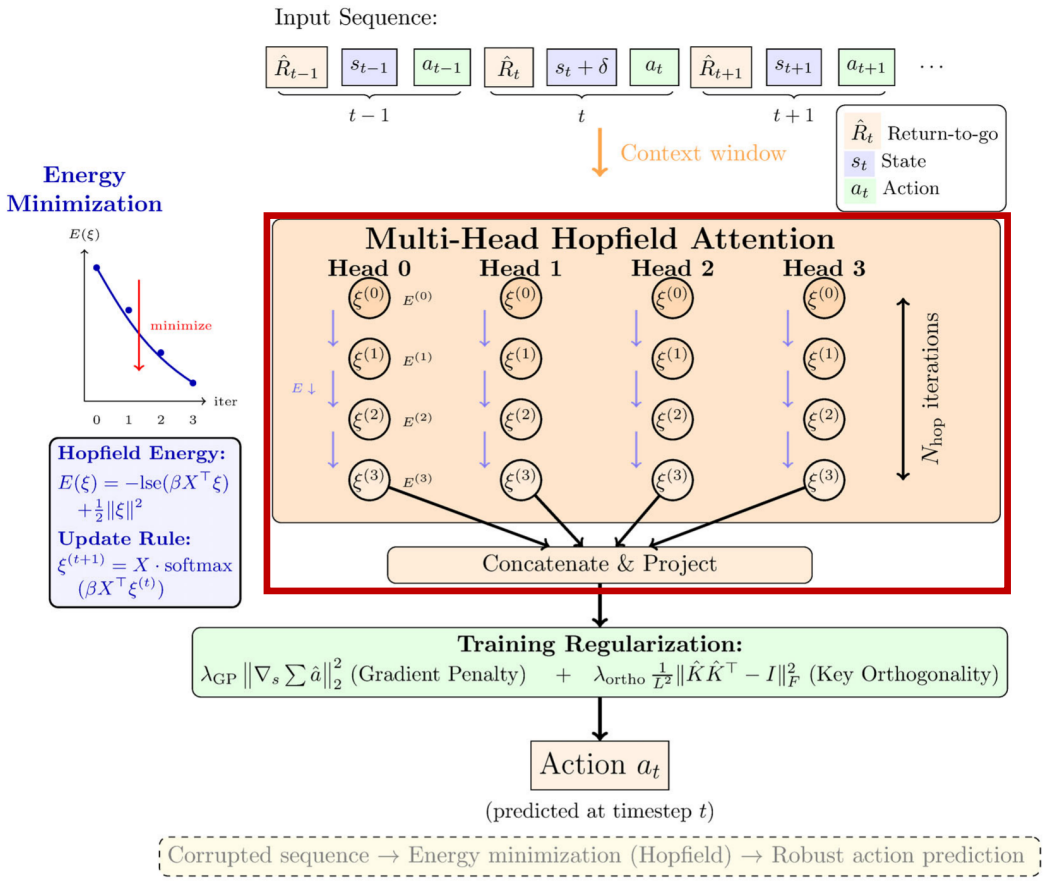
1. Hopfield attention with a causal mask

$$\xi^{(t+1)} = K\,\mathrm{softmax}(\beta K^\top \xi^{(t)}), \quad t = 0, \ldots, N-1$$
$$\mathrm{output} = V\,\mathrm{softmax}(\beta K^\top \xi^{(N)}), \quad N = N_{\mathrm{hop}}$$

- **Stored patterns:** the attention keys K of the context window
- **Refine, then read out:** N_{hop} updates move each query toward its fixed point; $N_{hop} = 1$ in all main results
- **Causal mask:** each position attends only to the past, so actions are predicted autoregressively

Distributed, Intelligent, and Scalable COmputing DISC Lab

RHDT: Robust Hopfield Decision Transformer



Paper Fig. 1 (it draws 3 updates; all main results use $N_{hop} = 1$)

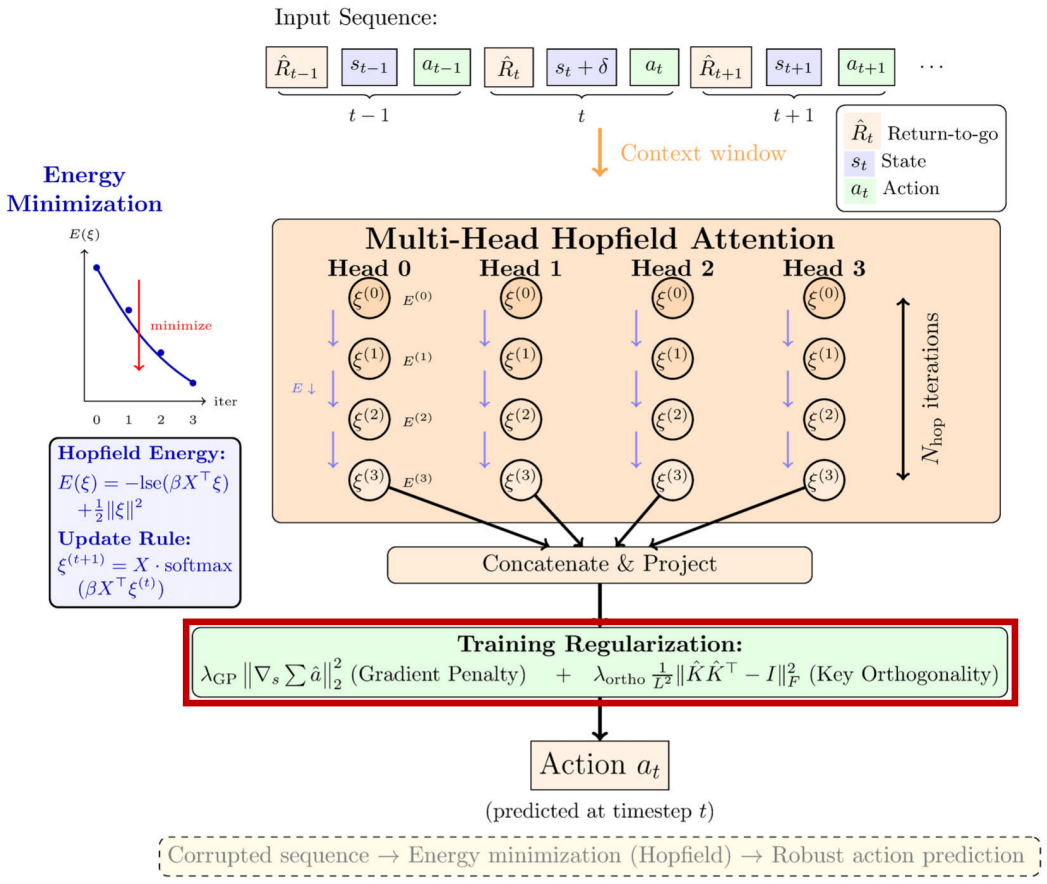
2. Multi-head retrieval (H = 4)

$$\text{attention output} = W_O [\, o_1; o_2; o_3; o_4 \,]$$

- **Independent retrievals:** each head runs its own Hopfield update with separate projections W_Q, W_K, W_V
- **Concatenate and project:** a learned weighted sum of head outputs o_h
- **Why:** combining several retrievals can reduce sensitivity to individual perturbations (baselines use 1 head)

Distributed, Intelligent, and Scalable COmputing DISC Lab

RHDT: Robust Hopfield Decision Transformer



Paper Fig. 1 (it draws 3 updates; all main results use $N_{hop} = 1$)

3. Two training regularizers

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda_{\text{GP}}\,\mathcal{L}_{\text{GP}} + \lambda_{\text{ortho}}\,\mathcal{L}_{\text{ortho}}$$

- **Gradient penalty** $\mathcal{L}_{GP}$: squared gradient of the predicted actions with respect to the input states → limits sensitivity (Eq. 5)
- **Key orthogonality** $\mathcal{L}_{ortho} = (1/L^2)\,\|\hat{K}\hat{K}^\top - I\|_F^2$ → pushes stored patterns apart (Eq. 8)
- $\mathcal{L}_{action}$: Gaussian-weighted action loss, as in RDT (Eq. 10)

Attention refines the query; the penalties limit sensitivity and separate patterns

Distributed, Intelligent, and Scalable COmputing DISC Lab

Why It Works: Theory

Theorem 1: Hopfield Attention Sensitivity Bound

$$\| \xi^{(N)}(\xi + \delta) - \xi^{(N)}(\xi) \|_2 \leq (\beta \|K\|_2^2)^N \|\delta\|_2 , \quad N = N_{hop}$$

In plain terms: each Hopfield update can multiply the worst-case effect of a perturbation δ by $\beta\|K\|_2^2$, so the bound grows exponentially with the number of updates when $\beta\|K\|_2^2 > 1$.

What we do about it

- Small, fixed number of updates ($N_{hop} = 1$)
- Gradient penalty limits action sensitivity
- Less similar keys lower the factor $\beta\|K\|_2^2$ for fixed key norms (Corollary 1)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Why It Works: Theory

Theorem 1: Hopfield Attention Sensitivity Bound

$$\| \xi^{(N)}(\xi + \delta) - \xi^{(N)}(\xi) \|_2 \leq (\beta \|K\|_2^2)^N \|\delta\|_2 , \quad N = N_{hop}$$

In plain terms: each Hopfield update can multiply the worst-case effect of a perturbation δ by $\beta\|K\|_2^2$, so the bound grows exponentially with the number of updates when $\beta\|K\|_2^2 > 1$.

Theorem 2: Attractor Basin Radius

$$r_i = \tfrac{1}{2} \min_{j \neq i} \| x_i - x_j \|_2$$

In plain terms: for equal-norm patterns and large β, every query within r_i of pattern x_i is mapped back to x_i. Better-separated keys → larger basins.

What we do about it

- Small, fixed number of updates ($N_{hop} = 1$)
- Gradient penalty limits action sensitivity
- Less similar keys lower the factor $\beta\|K\|_2^2$ for fixed key norms (Corollary 1)

Proposition 1: $\mathcal{L}_{GP}(\theta^*) \leq \ell_0 / \lambda_{GP}$

The root-mean-square penalized gradient on the training data is $O(1/\sqrt{\lambda_{GP}})$: a larger weight bounds average sensitivity more tightly (ℓ_0: action loss of the best constant policy)

Scope: Theorem 2 is a large-β limit, and RHDT uses $\beta = 1/\sqrt{d_k}$; it describes the geometry the penalty targets, not a guarantee (§4.3)

Outline

- Background and Motivation
- RHDT: Robust Hopfield Decision Transformer
  - Hopfield Attention for Decision Making
  - Regularizers and Theory
- Key Contributions
- Evaluation
- Conclusion

Distributed, Intelligent, and Scalable COmputing DISC Lab

What's New about RHDT

1

Hopfield attention inside a Decision Transformer

Causal masking and multi-head Hopfield layers for autoregressive action prediction; to our knowledge, the first use of MHN attention in Decision Transformers.

What's New about RHDT

1

Hopfield attention inside a Decision Transformer

Causal masking and multi-head Hopfield layers for autoregressive action prediction; to our knowledge, the first use of MHN attention in Decision Transformers.

2

Two regularizers motivated by theory

Gradient penalty limits action sensitivity (Proposition 1); key orthogonality separates the stored patterns (Theorem 2) and tightens the sensitivity bound (Corollary 1).

Distributed, Intelligent, and Scalable COmputing DISC Lab

What's New about RHDT

1

Hopfield attention inside a Decision Transformer

Causal masking and multi-head Hopfield layers for autoregressive action prediction; to our knowledge, the first use of MHN attention in Decision Transformers.

2

Two regularizers motivated by theory

Gradient penalty limits action sensitivity (Proposition 1); key orthogonality separates the stored patterns (Theorem 2) and tightens the sensitivity bound (Corollary 1).

3

The mechanism is measured, and its limits are reported

Query correction, local Lipschitz constant, convergence, and an ablation of every component, including where the Hopfield update adds little.

Distributed, Intelligent, and Scalable COmputing DISC Lab

What's New about RHDT

1

Hopfield attention inside a Decision Transformer

Causal masking and multi-head Hopfield layers for autoregressive action prediction; to our knowledge, the first use of MHN attention in Decision Transformers.

2

Two regularizers motivated by theory

Gradient penalty limits action sensitivity (Proposition 1); key orthogonality separates the stored patterns (Theorem 2) and tightens the sensitivity bound (Corollary 1).

3

The mechanism is measured, and its limits are reported

Query correction, local Lipschitz constant, convergence, and an ablation of every component, including where the Hopfield update adds little.

4

+14% mean score and +62% CVaR-10 at σ = 0.3

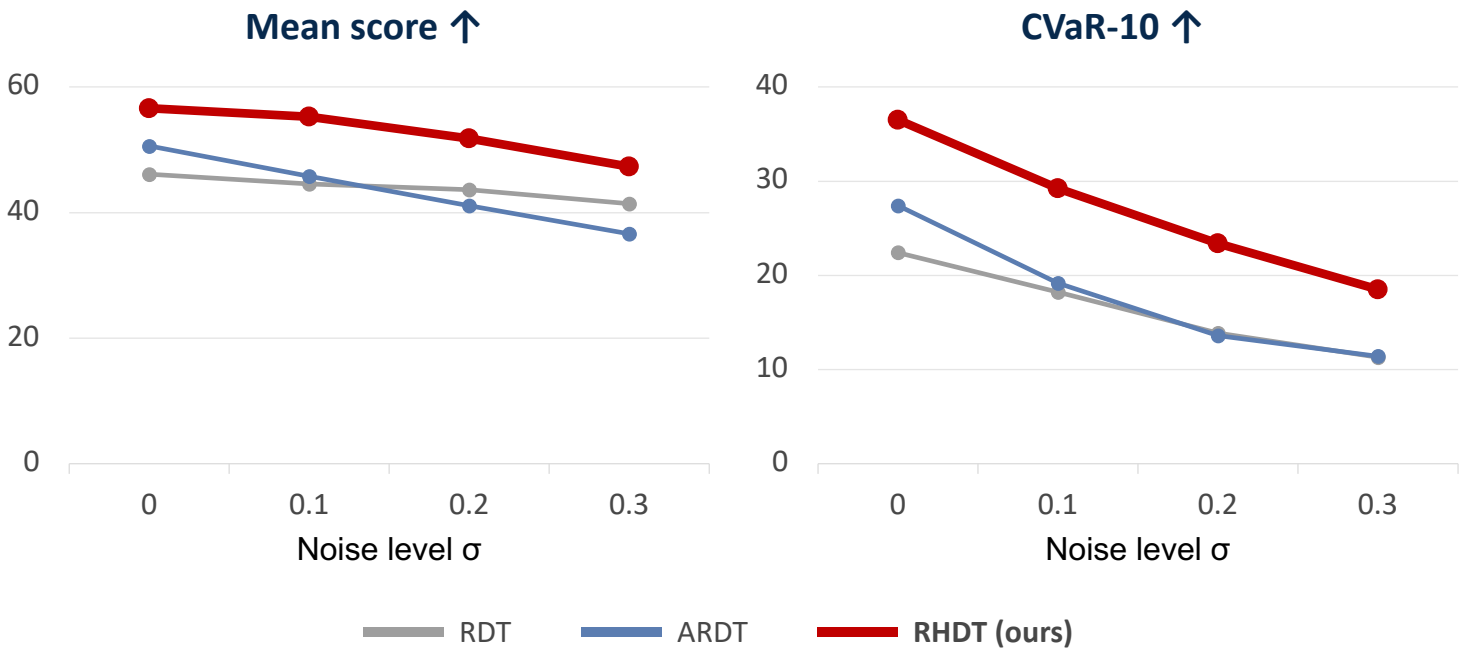
vs. the stronger baseline per metric (RDT / ARDT) on six D4RL tasks · best mean score in 24/24 settings, best CVaR-10 in 22/24 · CVaR-10 = mean return of the worst 10% of episodes

Outline

- Background and Motivation
- RHDT: Robust Hopfield Decision Transformer
  - Hopfield Attention for Decision Making
  - Regularizers and Theory
- Key Contributions
- Evaluation
- Conclusion

Distributed, Intelligent, and Scalable COmputing DISC Lab

Main Results: Gaussian Noise on 6 D4RL Tasks



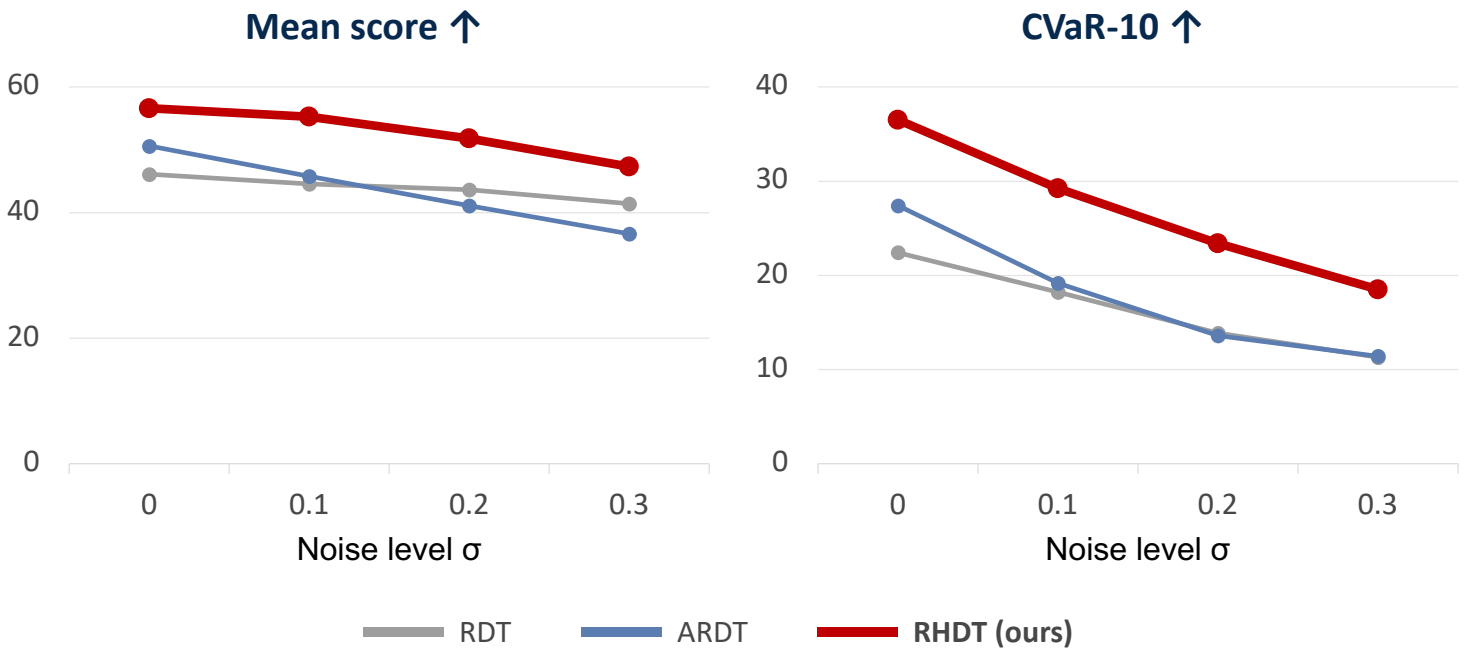
Task-averaged normalized score, six tasks × 6 training seeds (paper Table 12a). DT at σ = 0: 53.9 mean, 31.3 CVaR-10.

Setup

- 6 D4RL locomotion tasks: HalfCheetah, Hopper, Walker2d × medium, medium-replay
- 3-layer transformer, embedding 128, context 20; 6 training seeds × 500 episodes each
- Test-time Gaussian state noise, $\sigma \in \{0, 0.1, 0.2, 0.3\}$
- Baselines: RDT (loss reweighting), ARDT (return relabeling); DT on clean data only
- **Mean score** = normalized return, 0 random, 100 expert (higher is better)
- **CVaR-10** = mean of the worst 10% of episodes (higher is better)
- RHDT: 4 heads, $N_{hop} = 1$, per-task λ_{GP}, λ_{ortho}

Distributed, Intelligent, and Scalable COmputing DISC Lab

Main Results: Gaussian Noise on 6 D4RL Tasks



Task-averaged normalized score, six tasks × 6 training seeds (paper Table 12a). DT at σ = 0: 53.9 mean, 31.3 CVaR-10.

Mean score, σ = 0.3

+14%

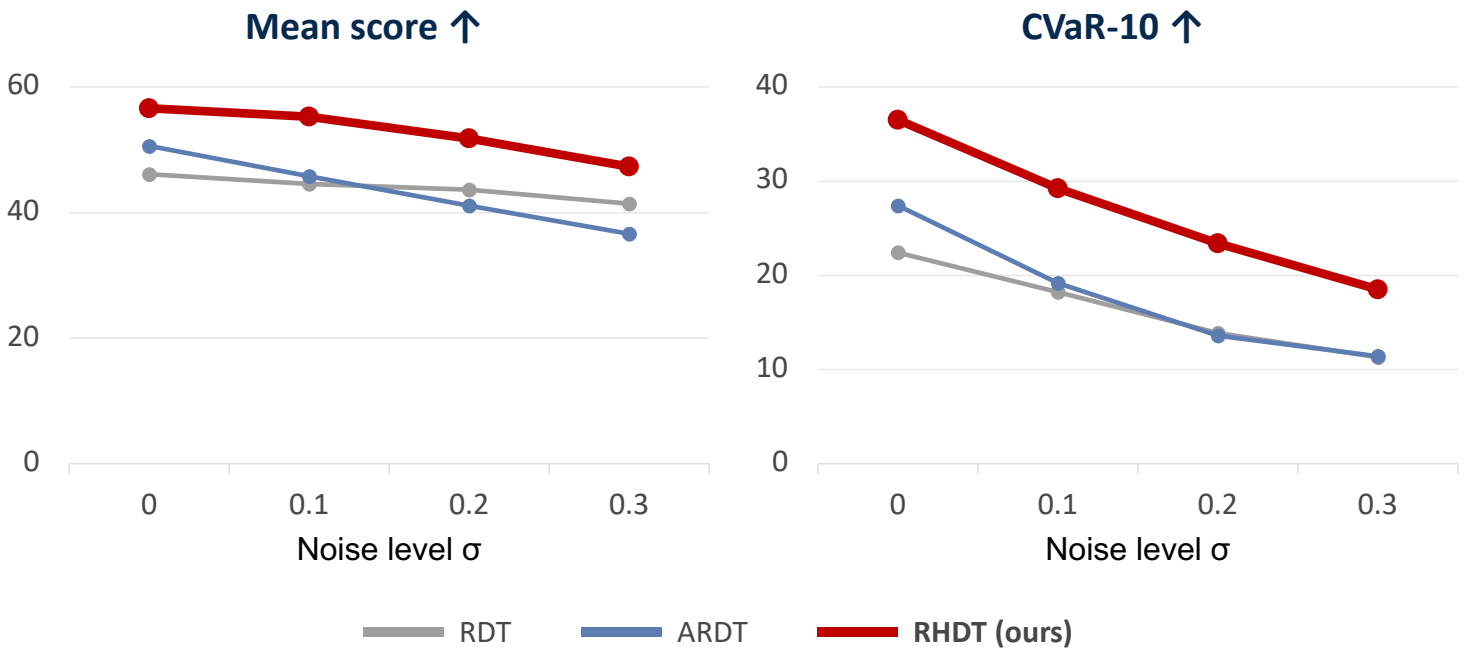
47.3 vs 41.4 (RDT), 36.6 (ARDT)

Setup

- 6 D4RL locomotion tasks: HalfCheetah, Hopper, Walker2d × medium, medium-replay
- 3-layer transformer, embedding 128, context 20; 6 training seeds × 500 episodes each
- Test-time Gaussian state noise, $\sigma \in \{0, 0.1, 0.2, 0.3\}$
- Baselines: RDT (loss reweighting), ARDT (return relabeling); DT on clean data only
- **Mean score** = normalized return, 0 random, 100 expert (higher is better)
- **CVaR-10** = mean of the worst 10% of episodes (higher is better)
- RHDT: 4 heads, $N_{hop} = 1$, per-task λ_{GP}, λ_{ortho}

Distributed, Intelligent, and Scalable COmputing DISC Lab

Main Results: Gaussian Noise on 6 D4RL Tasks



Task-averaged normalized score, six tasks × 6 training seeds (paper Table 12a). DT at σ = 0: 53.9 mean, 31.3 CVaR-10.

Setup

- 6 D4RL locomotion tasks: HalfCheetah, Hopper, Walker2d × medium, medium-replay
- 3-layer transformer, embedding 128, context 20; 6 training seeds × 500 episodes each
- Test-time Gaussian state noise, $\sigma \in \{0, 0.1, 0.2, 0.3\}$
- Baselines: RDT (loss reweighting), ARDT (return relabeling); DT on clean data only
- **Mean score** = normalized return, 0 random, 100 expert (higher is better)
- **CVaR-10** = mean of the worst 10% of episodes (higher is better)
- RHDT: 4 heads, $N_{hop} = 1$, per-task λ_{GP}, λ_{ortho}

Mean score, σ = 0.3

+14%

47.3 vs 41.4 (RDT), 36.6 (ARDT)

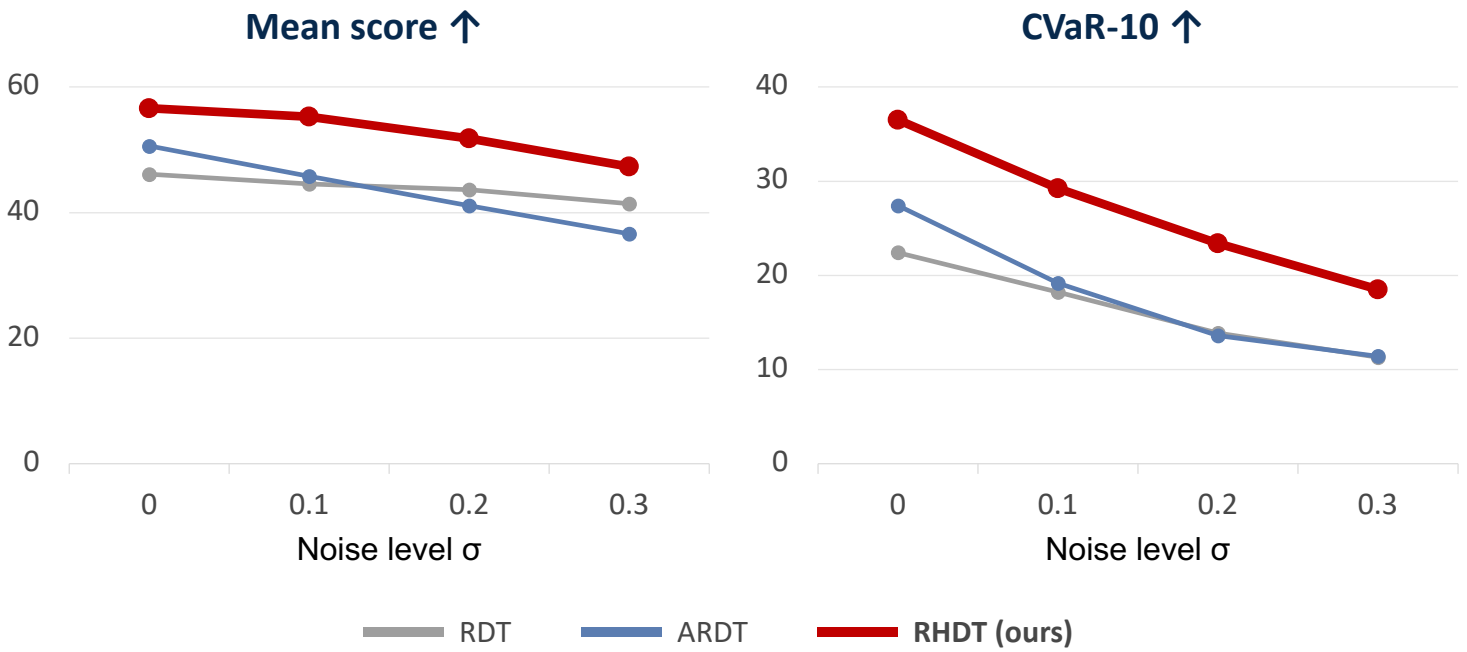
CVaR-10, σ = 0.3

+62%

18.5 vs 11.4 (ARDT), 11.3 (RDT)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Main Results: Gaussian Noise on 6 D4RL Tasks



Task-averaged normalized score, six tasks × 6 training seeds (paper Table 12a). DT at σ = 0: 53.9 mean, 31.3 CVaR-10.

Setup

- 6 D4RL locomotion tasks: HalfCheetah, Hopper, Walker2d × medium, medium-replay
- 3-layer transformer, embedding 128, context 20; 6 training seeds × 500 episodes each
- Test-time Gaussian state noise, $\sigma \in \{0, 0.1, 0.2, 0.3\}$
- Baselines: RDT (loss reweighting), ARDT (return relabeling); DT on clean data only
- **Mean score** = normalized return, 0 random, 100 expert (higher is better)
- **CVaR-10** = mean of the worst 10% of episodes (higher is better)
- RHDT: 4 heads, $N_{hop} = 1$, per-task λ_{GP}, λ_{ortho}

Mean score, σ = 0.3

+14%

47.3 vs 41.4 (RDT), 36.6 (ARDT)

CVaR-10, σ = 0.3

+62%

18.5 vs 11.4 (ARDT), 11.3 (RDT)

Clean score, σ = 0

56.6

vs 53.9 (DT), 50.6 (ARDT), 46.1 (RDT)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Are the Gains Statistically Reliable?

All at σ = 0.3, ↑ higher is better. Intervals and probabilities: 95% stratified bootstrap over 36 runs, 6 tasks × 6 training seeds (paper Table 12). IQM: task average from paper Table 9.

Confidence intervals

mean score, average over runs

47.3

[46.3, 48.4]: above both baselines

RDT 41.4 [39.6, 43.1]
ARDT 36.6 [33.7, 39.0]

Probability of improvement

random RHDT run vs random baseline run

0.85–0.94

every interval excludes 0.5

vs RDT: 0.85 mean, 0.87 CVaR-10
vs ARDT: 0.92 mean, 0.94 CVaR-10

Typical episodes

IQM = mean of the middle 50% of episodes

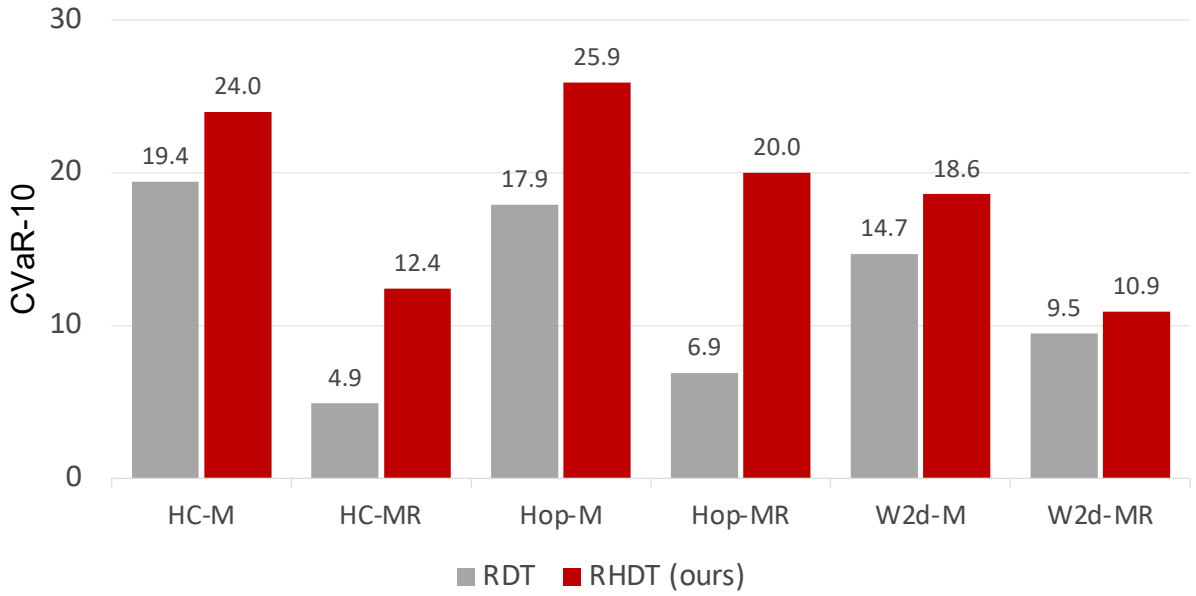
49.3

not driven by a few lucky episodes

RDT 42.3 · ARDT 37.0
+7.0 and +12.3 points

Fair tuning: every method's hyperparameters were selected on clean scores only, never on noisy evaluations (App. B.1)

Generalizes Across Corruptions and Domains

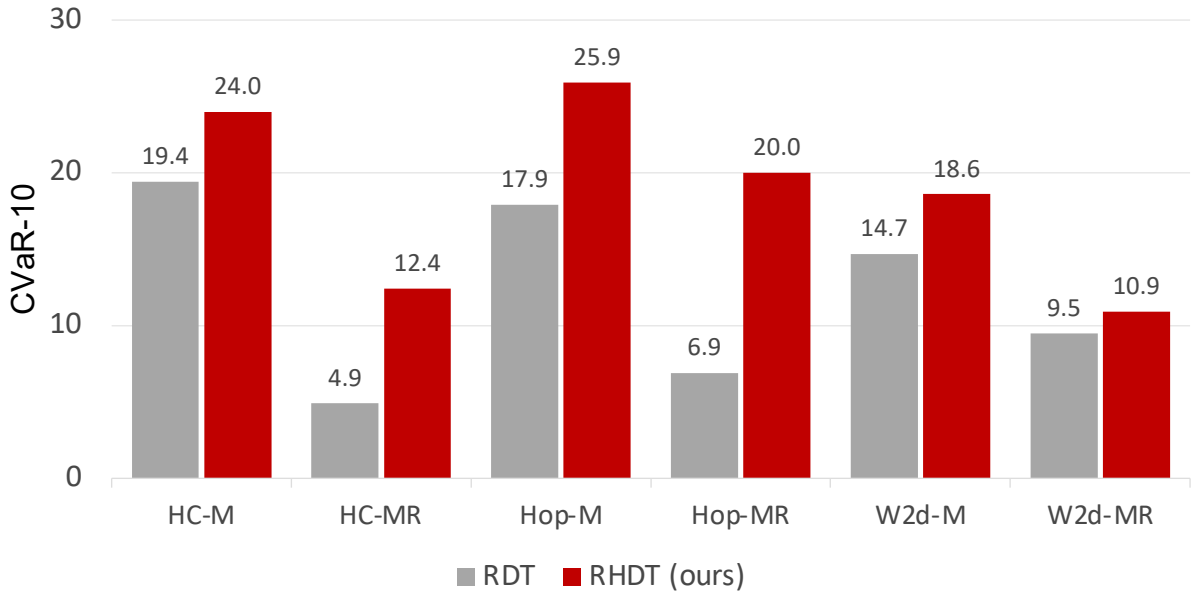


CVaR-10 ↑ under impulse noise: 5× dataset-std noise with probability p = 0.1 per timestep (paper Table 19). HC = HalfCheetah, Hop = Hopper, W2d = Walker2d; M = medium, MR = medium-replay.

Observation dropout (paper Table 18): ahead only at p = 0.1 (+2.7 mean, +2.4 CVaR-10, task average); behind RDT on Hopper Med-Replay (−12.9 mean score at p = 0.2).

Distributed, Intelligent, and Scalable COmputing DISC Lab

Generalizes Across Corruptions and Domains



CVaR-10 ↑ under impulse noise: 5× dataset-std noise with probability p = 0.1 per timestep (paper Table 19). HC = HalfCheetah, Hop = Hopper, W2d = Walker2d; M = medium, MR = medium-replay.

Observation dropout (paper Table 18): ahead only at p = 0.1 (+2.7 mean, +2.4 CVaR-10, task average); behind RDT on Hopper Med-Replay (−12.9 mean score at p = 0.2).

Other domains: mean score under noise (↑)

Setting	Best baseline	RHDT	Δ
Ant Medium (111-dim states)	71.8 (ARDT)	**74.6**	**+2.8**
Ant Med-Replay (111-dim)	68.1 (RDT)	**71.6**	**+3.5**
Kitchen (sparse rewards)	55.2 (RDT)	**57.0**	**+1.8**
Pen-Human (25 human demos)	175.5 (ARDT)	**178.1**	**+2.6**
Atari Qbert (pixels)	6.5 (RDT)	**41.7**	**+35.2**

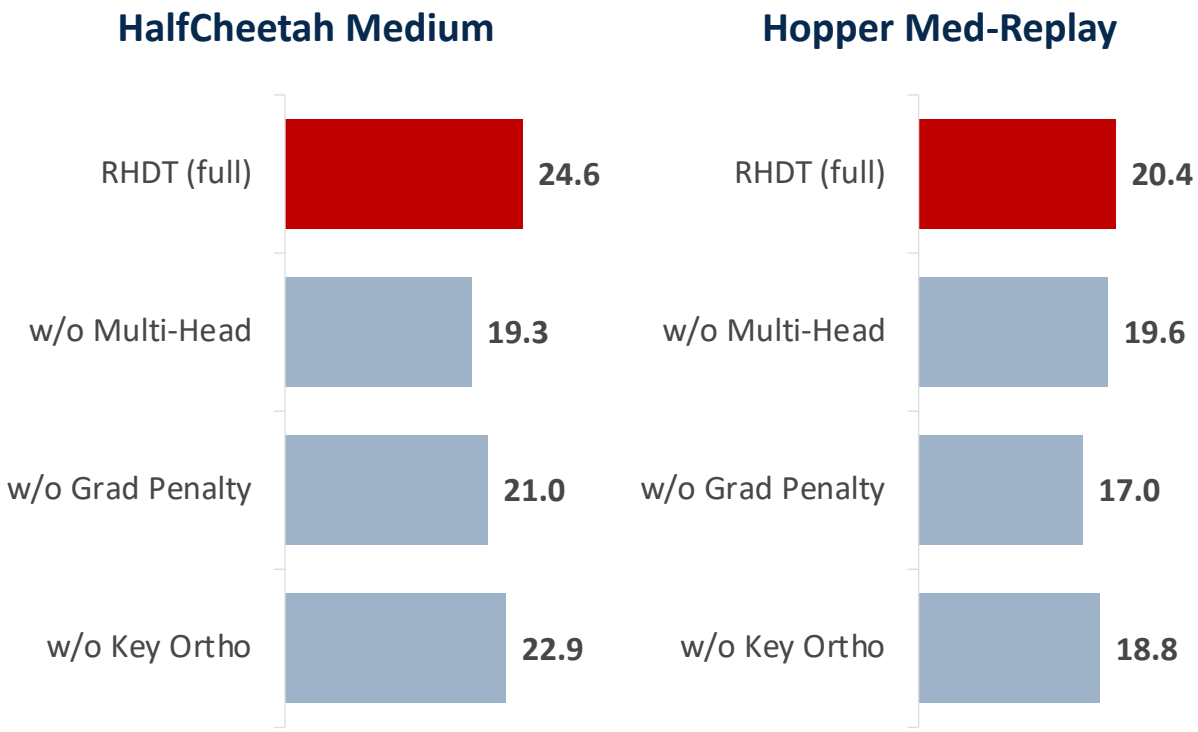
σ = 0.3; Qbert at pixel noise σ = 0.05 (at its strongest, σ = 0.1: 3.6 vs 0.8), one game, no ARDT (paper Tables 14–17)

Highest mean score at the strongest noise, in all five settings

but no CVaR-10 advantage on Kitchen:
8.7 vs 27.1 for RDT at σ = 0.3

Distributed, Intelligent, and Scalable COmputing DISC Lab

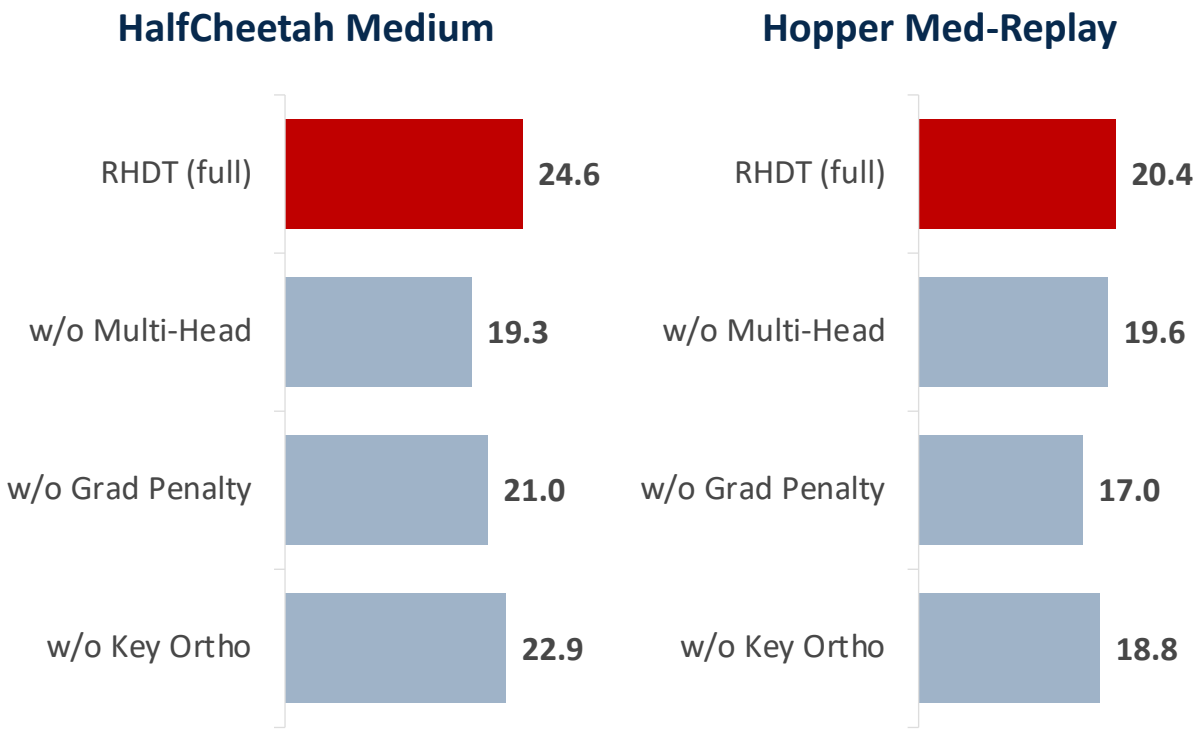
Ablation: What Each Component Adds



CVaR-10 ↑ at σ = 0.3, one component removed at a time, mean over 6 training seeds (paper Table 2)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Ablation: What Each Component Adds



CVaR-10 ↑ at σ = 0.3, one component removed at a time, mean over 6 training seeds (paper Table 2)

Gradient penalty: most consistent

Without it: mean score −1.7 and −5.3, CVaR-10 −3.6 and −3.4 (HalfCheetah, Hopper)

Multi-head: helps on both tasks

One head: CVaR-10 24.6 → 19.3 on HalfCheetah; mean score 52.2 → 45.6 on Hopper

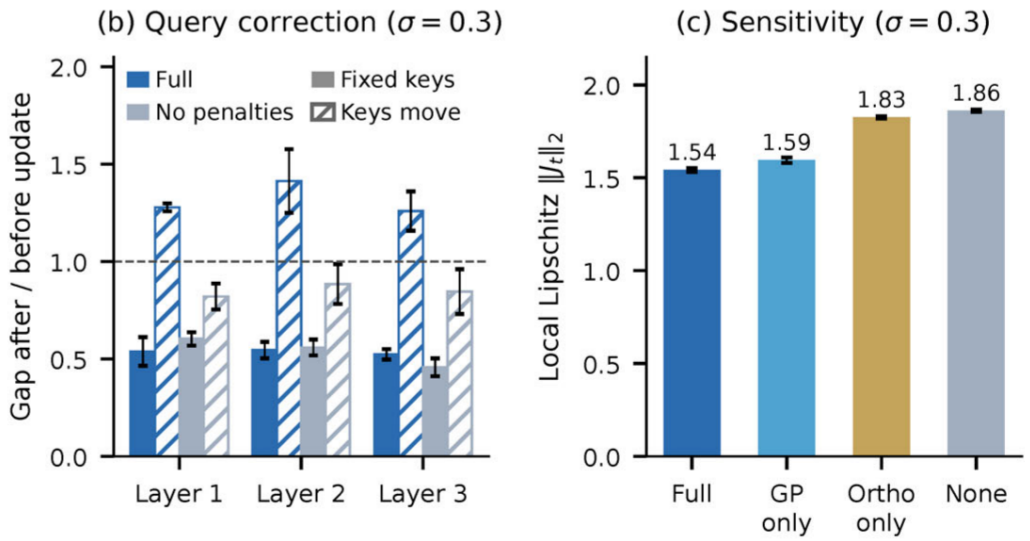
Key orthogonality: smallest effect

At most 1.7 points under noise; on Hopper it lowers clean CVaR-10 (31.8 vs 41.4 without it)

Full model is best in every column at σ = 0.3

Distributed, Intelligent, and Scalable COmputing DISC Lab

The Mechanism, Measured, and Its Cost



HalfCheetah Medium at σ = 0.3, 6 training seeds (paper Fig. 2b, 2c). Lower is better ↓ in both panels.

Query deviation after one Hopfield update, keys fixed

0.52–0.55× of before

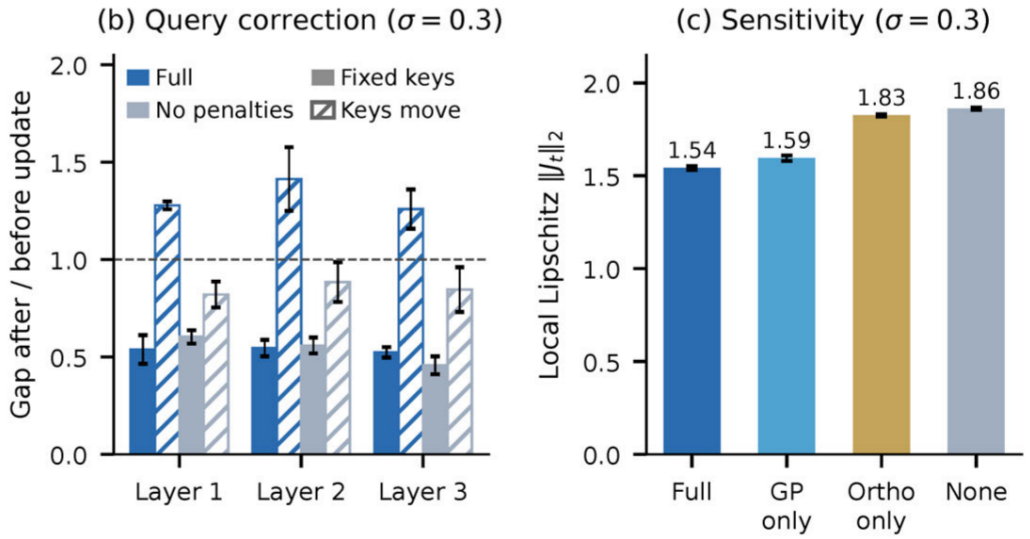
in every layer · gradient penalty: local Lipschitz ≈ 15% lower (1.54 vs 1.83)

Honest limits (§5.5, paper Table 7)

- Noise also moves the keys: in the full forward pass the deviation grows (ratio 1.26–1.41)
- Without the update ($N_{hop} = 0$): 45.7 mean, 18.0 CVaR-10 vs 47.3, 18.5 with it (task average, σ = 0.3)

Distributed, Intelligent, and Scalable COmputing DISC Lab

The Mechanism, Measured, and Its Cost



HalfCheetah Medium at σ = 0.3, 6 training seeds (paper Fig. 2b, 2c). Lower is better ↓ in both panels.

Query deviation after one Hopfield update, keys fixed

0.52–0.55× of before

in every layer · gradient penalty: local Lipschitz ≈ 15% lower (1.54 vs 1.83)

Honest limits (§5.5, paper Table 7)

- Noise also moves the keys: in the full forward pass the deviation grows (ratio 1.26–1.41)
- Without the update ($N_{hop} = 0$): 45.7 mean, 18.0 CVaR-10 vs 47.3, 18.5 with it (task average, σ = 0.3)

Computational cost ↓ (paper Table 3)

Method	Train ms/step	Train mem. MiB	Action ms	Params K
DT	4.53	174	1.25	859
RDT	4.96	174	1.28	859
ARDT	5.01	174	1.25	1,135
RHDT (ours)	**13.91**	**401**	**1.91**	**728**

- **2.8× longer per training step** than RDT (extra backward pass for the gradient penalty)
- **1.5× longer per action**: about 520 actions per second
- 15% fewer parameters (728K vs 859K)

Distributed, Intelligent, and Scalable COmputing DISC Lab

Outline

- Background and Motivation
- RHDT: Robust Hopfield Decision Transformer
  - Hopfield Attention for Decision Making
  - Regularizers and Theory
- Key Contributions
- Evaluation
- Conclusion

Distributed, Intelligent, and Scalable COmputing DISC Lab

Conclusion

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.

Distributed, Intelligent, and Scalable COmputing DISC Lab

Conclusion

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.
- **Problem:** Softmax attention forms each representation in a single step, with no explicit mechanism to limit corrupted observations (sensor noise); robust variants (RDT, ARDT) change only the training.

Conclusion

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.
- **Problem:** Softmax attention forms each representation in a single step, with no explicit mechanism to limit corrupted observations (sensor noise); robust variants (RDT, ARDT) change only the training.
- **Solution (RHDT):** Replace softmax attention with causally masked, multi-head Hopfield attention; regularize training with a gradient penalty and a key-orthogonality penalty.

Distributed, Intelligent, and Scalable COmputing DISC Lab

Conclusion

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.
- **Problem:** Softmax attention forms each representation in a single step, with no explicit mechanism to limit corrupted observations (sensor noise); robust variants (RDT, ARDT) change only the training.
- **Solution (RHDT):** Replace softmax attention with causally masked, multi-head Hopfield attention; regularize training with a gradient penalty and a key-orthogonality penalty.
- **Key Contribution:** First Hopfield attention in Decision Transformers (to our knowledge); Theorems 1 and 2 motivate the two regularizers; the mechanism is measured directly.

Distributed, Intelligent, and Scalable COmputing DISC Lab

Conclusion

- **Background:** Decision Transformer (DT) casts offline RL as sequence modeling: actions are predicted with softmax attention over past returns-to-go, states, and actions.
- **Problem:** Softmax attention forms each representation in a single step, with no explicit mechanism to limit corrupted observations (sensor noise); robust variants (RDT, ARDT) change only the training.
- **Solution (RHDT):** Replace softmax attention with causally masked, multi-head Hopfield attention; regularize training with a gradient penalty and a key-orthogonality penalty.
- **Key Contribution:** First Hopfield attention in Decision Transformers (to our knowledge); Theorems 1 and 2 motivate the two regularizers; the mechanism is measured directly.
- **Key Findings:** At noise σ = 0.3 on six D4RL tasks: +14% mean score vs RDT (47.3 vs 41.4) and +62% CVaR-10 vs ARDT (18.5 vs 11.4); best mean score in 24/24 settings, best CVaR-10 in 22/24. Cost vs RDT: 2.8× time per training step, 1.5× per action.

CVaR-10 = mean normalized return of the worst 10% of episodes · σ = noise std as a fraction of each state dimension's dataset std
RDT = Robust Decision Transformer · ARDT = Adversarially Robust Decision Transformer · D4RL = offline RL benchmark

Distributed, Intelligent, and Scalable COmputing DISC Lab

NEURAL INFORMATION PROCESSING SYSTEMS **NeurIPS 2026**

Distributed, Intelligent, and Scalable COmputing DISC Lab

Robust Hopfield Decision Transformer

Andreia Podasca and Anup Das

Drexel University, Philadelphia

Email: *andreia.podasca@drexel.edu*

LinkedIn: *linkedin.com/in/andreia-podasca*

GitHub

AMBITION CAN'T WAIT