CORRUPTION-ROBUST OFFLINE REINFORCEMENT LEARNING VIA CAUSAL DYNAMICS ADHERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning (RL) via sequence modeling is inherently vulnerable to training data corruptions. We observe that while data can be corrupted, the causal dynamics of the environment, such as state transitions, can remain invariant. Motivated by this key observation, we propose Causal Dynamics Adherence (CDA), a mechanism that disentangles task-relevant causal information from corruptions by constraining representations to adhere to environmental causal dynamics. To mitigate the attention leakage to corrupted inputs induced by Softmax in Decision Transformer (DT), we introduce Adaptive Sparse Attention (ASA), a plug-and-play module that augments Sparsemax with layer-specific learnable temperatures to adaptively suppress such leakage. On the D4RL benchmark, CDA-DT, which integrates CDA and ASA into DT, achieves the highest scores on 47 out of 54 (87%) tasks. The code is provided in the supplementary material and will be publicly released. Training results are available via an anonymous link.[1]

1 INTRODUCTION

Offline Reinforcement Learning (RL) leverages static datasets to optimize policies without direct environmental interaction, offering an effective paradigm for real-world applications where online exploration is impractical (Prudencio et al., 2023). Recently, Decision Transformer (DT) (Chen et al., 2021) has emerged as a prevailing approach by reframing offline RL as a supervised sequence modeling problem. However, owing to the inherent sensitivity to data fidelity in such supervised paradigms, maintaining the robustness of offline RL against training data corruptions—a field known as corruption-robust offline RL—has emerged as an active research topic (Xu et al., 2025; Liu et al., 2025; Yang et al., 2024).

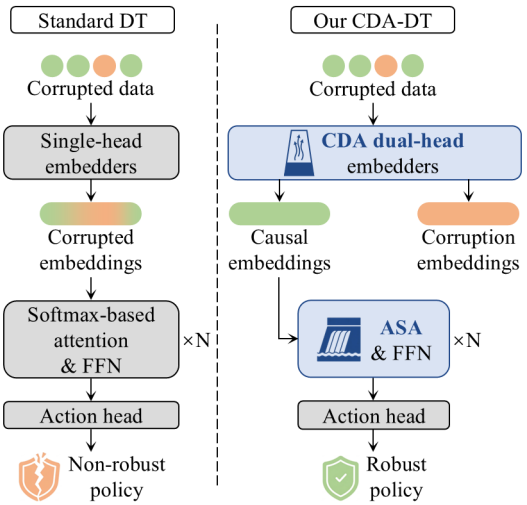

Figure 1: Comparison of DT (Chen et al., 2021) and our CDA-DT. CDA replaces DT's single-head embedders with dual-head embedders. ASA replaces DT's Softmax-based attention with adaptive sparse attention. "FFN": feedforward network.

A fundamental yet underexploited property of this field is that, while data values can be corrupted, the environmental causal dynamics (e.g., state transitions $(\mathbf{s}_t, \mathbf{a}_t) \rightarrow \mathbf{s}_{t+1}$ governed by physics simulation engines) often remain invariant. This property appears in various application domains: rigid body dynamics in robotics are preserved under sensor noise (Barrau & Bonnabel, 2017), and power flow equations in grid control hold under meter faults (Abur & Expósito, 2004). The invariant causal dynamics can therefore serve as a reliable criterion to disentangle task-relevant causal information from corruptions

[1] https://drive.google.com/drive/folders/15qnwhsYEVDVbgk00Kvpv1fKsmBK4JUPb?usp=sharing

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

in an unsupervised way: representations consistent with the dynamics capture causal features, while inconsistent representations absorb corrupted features.

Motivated by this insight, we propose Causal Dynamics Adherence (CDA), a mechanism that constrains representations to adhere to environmental causal dynamics, achieving unsupervised causal-corruption disentanglement. We use CDA to enhance vanilla DT, yielding a new model we call CDA-DT. As shown in Fig. 1, CDA replaces the standard single-head embedders in DT with dual-head embedders, mapping each input token to two embeddings: a causal embedding and a corruption embedding. We propagate only the causal embedding for action generation, thereby preventing corruptions from propagating into deeper networks.

To train the dual-head embedders, we design three synergistic loss functions:

(1) Causal dynamics loss $\mathcal{L}_{\text{caus}}$: It constrains the causal embedding to a dynamics-consistent causal manifold, providing a criterion for disentangling causal information from corruptions.

(2) Orthogonality loss $\mathcal{L}_{\text{orth}}$: To achieve thorough disentanglement, it minimizes the alignment between causal and corruption embeddings by encouraging their orthogonality.

(3) Reconstruction loss $\mathcal{L}_{\text{rec}}$: To prevent representation collapse during the unsupervised disentanglement, it preserves the input information by reconstructing the input token from causal and corruption embeddings.

Guided by the primary task objective (i.e., final action generation), $\mathcal{L}_{\text{caus}}$ refines the causal embedding into task-relevant features. Meanwhile, $\mathcal{L}_{\text{orth}}$ and $\mathcal{L}_{\text{rec}}$ work together to compel the corruption embedding to absorb perturbations. CDA functions as a causal information distiller, individually improving performance on 44 out of 54 tasks in the D4RL benchmark (Fu et al., 2020).

In vanilla DT, Softmax (Bridle, 1989)-based attention assigns non-zero weights to all tokens, thereby causing attention leakage into corrupted inputs. Sparsemax (Martins & Astudillo, 2016) offers a natural remedy by producing exactly zero weights. However, simply replacing Softmax with standard Sparsemax proves inadequate: as training proceeds, the attention leakage in Sparsemax becomes as severe as that in Softmax (see Fig. 4(a)). This degradation likely arises because standard Sparsemax lacks an explicit mechanism for adapting its sparsity level during training.

To overcome the limitation, we propose Adaptive Sparse Attention (ASA), a plug-and-play module that augments Sparsemax with layer-specific learnable sparse temperatures. We replace all Softmax-based attention layers with ASA, as shown in Fig. 1. To our knowledge, ASA is the *first* method that enables Sparsemax to adapt sparsity across layers and marks the *first* application of Sparsemax in corruption-robust offline RL. With ASA, deep layers tend to learn lower temperatures to aggressively filter out corrupted tokens, while shallow layers relax the sparsity constraint to preserve potentially useful information for downstream processing. Acting as an attention floodgate against data corruptions, ASA improves performance on 39 out of 54 tasks when used alone.

CDA-DT combines CDA's proactive corruption elimination with ASA's reactive suppression of residual corruption, providing complementary robustness at both the feature extraction and attention allocation stages. Evaluated on the D4RL benchmark (Fu et al., 2020), the proposed CDA-DT achieves the highest scores on 47 out of 54 (87%) tasks (5 of the remaining 7 are second-best), ranking first in cross-task averages for all six corruption types.

2 Related work

Offline RL and Decision Transformer (DT). Offline RL optimizes policies solely using static datasets, circumventing direct environmental interaction. Departing from classic value-based methods (Fujimoto et al., 2019; Kumar et al., 2020; Kostrikov et al., 2022), Decision Transformer (DT) (Chen et al., 2021) recasts offline RL as a sequence modeling problem, directly mapping target returns and historical trajectories to actions. Building upon DT, MGDT (Lee et al., 2022) uses fine-tuning to rapidly adapt to new tasks. EDT (Wu et al., 2023) enables trajectory stitching through adaptive history lengths. HDT (Correia & Alexandre, 2023) incorporates subgoal states to guide decision-making.

However, such supervised methods are inherently sensitive to data fidelity, making robustness against training data corruptions a critical challenge.

Corruption-robust offline RL. General robust offline RL can be broadly divided into two categories: defending against testing-time attacks (Yang et al., 2022; Panaganti et al., 2022; Yang & Xu, 2024) and mitigating the impact of training-time data corruptions (known as corruption-robust offline RL). Focusing on the latter, UWMSG (Ye et al., 2023) extends its defense against data corruptions to general function approximation settings. RIQL (Yang et al., 2024) enhances Implicit Q-learning (IQL) (Kostrikov et al., 2022) with Huber loss and quantile estimators to improve its robustness to diverse data corruptions.

Given the remarkable success of DT in offline RL (Prudencio et al., 2023), recent efforts have increasingly explored DT-based methods. DeFog (Hu et al., 2023) fortifies DT against incomplete trajectories with a two-stage training strategy based on frame masking. RDT (Xu et al., 2025) integrates Gaussian-weighted learning, embedding dropout, and iterative data correction into DT.

Despite recent progress, existing corruption-robust offline RL approaches have yet to systematically leverage the invariance of environmental causal dynamics.

3 PRELIMINARIES

RL and offline RL. The RL environment is typically modeled as a Markov Decision Process (MDP) defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ (Sutton et al., 1998). $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces. $\mathcal{P}$ represents the state transition function. r is the reward function. $\gamma \in [0, 1]$ is the discount factor. The standard objective in RL is to optimize a policy π that maximizes the expected return.

In offline RL, policy optimization is confined to a static dataset $\mathcal{D} = \{\tau^i\}_{i=1}^M$ containing M trajectories (Prudencio et al., 2023). Each trajectory is a sequence of transitions: $\tau^i = (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})_{t=1}^{L^i}$, where $\mathbf{s}_t$ and $\mathbf{a}_t$ are the state and action at time step t, and L^i is the trajectory length.

Decision Transformer. Decision Transformer (DT) (Chen et al., 2021) reframes offline RL as a sequence modeling problem. A trajectory is reorganized into an ordered sequence of returns, states, and actions: $\tau_L = (R_1, \mathbf{s}_1, \mathbf{a}_1, \ldots, R_L, \mathbf{s}_L, \mathbf{a}_L)$, where $R_t = \sum_{t'=t}^{L} \gamma^{t'-t} r_{t'}$. Each token in the trajectory is first mapped to a high-dimensional embedding through a linear layer. Then, DT employs a standard causal Transformer backbone (e.g., GPT-2 (Radford et al., 2019)) and an action head to predict actions from the embeddings. The model parameters are optimized with an action prediction loss $\mathcal{L}_{\text{act}}$ that measures the discrepancy between predicted and ground-truth actions.

Corruption-robust offline RL. Consistent with RDT (Xu et al., 2025), we apply two training data corruption schemes.

(1) Random corruptions. Random noise is sampled from a uniform distribution. For example, a state $\mathbf{s}$ may be corrupted as follows: $\mathbf{s}' = \mathbf{s} + \lambda \cdot \boldsymbol{\sigma}_{\mathbf{s}}$, where the noise scalar $\lambda \sim \text{Uniform}(-\alpha, \alpha)$ and $\boldsymbol{\sigma}_{\mathbf{s}}$ is the state standard deviation computed across the dataset $\mathcal{D}$.

(2) Adversarial corruptions. Adversarial attacks are performed using Projected Gradient Descent (Madry et al., 2018). Specifically, learnable perturbations are applied to minimize the Q-values of a value function pre-trained on clean data, thereby misleading the policy into suboptimal regions.

The same network architecture and learning objective are used for both schemes. Detailed corruption settings are provided in Appendix A.1.

Softmax and Sparsemax. In the standard attention mechanism (Vaswani et al., 2017), Softmax (Bridle, 1989) is used to map attention scores $\mathbf{z} \in \mathbb{R}^K$ (a real-valued vector) to attention weights $\mathbf{p} \in \Delta^{K-1}$ (a probability distribution over the standard $(K-1)$-simplex):

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}. \tag{1}$$

While Softmax can be derived as a special case of the p-entropy regularization framework with $p \to 1$, Sparsemax (Martins & Astudillo, 2016) corresponds to the case where $p = 2$ (Peters et al., 2019).

Sparsemax is formally defined as a Euclidean projection of $\mathbf{z}$ onto the probability simplex and can be computed exactly through a thresholding operation:

$$\text{sparsemax}(\mathbf{z}) = \underset{\mathbf{p}\in\Delta^{K-1}}{\arg\min} \|\mathbf{p}-\mathbf{z}\|_2^2 = \left[\max\left(0, z_i - \nu(\mathbf{z})\right)\right]_{i=1}^{K}. \tag{2}$$

Here, ν is a dynamic thresholding function that ensures the sum-to-one constraint of probability distributions.

In Transformer models, Sparsemax can assign exactly zero attention weights to lower-scoring tokens. This sparsity makes Sparsemax well suited to corruption-robust offline RL by enabling the model to block attention to corrupted tokens.

4 METHODS

We propose CDA-DT, a robust Decision Transformer (DT) (Chen et al., 2021) variant equipped with Causal Dynamics Adherence (CDA) and Adaptive Sparse Attention (ASA). As depicted in Fig. 1, CDA-DT replaces DT's single-head embedders with CDA dual-head embedders, while enhancing the standard attention layers with ASA. The pseudocode of CDA-DT is provided in Appendix A.3.

4.1 CAUSAL DYNAMICS ADHERENCE (CDA)

By leveraging the invariance of environmental causal dynamics, we propose CDA, a robust embedding strategy that achieves unsupervised causal-corruption disentanglement, guarding deeper layers against training data corruptions.

CDA employs dual-head embedders to map an input token $x \in \{\mathbf{s}, \mathbf{a}, R\}$ to a causal embedding $\mathbf{z}^x_{\text{caus}}$ and a corruption embedding $\mathbf{z}^x_{\text{corr}}$. Specifically, a shared linear layer with a GELU activation (Hendrycks & Gimpel, 2016) first extracts the intermediate feature $\mathbf{h}_x$, which is then projected to $\mathbf{z}^x_{\text{caus}}$ and $\mathbf{z}^x_{\text{corr}}$ by two parallel linear heads:

$$\mathbf{h}_x = \text{GELU}\left(\mathbf{W}^x_{\text{shared}} x + \mathbf{b}^x_{\text{shared}}\right), \quad \mathbf{z}^x_{\text{caus}} = \mathbf{W}^x_{\text{caus}}\mathbf{h}_x + \mathbf{b}^x_{\text{caus}}, \quad \mathbf{z}^x_{\text{corr}} = \mathbf{W}^x_{\text{corr}}\mathbf{h}_x + \mathbf{b}^x_{\text{corr}}. \tag{3}$$

Here, $\mathbf{W}^x_\star$ and $\mathbf{b}^x_\star$ ($\star \in \{\text{shared}, \text{caus}, \text{corr}\}$) are the learnable weights and biases of the linear layers, respectively. Only $\mathbf{z}^x_{\text{caus}}$ is propagated to generate final actions. During inference, $\mathbf{z}^x_{\text{corr}}$ is no longer computed to enhance computational efficiency.

To train the dual-head embedders, we design three synergistic loss functions: causal dynamics loss $\mathcal{L}_{\text{caus}}$, orthogonality loss $\mathcal{L}_{\text{orth}}$, and reconstruction loss $\mathcal{L}_{\text{rec}}$.

Causal dynamics loss $\mathcal{L}_{\text{caus}}$. To guide $\mathbf{z}^x_{\text{caus}}$ toward causal representations and filter out non-causal corruptions, we constrain $\mathbf{z}^x_{\text{caus}}$ to a dynamics-consistent causal manifold in the representation space by minimizing $\mathcal{L}_{\text{caus}}$. We formulate $\mathcal{L}_{\text{caus}}$ separately for states/actions and returns to respect their heterogeneous causal dynamics.

(1) Causal dynamics loss for states and actions ($\mathcal{L}^{\mathbf{s},\mathbf{a}}_{\text{caus}}$). The causal flow of state transitions can be formulated as:

$$(\mathbf{s}_t, \mathbf{a}_t) \to \mathbf{s}_{t+1}. \tag{4}$$

Based on Eq. (4), we constrain the state causal embedding $\mathbf{z}^{\mathbf{s}_t}_{\text{caus}}$ and the action causal embedding $\mathbf{z}^{\mathbf{a}_t}_{\text{caus}}$ to conform to the causal dynamics of state transitions:

$$\mathcal{L}^{\mathbf{s},\mathbf{a}}_{\text{caus}} = \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_{t+1})\sim\mathcal{D}}\left[\left\| f^{\mathbf{s}}_{\text{caus}}\left([\mathbf{z}^{\mathbf{s}_t}_{\text{caus}}; \mathbf{z}^{\mathbf{a}_t}_{\text{caus}}]\right) - \mathbf{z}^{\mathbf{s}_{t+1}}_{\text{caus}}\right\|_2^2\right]. \tag{5}$$

Here, $f^{\mathbf{s}}_{\text{caus}}$ represents a state embedding predictor that maps the concatenation of $\mathbf{z}^{\mathbf{s}_t}_{\text{caus}}$ and $\mathbf{z}^{\mathbf{a}_t}_{\text{caus}}$ to the estimated next-state causal embedding $\hat{\mathbf{z}}^{\mathbf{s}_{t+1}}_{\text{caus}}$. Further details of $f^{\mathbf{s}}_{\text{caus}}$ are provided in Appendix A.4.

(2) Causal dynamics loss for returns ($\mathcal{L}^{R}_{\text{caus}}$). The causal flow of return evolution (including reward generations) can be formulated as:

$$(R_t, r_t(\mathbf{s}_t, \mathbf{a}_t)) \to R_{t+1}. \tag{6}$$

Based on Eq. (6), we constrain the return causal embedding $\mathbf{z}^{R_t}_{\text{caus}}$ to follow the causal dynamics of return evolutions:

$$\mathcal{L}^{R}_{\text{caus}} = \mathbb{E}_{(R_t,\mathbf{s}_t,\mathbf{a}_t)\sim\mathcal{D}}\left[\left\| f^{R}_{\text{caus}}\left(\left[\mathbf{z}^{R_t}_{\text{caus}}; \mathbf{z}^{\mathbf{s}_t}_{\text{caus}}; \mathbf{z}^{\mathbf{a}_t}_{\text{caus}}\right]\right) - \mathbf{z}^{R_{t+1}}_{\text{caus}}\right\|_2^2\right]. \tag{7}$$

Here, f^R_{caus} represents a return embedding predictor that estimates $\hat{\mathbf{z}}^{R_{t+1}}_{\text{caus}}$.

In Eqs. (5) and (7), for clean tokens (e.g., actions unaffected by state corruptions), causal embeddings are generated by the original single-head embedders rather than the CDA embedders, and are detached from the computation graphs to serve as reference anchors. These anchors align the causal embeddings of corrupted tokens with the environmental causal dynamics, and prevent representation collapse during the unsupervised disentanglement.

Orthogonality loss $\mathcal{L}_{\text{orth}}$. To achieve thorough disentanglement, we augment $\mathcal{L}_{\text{caus}}$ with an orthogonality loss $\mathcal{L}_{\text{orth}}$. It encourages orthogonality between $\mathbf{z}^x_{\text{caus}}$ and $\mathbf{z}^x_{\text{corr}}$ by minimizing their squared cosine similarity:

$$\mathcal{L}_{\text{orth}} = \mathbb{E}_{x\sim\mathcal{D}}\left[\left(\frac{\left(\mathbf{z}^x_{\text{caus}}\right)^\top \mathbf{z}^x_{\text{corr}}}{\|\mathbf{z}^x_{\text{caus}}\|_2\|\mathbf{z}^x_{\text{corr}}\|_2}\right)^2\right]. \tag{8}$$

Reconstruction loss $\mathcal{L}_{\text{rec}}$. To prevent representation collapse during the unsupervised disentanglement, we preserve the input information by reconstructing x from the sum of $\mathbf{z}^x_{\text{caus}}$ and $\mathbf{z}^x_{\text{corr}}$:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x\sim\mathcal{D}}\left[\left\|f^x_{\text{rec}}\left(\mathbf{z}^x_{\text{caus}} + \mathbf{z}^x_{\text{corr}}\right) - x\right\|_2^2\right]. \tag{9}$$

Here, f^x_{rec} represents a reconstruction head that maps the sum of $\mathbf{z}^x_{\text{caus}}$ and $\mathbf{z}^x_{\text{corr}}$ to a reconstructed input token $\hat{x}$.

Combining the above components, we formulate the comprehensive CDA loss $\mathcal{L}_{\text{CDA}}$ as a weighted sum of $\mathcal{L}_{\text{caus}}$, $\mathcal{L}_{\text{orth}}$, and $\mathcal{L}_{\text{rec}}$:

$$\mathcal{L}_{\text{CDA}} = \lambda_{\text{caus}}\mathcal{L}_{\text{caus}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}. \tag{10}$$

Here, λ_{caus}, λ_{orth}, and λ_{rec} are CDA loss coefficients. Their values are held fixed across all environments (detailed in Appendix A.2).

CDA is applied to states, actions, or returns only when the corresponding data type is corrupted.

The total loss $\mathcal{L}_{\text{tot}}$ integrates $\mathcal{L}_{\text{CDA}}$ with the primary action prediction loss $\mathcal{L}_{\text{act}}$:

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{\tau_t\sim\mathcal{D}}\left[-w_{\text{Gauss}}\log\pi_{\boldsymbol{\theta}}\left(\mathbf{a}_t \mid \tau_t \setminus \mathbf{a}_t\right)\right], \tag{11}$$

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{CDA}}. \tag{12}$$

Here, π_θ denotes a policy (parametrized by network weights $\boldsymbol{\theta}$) that generates action probability distributions. In the presence of corrupted action labels, we adopt the Gaussian weight w_{Gauss} from RDT (Xu et al., 2025) to exponentially down-weight samples with high prediction errors, thereby mitigating the detrimental effects of corrupted labels.

4.2 Adaptive Sparse Attention (ASA)

To mitigate the harmful attention leakage toward corrupted data, we propose ASA, a plug-and-play module that redesigns the attention routing process.

We augment standard Sparsemax with layer-specific learnable sparse temperatures T_d (where d indexes the Transformer layer) and replace all Softmax (Bridle, 1989) operations across Transformer layers with this adaptive variant. ASA can assign exactly zero attention weights to corrupted tokens and adapt its sparsity level across layers. For the d-th Transformer layer, ASA is formulated as:

$$\text{ASA}_d\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right) = \text{sparsemax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{T_d\sqrt{d_k}}\right)\mathbf{V}. \tag{13}$$

Here, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value matrices, respectively. d_k is the dimension of each key vector. T_d is the learnable sparse temperature of the d-th Transformer layer, controlling the sparsity of the attention distribution in that layer. The initial value of T_d is the same for all layers and tasks. To balance adaptability with training stability, we update T_d every five epochs.

Table 1: Main experiment results. Results are reported as mean ± sample standard deviation (over four random seeds). **Bold blue**: the highest scores (ties broken by lower standard deviation). "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption. CDA-DT achieves the highest scores on 47 out of 54 (87%) tasks, surpassing all baselines across all six corruption types.

Task	Corrupt.		BC	RBC	UWMSG	RIQL	DT	DeFog	RDT	CDA-DT (ours)
door (1%)	adv	act	8.60±5.23	26.36±7.68	-0.18±0.02	89.29±8.70	102.54±0.91	**104.02±0.44**	103.33±0.37	102.95±1.44
		sta	85.69±11.98	87.49±12.27	-0.20±0.00	91.36±7.88	105.46±0.17	104.78±0.58	105.12±0.31	**105.67±0.37**
		rew	94.35±9.50	101.60±1.26	-0.20±0.01	61.50±20.77	105.03±0.25	104.99±0.16	105.02±0.58	**105.13±0.22**
	rnd	act	78.80±14.77	85.06±10.78	-0.19±0.04	75.99±9.53	105.08±0.55	105.25±0.25	105.04±0.14	**105.36±0.11**
		sta	92.43±8.65	93.32±8.84	0.07±0.38	89.10±8.74	**105.63±0.20**	104.42±1.17	105.14±0.35	105.50±0.43
		rew	94.35±9.50	101.60±1.26	-0.24±0.10	84.80±5.65	105.19±0.38	104.92±0.21	104.85±0.34	**105.24±0.39**
hammer (1%)	adv	act	65.13±14.28	66.87±21.09	1.28±1.20	99.50±23.13	89.95±11.09	86.00±9.95	115.77±10.41	**122.83±1.82**
		sta	114.14±10.04	105.90±21.58	0.23±0.04	80.58±16.84	124.54±1.88	125.67±0.87	126.62±0.23	**127.02±0.32**
		rew	111.80±5.84	119.15±7.57	0.26±0.02	80.81±19.93	125.58±0.09	125.31±0.73	127.44±0.66	**127.73±0.08**
	rnd	act	101.68±13.33	106.31±16.10	0.49±0.50	86.01±17.85	122.85±6.22	125.47±0.68	123.23±7.32	**127.77±0.87**
		sta	107.84±12.47	107.63±15.75	0.13±0.10	88.83±14.10	125.60±1.12	125.51±0.67	122.75±5.75	**126.91±0.25**
		rew	111.80±5.84	119.15±7.57	0.39±0.28	94.50±11.86	124.29±2.36	125.78±0.89	126.88±0.31	**127.32±0.29**
relocate (1%)	adv	act	14.46±12.24	21.12±14.76	-0.30±0.03	4.84±2.06	26.11±7.72	32.71±7.28	67.94±4.41	**69.88±6.68**
		sta	32.74±4.65	33.76±10.53	-0.30±0.03	25.29±12.80	89.57±10.43	44.87±10.38	85.27±7.45	**93.09±3.85**
		rew	56.15±8.67	58.72±12.40	-0.30±0.03	19.51±4.47	91.02±9.02	71.01±6.41	84.70±5.43	**98.43±2.58**
	rnd	act	41.05±22.02	49.39±10.37	-0.30±0.03	23.08±10.24	80.38±2.35	68.02±9.44	85.38±8.12	**92.38±4.19**
		sta	26.40±8.80	17.36±1.79	-0.30±0.03	23.20±4.59	83.83±4.82	31.42±4.95	75.36±10.06	**84.38±7.45**
		rew	56.15±8.67	58.72±12.40	-0.30±0.03	22.02±13.74	91.82±10.88	72.34±9.54	94.64±8.37	**95.51±6.24**
halfcheetah (10%)	adv	act	0.49±0.17	0.81±0.38	7.63±2.48	1.40±0.62	20.33±1.20	13.89±2.34	**23.40±0.96**	22.94±1.42
		sta	3.68±0.40	3.45±0.75	10.33±6.54	5.74±1.61	11.56±1.45	11.85±2.41	10.87±0.89	**13.86±0.75**
		rew	6.26±2.03	5.89±1.49	2.18±0.39	13.74±1.33	24.78±3.76	18.34±2.56	27.61±3.84	**34.95±1.19**
	rnd	act	1.74±0.39	2.04±0.63	11.26±3.95	3.78±0.36	18.48±3.40	16.55±2.13	20.49±4.27	**26.73±0.98**
		sta	3.36±0.83	3.38±0.17	4.71±1.09	6.29±1.02	11.56±1.42	11.15±2.86	12.93±1.19	**18.36±2.85**
		rew	6.26±2.03	5.89±1.49	3.10±0.58	11.30±1.54	23.78±3.46	19.75±1.15	25.63±4.39	**35.40±0.77**
hopper (10%)	adv	act	19.54±5.13	17.25±2.88	45.17±14.47	32.23±11.27	41.82±8.04	45.67±4.18	57.70±7.28	**58.49±4.28**
		sta	23.72±2.86	26.74±4.98	25.32±9.29	20.00±7.07	50.82±5.10	34.32±8.27	47.42±3.08	**60.45±3.06**
		rew	29.79±7.59	30.01±6.22	47.08±16.79	45.93±5.42	51.49±4.36	49.80±16.17	55.08±2.54	**57.80±5.15**
	rnd	act	25.96±5.46	23.47±1.30	50.03±27.28	32.01±4.19	46.93±3.28	46.24±4.33	52.45±4.45	**53.03±4.46**
		sta	20.30±3.96	18.54±5.93	21.65±8.58	30.17±5.70	44.46±2.31	36.06±2.35	44.70±3.70	**56.09±4.11**
		rew	29.79±7.59	30.01±6.22	27.66±4.23	40.85±2.32	51.75±2.52	53.07±13.29	50.42±8.09	**58.69±4.26**
walker2d (10%)	adv	act	5.69±1.25	4.59±1.53	6.22±0.41	6.48±1.94	25.95±3.14	16.64±5.79	27.93±6.25	**32.07±1.91**
		sta	16.44±4.12	18.31±5.17	5.51±2.30	18.02±2.91	38.41±2.96	20.01±2.66	34.07±9.55	**44.04±9.46**
		rew	18.02±2.11	20.24±7.52	24.58±8.09	15.18±4.18	47.71±1.97	26.13±8.01	49.49±3.01	**55.29±7.36**
	rnd	act	9.65±0.82	12.67±2.80	12.44±3.63	17.33±5.06	36.46±2.27	27.49±3.88	48.83±3.96	**50.52±4.97**
		sta	11.22±1.96	13.51±4.93	-0.41±0.35	16.23±2.61	30.96±1.10	19.55±2.92	31.14±3.94	**34.66±4.97**
		rew	18.02±2.11	20.24±7.52	11.50±4.69	12.83±3.64	53.54±7.40	27.14±4.14	49.46±7.43	**57.25±8.48**
kitchen-complete	adv	act	11.88±7.18	10.00±2.04	0.00±0.00	24.38±5.15	25.62±5.15	25.62±1.25	26.88±1.25	**30.00±4.56**
		sta	48.12±12.14	46.25±6.61	0.00±0.00	46.25±7.50	85.00±5.00	64.38±3.15	76.25±4.33	**86.25±6.61**
		rew	36.25±21.26	50.62±6.25	0.00±0.00	60.62±7.47	**95.00±0.00**	65.62±6.88	81.88±3.75	84.38±5.54
	rnd	act	10.00±8.90	10.00±6.12	0.00±0.00	26.88±5.54	57.50±8.42	48.75±4.79	56.88±1.25	**58.12±2.39**
		sta	31.88±5.54	31.25±15.88	0.00±0.00	48.12±6.57	68.75±4.33	68.12±16.38	65.62±4.27	**68.75±3.23**
		rew	36.25±21.26	50.62±6.25	0.00±0.00	65.62±5.54	**93.75±4.33**	66.25±1.44	81.88±8.75	83.12±4.27
kitchen-mixed	adv	act	16.25±8.29	46.25±4.33	0.00±0.00	7.50±4.08	5.00±5.00	3.75±5.95	49.38±3.75	**55.00±6.12**
		sta	48.12±5.54	51.88±7.18	0.00±0.00	41.25±7.77	52.50±4.56	23.75±21.75	56.25±9.46	**56.25±4.79**
		rew	51.88±2.39	57.50±5.40	0.00±0.00	48.75±4.79	58.75±4.79	23.75±13.62	55.62±4.73	**59.38±6.88**
	rnd	act	54.38±3.15	51.88±3.15	0.00±0.00	48.75±6.61	51.25±4.79	31.88±4.27	53.75±6.61	**56.88±5.54**
		sta	34.38±4.73	37.50±2.04	0.00±0.00	24.38±17.49	48.75±1.44	30.00±14.86	50.62±1.25	**53.75±1.44**
		rew	51.88±2.39	57.50±5.40	0.00±0.00	55.00±10.61	53.12±3.15	26.88±16.50	**60.62±2.39**	56.25±9.46
kitchen-partial	adv	act	16.88±8.00	19.38±6.25	6.25±12.50	0.00±0.00	9.38±3.15	0.00±0.00	34.38±9.66	**43.12±7.47**
		sta	36.25±1.44	41.88±4.73	0.00±0.00	27.50±6.45	55.00±6.12	25.00±2.04	**67.50±7.91**	66.88±5.91
		rew	45.00±2.04	49.38±3.75	0.00±0.00	46.88±9.44	56.88±5.91	22.50±3.54	59.38±4.73	**63.75±4.79**
	rnd	act	47.50±2.89	46.88±2.39	0.00±0.00	51.88±6.88	56.25±11.27	25.00±14.29	**60.00±6.12**	**60.00±6.12**
		sta	31.25±1.44	33.12±3.75	0.00±0.00	12.50±8.90	60.62±5.15	20.62±8.75	58.75±9.24	**63.12±5.54**
		rew	45.00±2.04	49.38±3.75	0.00±0.00	63.12±13.13	56.88±2.39	21.25±7.50	68.12±4.27	**68.75±1.44**
Cross-task average	adv	act	17.66	23.63	7.34	29.51	38.52	36.48	56.30	**59.70**
		sta	45.43	46.18	4.54	39.55	68.10	50.51	67.71	**72.61**
		rew	49.94	54.79	8.18	43.66	72.92	56.38	71.80	**76.32**
	rnd	act	41.20	43.08	8.19	40.63	63.91	54.96	67.34	**70.09**
		sta	39.90	39.51	2.87	37.65	64.46	49.65	63.00	**67.95**
		rew	49.94	54.79	4.68	50.00	72.68	57.49	73.61	**76.39**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

Experiments are conducted using PyTorch on a Linux platform. The hardware configuration consists of an NVIDIA® GeForce RTX™ 4090 D GPU (24 GB VRAM) and an AMD® Ryzen™ 9 7950X CPU.

The task and data corruption settings are the same as those in RDT (Xu et al., 2025). We evaluate the proposed CDA-DT and baseline methods on the widely used D4RL offline RL dataset (Fu et al., 2020). Random and adversarial training data corruptions are applied to tasks from the Adroit (Rajeswaran et al., 2018), Kitchen (Gupta et al., 2020), and MuJoCo (Duan et al., 2016) environments. We apply one corruption scheme to one data type (state, action, or reward) in each task. Following RDT, we subsample 1% and 10% of the data for Adroit and MuJoCo tasks, respectively.

The training and evaluation hyperparameters for CDA-DT and baselines follow those in RDT (where applicable). We adopt the normalized score defined in the D4RL benchmark as the primary evaluation metric. A score of 0 corresponds to a random policy and 100 represents an expert policy. Following RDT, all experiments are conducted across four random seeds (held fixed across all tasks and all algorithms for a fair comparison). More training and evaluation settings are presented in Appendix A.4.

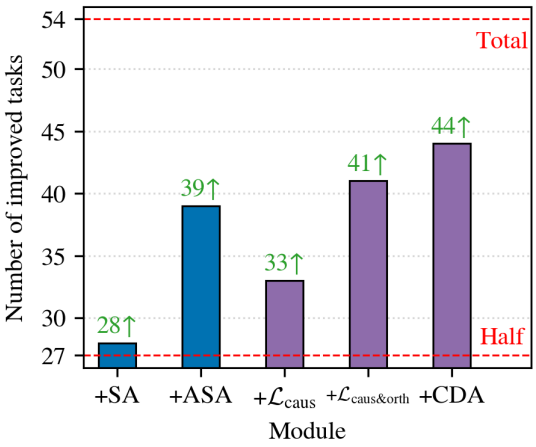

Figure 2: Per-module task improvements over our base model (CDA-DT without either CDA or ASA). The bars illustrate the number of tasks improved by each module. Results are averaged over four random seeds. "+SA": base model with standard sparse attention. "+ASA": base model with our adaptive sparse attention. "+$\mathcal{L}_{\text{caus}}$": base model with causal dynamics loss. "+$\mathcal{L}_{\text{caus\&orth}}$": base model with causal dynamics loss and orthogonality loss. "+CDA": base model with full CDA. Detailed results are presented in Appendix A.6.

5.2 Main Experiments

We compare CDA-DT with diverse corruption-robust offline RL methods and their non-robust base algorithms: Behavior Cloning (BC) (Bain & Sammut, 1995) and RBC (Sasaki & Yamashina, 2020), value-based methods including UWMSG (Ye et al., 2023) and RIQL (Yang et al., 2024), as well as sequence modeling methods including DT (Chen et al., 2021), DeFog (Hu et al., 2023), and RDT (Xu et al., 2025).

As presented in Table 1, CDA-DT achieves the highest scores on 47 out of 54 (87%) tasks across all environments (Adroit (Rajeswaran et al., 2018), Kitchen (Gupta et al., 2020), and MuJoCo (Duan et al., 2016)). CDA-DT also achieves the highest cross-task average for each of the six corruption types. For tasks on which CDA-DT achieves mean scores comparable to those of the best-performing baseline (e.g., the "kitchen-mixed-adv-sta" task), CDA-DT exhibits lower sample standard deviations, indicating greater cross-seed stability. An additional comparison with CQL (Kumar et al., 2020) is provided in Appendix A.12. Moreover, CDA-DT adds only a small number of parameters at inference time compared with DT, as detailed in Appendix A.5.

5.3 Ablation Study

All experimental setups in the ablation study are identical to those in the main experiments. Results are averaged over all 54 tasks and four random seeds.

The effect of CDA. To isolate the contribution of CDA, we incrementally add its three loss functions ($\mathcal{L}_{\text{caus}}$, $\mathcal{L}_{\text{orth}}$, and $\mathcal{L}_{\text{rec}}$) to our base model (CDA-DT without either CDA or ASA). As shown by the three purple bars on the far right of Fig. 2, each successive addition of a loss function brings a further performance gain.

Applying $\mathcal{L}_{\text{caus}}$ alone improves performance on 33 out of 54 tasks relative to the base model, suggesting that the invariant environmental causal dynamics can provide an effective criterion for the unsupervised causal-corruption disentanglement.

Augmenting $\mathcal{L}_{\text{caus}}$ with $\mathcal{L}_{\text{orth}}$ substantially raises the number of improved tasks from 33 to 41. The orthogonality loss makes the disentanglement more thorough and suppresses the leakage of residual corruptions into causal embeddings.

Incorporating $\mathcal{L}_{\text{rec}}$ to form the full CDA further increases the number of improved tasks from 41 to 44. This reconstruction objective prevents representation collapse by preserving input information during the unsupervised disentanglement. Moreover, applying full CDA to the base model reduces the average sample standard deviation from 4.53 to 4.28. CDA is robust to the choice of its three loss coefficients, as detailed in Appendix A.9.

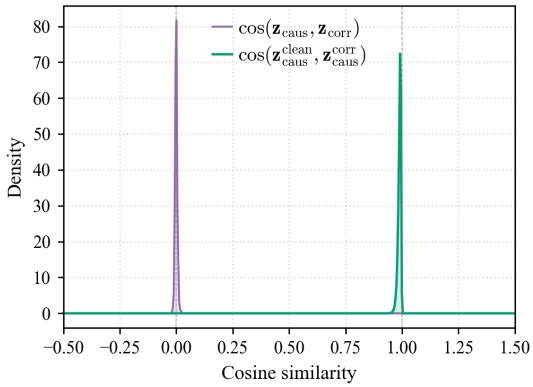

Figure 3: Disentanglement quality of CDA-DT on the "kitchen-complete-rnd-sta" task (seed 0). The purple "$\cos(\mathbf{z}_{\text{caus}}, \mathbf{z}_{\text{corr}})$" curve measures causal-corruption orthogonality by the cosine similarity between causal and corruption embeddings of the same state, while the green "$\cos(\mathbf{z}^{\text{clean}}_{\text{caus}}, \mathbf{z}^{\text{corr}}_{\text{caus}})$" curve measures causal invariance by the cosine similarity between causal embeddings of clean states and those of corrupted counterparts.

We further visualize CDA-DT's disentanglement quality on the challenging "kitchen-complete-rnd-sta" task whose states are high-dimensional, as shown in Fig. 3 (each curve is aggregated over about 1,500 samples). The purple "$\cos(\mathbf{z}_{\text{caus}}, \mathbf{z}_{\text{corr}})$" curve measures the cosine similarity between causal and corruption embeddings of the same state. Its concentration near 0 indicates that the two embeddings are nearly orthogonal. The green "$\cos(\mathbf{z}^{\text{clean}}_{\text{caus}}, \mathbf{z}^{\text{corr}}_{\text{caus}})$" curve measures the cosine similarity between causal embeddings of clean states and those of corrupted counterparts. Its concentration near 1 shows that clean and corrupted states yield highly similar causal embeddings. These results provide direct evidence that CDA effectively disentangles causal and corruption embeddings. A complementary t-SNE (van der Maaten & Hinton, 2008) visualization of the two embeddings under clean and corrupted inputs is provided in Appendix A.7.

The effect of ASA. As shown by the two blue bars on the far left of Fig. 2, standard Sparsemax improves performance on 28 of 54 tasks relative to the base model, while ASA raises this number to 39. The performance gap between SA and ASA confirms that the sparse temperature T_d is essential.

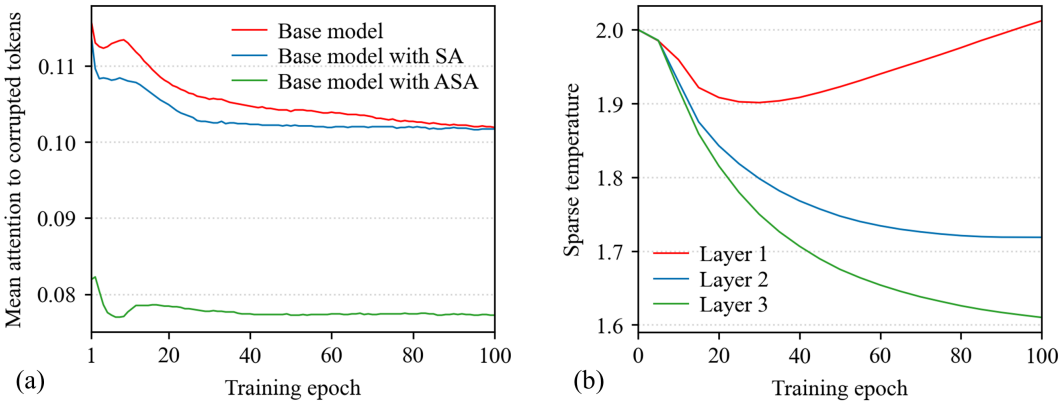

Figure 4: Effect and trend of the layer-specific learnable sparse temperatures T_d. Results are aggregated over all 54 tasks and four random seeds. **(a)** Attention to corrupted tokens throughout training. The curves show the mean attention weight that non-corrupted query tokens assign to corrupted key tokens. **(b)** The sparse temperatures of CDA-DT throughout training. The curves show the mean sparse temperature of each Transformer layer after every five-epoch update. "Layer 1": the shallowest layer (closest to the input). Detailed results of (a) and (b) are presented in Appendix A.8.

Table 2: Results under both training corruptions and testing attacks. Results are averaged over four random seeds. **Bold blue**: the highest scores (ties broken by lower standard deviation). "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption. CDA-DT consistently achieves the best performance across all six perturbation types. Detailed results are provided in Appendix A.10.

Perturb.		BC	RBC	UWMSG	RIQL	DT	DeFog	RDT	CDA-DT (ours)
adv	act	12.36	15.93	5.42	23.13	30.91	5.93	46.30	**48.15**
	sta	19.80	20.99	3.17	30.09	53.67	41.27	60.62	**63.75**
	rew	49.94	54.79	8.18	43.66	72.39	46.23	72.24	**76.43**
rnd	act	23.94	26.03	4.95	26.87	41.97	6.19	48.81	**52.28**
	sta	22.71	23.46	2.64	29.36	53.19	39.35	59.21	**61.82**
	rew	49.94	54.79	4.68	49.96	70.22	46.71	70.35	**71.64**

To understand how the sparse temperatures optimize attention routing, in Fig. 4(a), we visualize the mean attention weights (averaged over each training epoch and across all Transformer layers) from non-corrupted queries to corrupted keys throughout training. The initial advantage of SA diminishes as training proceeds: the blue "+SA" curve and the red "Base model" curve converge to 0.102 at the end of training.

ASA addresses this limitation through its sparse temperatures. As shown in Fig. 4(a), the green "+ASA" curve maintains lower attention to corrupted tokens than the blue "+SA" curve in the first training epoch, indicating that the sparse temperatures rapidly adjust the sparsity level for each task. This adaptation provides a good initialization for training. By the end of training, the "+ASA" curve converges to approximately 0.077, showing a decrease of 24.51% compared with the other two curves. Figure 4(a) confirms that the proposed layer-specific learnable sparse temperatures transform Sparsemax from a one-size-fits-all sparsification operator into an adaptive attention gate. Moreover, applying ASA to the base model reduces the average sample standard deviation from 4.53 to 3.79. ASA is insensitive to the initialization of its sparse temperatures, as detailed in Appendix A.9.

We further plot the evolution of the sparse temperatures in each Transformer layer throughout training (before training and after every five-epoch update) in Fig. 4(b). All sparse temperatures are initialized to 2.0. As training proceeds, the sparse temperatures of deep layers decrease (to 1.72 and 1.61 in "Layer 2" and "Layer 3", respectively), while the shallow layer rebounds (to 2.01 in "Layer 1") after an initial decrease. This layer-wise divergence reveals a hierarchical sparsity pattern: deeper layers, being nearest to the action head, converge to lower sparse temperatures that aggressively filter out corrupted inputs, whereas the shallow layer maintains higher temperatures to retain potentially useful information for downstream layers.

5.4 Robustness under both training corruptions and testing attacks

Following RDT (Xu et al., 2025), we further evaluate CDA-DT under a more challenging setting that combines training corruptions with testing attacks. Each testing attack applies the same perturbation scheme (random or adversarial) to the same data type (state, action, or reward) as its training corruption. However, testing attacks differ from training corruptions in terms of sampling distributions and adversarial procedures: random attacks are sampled from Gaussian rather than uniform distributions, while adversarial attacks are generated via zero-order policy-deviation search (Yang et al., 2022; Zhang et al., 2020) instead of critic-based Q-minimization (Madry et al., 2018). As shown in Table 2, CDA-DT consistently achieves the highest cross-task average scores across all six perturbation types.

6 Conclusion

We propose CDA-DT, a corruption-robust offline RL method that exploits the invariance of environmental causal dynamics. Causal Dynamics Adherence (CDA) constrains representations to adhere to causal dynamics through three synergistic loss functions, achieving unsupervised causal-corruption disentanglement. Adaptive Sparse Attention (ASA) introduces layer-specific learnable sparse temperatures, adaptively suppressing attention leakage to corrupted inputs. Experiments on D4RL show that CDA-DT achieves the highest scores on 47 out of 54 (87%) tasks.

STATEMENTS

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation and interpreting results. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting dataset, or supporting qualitative and thematic data analysis. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, drafting parts of a research paper, summarizing or analysing existing literature, sourcing/searching for information, editing a research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. LLM-generated result analysis was verified for correctness. We reviewed AI-assisted draft text for accuracy and readability. We manually reviewed the references suggested by LLMs to verify their accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies corruption-robust offline reinforcement learning using publicly available D4RL datasets and simulated Adroit, Kitchen, and MuJoCo environments. Our experiments do not involve recruiting human participants, collecting personal data, or deploying policies on physical systems. Random and adversarial perturbations are applied only within these benchmark experiments to evaluate robustness. Although CDA-DT demonstrates robustness to data corruption in the evaluated simulation environments, its application in safety-critical real-world systems would require further validation under deployment-specific conditions, appropriate safeguards, and human oversight. This research is intended to support beneficial applications of reliable decision-making, and we discourage any use that could harm individuals or society.

REPRODUCIBILITY STATEMENT

Our experiments use the publicly available D4RL benchmark. Section 5.1 describes the benchmark tasks, data subsampling, hardware, and evaluation metric. Section 4 describes CDA-DT in detail, and Appendix A.3 summarizes the training procedure. Appendix A.1 specifies the data corruption schemes, Appendix A.2 gives the CDA loss coefficients, and Appendix A.4 details the network architecture, training hyperparameters, and evaluation protocol. Our implementation is provided in the supplementary material and will be publicly released after the review process. Training results are available at this anonymous link: `https://drive.google.com/drive/folders/15qnwhsYEVDVbgk00Kvpv1fKsmBK4JUPb?usp=sharing`.

REFERENCES

Ali Abur and Antonio Gómez Expósito. *Power System State Estimation: Theory and Implementation*. Marcel Dekker, 2004. ISBN 0-8247-5570-7.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pp. 103–129, 1995.

Axel Barrau and Silvère Bonnabel. The invariant extended Kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, 2017. doi: 10.1109/TAC.2016.2594085.

John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

André Correia and Luís A Alexandre. Hierarchical decision transformer. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1661–1666. IEEE, 2023.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

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.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pp. 1025–1037. PMLR, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Kaizhe Hu, Ray Chen Zheng, Yang Gao, and Huazhe Xu. Decision transformer under random frame dropping. In *The Eleventh International Conference on Learning Representations*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35:27921–27936, 2022.

Zeyuan Liu, Zhihe Yang, Jiawei Xu, Rui Yang, Jiafei Lyu, Baoxiang Wang, Yunjian Xu, and Xiu Li. Adg: Ambient diffusion-guided dataset recovery for corruption-robust offline reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pp. 1614–1623. PMLR, 2016.

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust reinforcement learning using offline data. *Advances in neural information processing systems*, 35:32211–32224, 2022.

Ben Peters, Vlad Niculae, and André FT Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1504–1519, 2019.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *Robotics: Science and Systems XIV*, 2018.

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.

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

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yueh-Hua Wu, Xiaolong Wang, and Masashi Hamaya. Elastic decision transformer. *Advances in neural information processing systems*, 36:18532–18550, 2023.

Jiawei Xu, Rui Yang, Shuang Qiu, Feng Luo, Meng Fang, Baoxiang Wang, and Lei Han. Tackling data corruption in offline reinforcement learning via sequence modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. Rorl: Robust offline reinforcement learning via conservative smoothing. *Advances in neural information processing systems*, 35:23851–23866, 2022.

Rui Yang, Han Zhong, Jiawei Xu, Amy Zhang, Chongjie Zhang, Lei Han, and Tong Zhang. Towards robust offline reinforcement learning under diverse data corruption. In *International Conference on Learning Representations*, volume 2024, pp. 15512–15543, 2024.

Zhihe Yang and Yunjian Xu. Dmbp: Diffusion model-based predictor for robust offline reinforcement learning against state observation perturbations. In *International Conference on Learning Representations*, volume 2024, pp. 38549–38578, 2024.

Chenlu Ye, Rui Yang, Quanquan Gu, and Tong Zhang. Corruption-robust offline reinforcement learning with general function approximation. *Advances in Neural Information Processing Systems*, 36:36208–36221, 2023.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020.

A Appendix

A.1 Data corruption schemes

All corruption settings are consistent with those in RDT. The corruption rate is set to 30%. We reuse the corrupted datasets across all methods for a fair comparison.

Random corruptions. For state and action corruption, we set the noise magnitude α to 1.0. For reward corruption, each selected reward is replaced by $r'_t \sim \mathrm{Uniform}(-30, 30)$. Returns are then recomputed from the corrupted reward sequences.

Adversarial corruptions. For each selected transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$, we obtain state and action perturbations using the critic network of an IQL model pre-trained on clean data as a Q-function:

$$\boldsymbol{\delta}_t^{\mathbf{s}} \in \underset{|\delta_{t,j}^{\mathbf{s}}| \le \sigma_{\mathbf{s},j}}{\arg\min}\ Q(\mathbf{s}_t + \boldsymbol{\delta}_t^{\mathbf{s}}, \mathbf{a}_t)\,, \tag{14}$$

$$\boldsymbol{\delta}_t^{\mathbf{a}} \in \underset{|\delta_{t,j}^{\mathbf{a}}| \le \sigma_{\mathbf{a},j}}{\arg\min}\ Q(\mathbf{s}_t, \mathbf{a}_t + \boldsymbol{\delta}_t^{\mathbf{a}})\,. \tag{15}$$

The corrupted inputs are $\mathbf{s}'_t = \mathbf{s}_t + \boldsymbol{\delta}^{\mathbf{s}}_t$ and $\mathbf{a}'_t = \mathbf{a}_t + \boldsymbol{\delta}^{\mathbf{a}}_t$. Each perturbation is initialized uniformly within its feasible set and optimized for 100 steps using Adam with a learning rate of 0.01. For adversarial reward corruption, each selected reward is negated, i.e., $r'_t = -r_t$, after which the returns are recomputed.

A.2 CDA LOSS COEFFICIENTS

The coefficients of the CDA objectives are fixed across all environments and both random and adversarial corruption schemes. The causal dynamics loss weight λ_{caus} is set to 2.0 under action corruptions and set to 0.5 under state and reward corruptions. The larger weight for actions emphasizes the consistency with the causal dynamics of action representations, as actions directly mediate state transitions. The orthogonality loss weight λ_{orth} is set to 0.1. The reconstruction loss weight λ_{rec} is set to 0.5 under state corruptions and set to 0.1 under action and reward corruptions. The larger weight for states accounts for the greater reconstruction difficulty of high-dimensional state vectors.

Algorithm 1 CDA-DT training

Require: Corrupted dataset $\mathcal{D}$, epochs N, CDA loss coefficients λ_{caus}, λ_{orth}, λ_{rec}
Ensure: Model parameters $\boldsymbol{\theta}$, layer-wise sparse temperatures $T_d, d = 1, \ldots, D$

Initialize $\boldsymbol{\theta}$ and T_d
for epoch $= 1$ **to** N **do**
  for each mini-batch $\tau \sim \mathcal{D}$ **do**
    // CDA: dual-head embedding
    for each token $x \in \{\mathbf{s}, \mathbf{a}, R\}$ in τ **do**
      if the task corrupts the data type corresponding to x **then**
        $\mathbf{z}^x_{\text{caus}} \leftarrow \texttt{CausalHead}(\texttt{SharedEnc}(x))$
        $\mathbf{z}^x_{\text{corr}} \leftarrow \texttt{CorrHead}(\texttt{SharedEnc}(x))$ ▷ only during training
      else
        $\mathbf{z}^x_{\text{caus}} \leftarrow \texttt{SingleHeadEmbed}(x)$
      end if
    end for
    // ASA: adaptive sparse attention
    Feed $\mathbf{z}^x_{\text{caus}}$ through D Transformer layers with:
      $\text{ASA}_d(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{sparsemax}\big(\frac{\mathbf{Q}\mathbf{K}^\top}{T_d\sqrt{d_k}}\big)\mathbf{V}$ ▷ Eq. (13)
    $\mathbf{z}^x_{\text{out}} \leftarrow \texttt{Transformer}_{\text{ASA}}(\mathbf{z}^x_{\text{caus}})$
    $\hat{\mathbf{a}} \leftarrow \texttt{ActionHead}(\mathbf{z}^{\mathbf{s}}_{\text{out}})$
    // Loss computation and parameter update
    Compute CDA losses only for active types; detach clean anchors in $\mathcal{L}_{\text{caus}}$
    Compute $\mathcal{L}_{\text{caus}}$ by Eqs. (5) and (7), $\mathcal{L}_{\text{orth}}$ by Eq. (8), $\mathcal{L}_{\text{rec}}$ by Eq. (9), $\mathcal{L}_{\text{act}}$ by Eq. (11)
    $\mathcal{L}_{\text{CDA}} \leftarrow \lambda_{\text{caus}}\mathcal{L}_{\text{caus}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}}$ ▷ Eq. (10)
    $\mathcal{L}_{\text{tot}} \leftarrow \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{CDA}}$ ▷ Eq. (12)
    Update $\boldsymbol{\theta}$ using AdamW on $\mathcal{L}_{\text{tot}}$
    if $(\text{epoch} - 1) \bmod 5 = 0$ **then**
      Update T_d using AdamW on $\mathcal{L}_{\text{tot}}$
    end if
  end for
end for
return $\boldsymbol{\theta}$, T_d

A.3 ALGORITHMIC WORKFLOW OF CDA-DT

To facilitate reproducibility, we summarize the complete training procedure of CDA-DT in Algorithm 1.

A.4 FURTHER IMPLEMENTATION DETAILS

Network architecture. CDA-DT uses three Transformer layers with one attention head per layer. The embedding dimension is 128. The input sequences follow the return–state–action order. Each

causal embedding predictor and reconstruction head consists of two linear layers with a GELU activation. The sparse temperatures are constrained to $[10^{-4}, 100]$.

Training. We train CDA-DT for 100 epochs with 1,000 gradient updates per epoch. The batch size is set to 64. The AdamW optimizer (Loshchilov & Hutter, 2017) is used with a learning rate of 10^{-4}, weight decay of 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and a gradient-norm limit of 0.25. The learning rate is linearly warmed up over the first 10% of updates. During training, the input sequences consist of at most 20 state transitions, and shorter sequences are right-padded with zeros.

Evaluation. Every ten training epochs, we evaluate CDA-DT for ten episodes. The context window is limited to 20 state transitions. Gaussian policies use their mean actions during evaluation. Rollouts terminate when the environment returns a terminal signal or after 1,000 steps.

A.5 Computational cost

Table 3 reports the number of parameters used at inference for each method across the nine environments. For the value-based methods (BC, RBC, UWMSG, RIQL), only the actor network is required at inference, so the critics are excluded. For DT, RDT, and DeFog, training-only prediction heads are excluded. For CDA-DT, only the causal encoding path is used at inference, so the corruption heads, decoders, and dynamics predictors are excluded. Consequently, CDA-DT incurs only a marginal increase in the number of inference-time parameters compared with DT.

Table 3: Number of parameters used at inference (in millions, M) for each method on each of the nine environments.

Environment	BC	RBC	UWMSG	RIQL	DT	DeFog	RDT	CDA-DT (ours)
door (1%)	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76
hammer (1%)	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76
relocate (1%)	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76
halfcheetah (10%)	0.07	0.07	0.07	0.07	0.73	0.86	0.73	0.75
hopper (10%)	0.07	0.07	0.07	0.07	0.73	0.85	0.73	0.75
walker2d (10%)	0.07	0.07	0.07	0.07	0.73	0.86	0.73	0.75
kitchen-complete	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76
kitchen-mixed	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76
kitchen-partial	0.08	0.08	0.09	0.08	0.74	0.86	0.74	0.76

A.6 Detailed results of per-module improvements

Table 4 presents the detailed results of per-module improvements. Compared with the base model (CDA-DT without either CDA or ASA), $\mathcal{L}_{\text{caus}}$ alone improves 33 of 54 tasks. Adding $\mathcal{L}_{\text{orth}}$ raises this count to 41, and full CDA with all three loss functions reaches 44. Replacing standard sparse attention with ASA raises the number of improved tasks from 28 to 39.

A.7 Visualization of CDA representations

By using t-SNE, we visualize the causal and corruption embeddings extracted from clean states and their corrupted counterparts on the "kitchen-complete-rnd-sta" task (seed 0), as shown in Fig. 5. Under both input conditions, the causal and corruption embeddings exhibit clear separation in the projected space, providing qualitative evidence that CDA produces differentiated representation subspaces.

A.8 Detailed results of corrupted attention and sparse temperatures

As shown in Fig. 6 and Fig. 7, the trends of corrupted attention and sparse temperatures under six corruption types are broadly consistent with the trends in Fig. 4(a) and Fig. 4(b), respectively. Specifically, across all corruption types, ASA consistently reduces the attention leakage to corrupted tokens compared with both the base model and standard sparse attention, and the layer-wise sparse

Table 4: Per-task results of per-module improvements. Results are reported as mean ± sample standard deviation (over four random seeds). **Bold**: methods that outperform the base model (ties broken by lower standard deviation). "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption. "Base model": CDA-DT without either CDA or ASA. "+SA": base model with standard sparse attention. "+ASA": base model with our adaptive sparse attention. "+$\mathcal{L}_{\text{caus}}$": base model with causal dynamics loss $\mathcal{L}_{\text{caus}}$. "+$\mathcal{L}_{\text{caus\&orth}}$": base model with $\mathcal{L}_{\text{caus}}$ and orthogonality loss $\mathcal{L}_{\text{orth}}$. "+CDA": base model with full CDA that comprises three loss functions.

Task	Corrupt.		Base model	+SA	+ASA	+$\mathcal{L}_{\text{caus}}$	+$\mathcal{L}_{\text{caus\&orth}}$	+CDA
door (1%)	adv	act	103.35±0.88	**104.13±0.35**	**103.92±0.31**	**103.50±0.92**	**103.66±1.16**	**103.71±0.53**
		sta	105.41±0.24	**105.78±0.15**	**105.79±0.31**	105.41±0.32	**105.47±0.35**	**105.60±0.31**
		rew	105.13±0.72	**105.30±0.23**	**105.48±0.24**	**105.35±0.27**	**105.44±0.21**	**105.25±0.45**
	rnd	act	105.51±0.56	105.24±0.97	**105.80±0.40**	105.35±0.55	**105.84±0.10**	**105.56±0.19**
		sta	105.45±0.27	**105.87±0.13**	**105.98±0.06**	105.37±0.37	**105.62±0.08**	**105.45±0.15**
		rew	105.17±0.13	**105.44±0.30**	**105.19±0.27**	104.93±0.37	104.95±0.49	**105.28±0.32**
hammer (1%)	adv	act	122.45±3.34	122.05±6.21	**124.47±1.06**	**124.17±0.98**	**124.43±0.72**	116.91±14.44
		sta	126.76±0.14	126.51±1.13	**126.78±0.98**	126.70±0.31	126.67±0.36	**127.04±0.51**
		rew	127.18±0.87	**127.34±0.62**	**127.56±0.55**	**127.57±0.80**	126.90±0.14	127.11±0.20
	rnd	act	126.84±0.55	126.79±0.45	**126.96±0.34**	126.60±0.72	126.69±0.48	126.50±0.44
		sta	126.59±0.42	126.39±1.56	**126.93±0.56**	**127.02±0.41**	**126.87±0.68**	**127.01±0.46**
		rew	127.30±0.51	126.98±0.23	**127.31±0.31**	**127.57±0.55**	**127.65±0.39**	**127.44±0.51**
relocate (1%)	adv	act	52.80±4.71	51.63±8.87	**57.53±6.55**	**60.51±9.63**	**63.02±9.92**	**71.00±11.93**
		sta	83.46±8.76	83.20±6.01	**84.43±5.94**	**99.14±3.22**	**100.89±2.94**	**98.04±2.31**
		rew	83.38±3.45	82.26±4.35	**83.82±3.77**	**99.86±3.08**	**102.90±4.07**	**96.09±8.86**
	rnd	act	72.46±8.38	70.53±2.82	**72.86±3.32**	**94.50±9.31**	**98.30±1.58**	**95.78±4.00**
		sta	73.49±3.58	**73.96±4.28**	**76.65±4.11**	**88.42±2.86**	**86.98±3.61**	**85.31±6.51**
		rew	96.94±5.23	89.90±4.46	**97.24±4.23**	**102.60±5.20**	**97.07±1.96**	**99.08±5.76**
halfcheetah (10%)	adv	act	19.54±2.58	**21.69±1.12**	**22.49±1.76**	**24.56±1.02**	**24.76±1.90**	**28.00±2.28**
		sta	11.69±1.36	**14.01±1.34**	**14.27±0.77**	10.94±1.37	**11.77±1.77**	**11.86±1.00**
		rew	31.84±3.53	**32.38±1.65**	**32.19±3.14**	**34.46±2.13**	**32.40±1.59**	**34.00±1.59**
	rnd	act	22.82±2.66	**23.00±2.42**	**23.34±1.82**	**27.55±1.45**	**27.58±2.14**	**29.14±2.33**
		sta	12.80±0.31	**15.93±1.86**	**16.79±1.22**	**13.79±0.96**	**15.31±1.40**	**14.50±1.88**
		rew	29.44±1.34	**33.48±2.22**	**30.17±1.54**	**34.07±1.71**	**35.37±0.74**	**34.85±0.87**
hopper (10%)	adv	act	46.42±8.42	**50.41±6.90**	**51.92±2.17**	**51.42±5.83**	**58.55±3.78**	**56.19±3.19**
		sta	53.70±2.59	50.84±1.38	51.33±6.43	**59.48±4.57**	**57.53±3.90**	**60.85±3.35**
		rew	59.81±14.84	58.69±10.56	58.48±5.31	55.43±13.87	**64.90±5.83**	**64.09±0.83**
	rnd	act	48.92±5.79	45.75±8.84	48.40±6.34	**51.61±4.70**	**55.40±1.64**	**53.30±4.25**
		sta	50.22±4.03	**50.74±8.36**	**51.01±6.16**	**56.85±4.36**	**54.01±7.51**	**53.01±6.63**
		rew	55.17±4.22	**60.01±5.78**	**60.05±7.58**	**63.34±2.22**	**63.77±3.43**	**67.29±5.13**
walker2d (10%)	adv	act	36.58±5.62	33.45±3.64	32.99±5.57	35.33±3.12	**41.28±14.44**	**42.92±8.25**
		sta	42.02±9.44	**52.46±2.93**	**45.56±5.76**	32.24±1.63	34.37±3.53	34.41±7.33
		rew	54.44±4.63	53.91±7.45	**55.40±8.66**	**60.73±6.26**	**56.99±6.83**	**59.45±5.91**
	rnd	act	52.36±5.65	**52.67±8.16**	48.96±3.11	**52.66±3.95**	**55.43±4.02**	**54.94±7.31**
		sta	42.63±6.29	38.25±2.67	**42.87±5.82**	31.61±4.77	31.51±6.28	36.48±8.16
		rew	54.58±6.94	**58.54±8.65**	**56.21±3.46**	**58.87±4.14**	**56.16±8.19**	**56.89±5.65**
kitchen-complete	adv	act	28.75±2.50	**29.38±3.15**	28.12±4.27	**31.25±5.95**	**30.62±4.27**	**31.25±4.79**
		sta	81.88±3.75	80.62±4.73	**83.75±3.23**	80.62±5.91	80.62±5.54	76.88±4.27
		rew	81.88±4.73	**85.00±7.36**	81.25±2.50	81.25±1.44	**82.50±5.40**	**82.50±8.66**
	rnd	act	60.00±6.12	58.12±4.27	56.88±3.15	57.50±3.54	58.12±2.39	**60.62±2.39**
		sta	62.50±3.54	**68.12±3.75**	**71.25±4.79**	61.88±3.75	58.75±1.44	**63.12±3.75**
		rew	77.50±4.56	**81.88±7.47**	**82.50±6.45**	**86.25±5.20**	**89.38±3.15**	**80.00±5.40**
kitchen-mixed	adv	act	50.00±2.04	**55.00±7.07**	**54.38±5.54**	50.00±4.08	**53.75±3.23**	**51.88±1.25**
		sta	53.75±7.50	48.75±4.33	51.88±5.54	**56.25±9.24**	**59.38±6.88**	**58.12±4.27**
		rew	54.38±3.75	54.38±7.47	**56.25±5.95**	**62.50±6.45**	**63.12±3.15**	**66.25±10.31**
	rnd	act	58.12±8.75	57.50±7.36	55.62±2.39	56.88±3.75	54.38±6.25	54.38±4.27
		sta	47.50±2.89	45.00±4.56	45.62±4.27	**52.50±6.12**	**53.75±5.95**	**53.75±5.95**
		rew	60.62±4.73	**61.25±5.20**	56.25±3.23	58.75±3.23	**65.00±6.45**	53.12±2.39
kitchen-partial	adv	act	46.88±12.48	41.25±5.20	43.12±8.98	45.62±11.97	38.75±5.95	37.50±11.73
		sta	51.88±15.60	**66.88±8.98**	**63.12±8.98**	**67.50±3.54**	**70.00±2.89**	**64.38±5.54**
		rew	61.88±5.54	56.88±5.54	56.25±7.50	**70.62±8.51**	**65.00±7.36**	**63.12±2.39**
	rnd	act	60.00±8.42	**60.62±9.87**	**64.38±9.66**	55.62±4.27	55.00±3.54	**66.25±7.50**
		sta	53.75±11.09	**58.75±11.09**	**58.75±7.22**	**68.12±7.74**	**68.75±7.77**	**63.75±5.20**
		rew	70.00±3.54	63.12±7.18	64.38±1.25	64.38±3.75	65.00±5.40	59.38±7.18
No. of improved tasks			–	28	39	33	41	44

Table 5: Per-task results under variations of key hyperparameters. Results are reported as mean ± sample standard deviation (over four random seeds). **Bold blue**: the highest scores (ties broken by lower standard deviation). "ASA −50%" and "ASA +50%" set ASA's initial sparse temperatures to 50% and 150% of their default values, respectively. "CDA −50%" and "CDA +50%" apply the same scaling to all three CDA loss coefficients. "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption.

Task	Corrupt.		CDA-DT	CDA −50%	CDA +50%	ASA −50%	ASA +50%
door (1%)	adv	act	102.95±1.44	103.03±0.50	103.22±0.73	**103.83±0.46**	103.24±0.79
		sta	105.67±0.37	105.65±0.20	**105.88±0.20**	105.71±0.39	105.53±0.40
		rew	105.13±0.22	**105.59±0.38**	105.51±0.25	105.40±0.34	105.49±0.29
	rnd	act	105.36±0.11	**105.53±0.46**	105.12±0.25	105.44±0.45	105.32±0.54
		sta	105.50±0.43	**105.82±0.27**	105.56±0.39	105.76±0.28	105.73±0.33
		rew	105.24±0.39	**105.42±0.41**	105.33±0.36	104.95±0.37	105.10±0.37
hammer (1%)	adv	act	122.83±1.82	117.46±6.93	123.37±1.35	**123.55±1.26**	122.89±2.20
		sta	127.02±0.32	**127.32±0.33**	126.99±0.38	127.24±0.59	126.80±0.63
		rew	**127.73±0.08**	127.71±0.92	127.56±0.42	127.58±0.57	127.32±0.52
	rnd	act	**127.77±0.87**	127.41±1.05	127.26±0.57	127.63±0.76	127.70±0.94
		sta	126.91±0.25	126.60±0.62	126.82±0.47	126.95±0.49	**127.21±0.53**
		rew	127.32±0.29	127.35±0.36	127.46±0.18	127.48±0.38	**127.80±0.24**
relocate (1%)	adv	act	69.88±6.68	63.48±5.05	70.30±4.54	**76.95±13.05**	67.47±8.44
		sta	93.09±3.85	92.79±5.11	93.98±5.15	**95.75±2.61**	92.99±5.29
		rew	98.43±2.58	**100.31±3.00**	98.91±6.42	97.50±5.36	97.52±4.85
	rnd	act	92.38±4.19	93.84±4.34	89.65±3.43	89.14±3.97	**94.00±3.91**
		sta	**84.38±7.45**	79.66±5.40	82.34±10.47	80.08±1.52	74.75±2.97
		rew	95.51±6.24	99.04±2.58	97.24±4.27	**100.96±3.86**	99.85±4.26
halfcheetah (10%)	adv	act	22.94±1.42	25.35±0.47	21.81±1.35	23.59±1.20	**25.97±2.50**
		sta	13.86±0.75	14.92±1.81	**15.13±2.18**	11.55±1.82	14.07±0.47
		rew	34.95±1.19	33.72±0.37	**35.27±1.41**	34.82±1.72	34.89±0.66
	rnd	act	26.73±0.98	26.98±1.59	26.76±1.69	27.73±1.53	**28.39±2.15**
		sta	18.36±2.85	**19.08±1.21**	18.38±2.09	15.39±2.84	18.00±1.09
		rew	**35.40±0.77**	35.23±0.79	34.39±2.41	34.49±1.16	34.41±1.65
hopper (10%)	adv	act	58.49±4.28	58.51±3.87	57.13±6.65	58.40±4.90	**60.98±3.59**
		sta	60.45±3.06	56.76±4.54	56.32±3.22	**60.57±2.05**	55.53±2.28
		rew	57.80±5.15	**58.16±5.00**	54.57±11.90	52.81±4.96	52.20±7.83
	rnd	act	53.03±4.46	53.90±4.55	**54.26±5.92**	53.42±1.83	53.90±3.61
		sta	**56.09±4.11**	53.88±2.55	55.97±1.36	56.04±4.15	55.59±2.97
		rew	58.69±4.26	53.80±2.48	54.21±7.79	52.48±10.38	**61.44±4.53**
walker2d (10%)	adv	act	32.07±1.91	**34.26±4.92**	27.71±6.82	31.05±1.51	30.97±4.91
		sta	44.04±9.46	44.41±9.44	43.85±4.73	37.16±3.43	**47.84±7.95**
		rew	55.29±7.36	57.80±3.76	56.21±8.77	**59.56±2.87**	57.37±4.86
	rnd	act	50.52±4.97	54.50±9.19	49.71±3.28	45.82±2.81	**55.12±5.93**
		sta	34.66±4.97	36.43±6.93	33.44±4.76	39.59±2.75	**42.97±4.47**
		rew	57.25±8.48	58.17±3.66	57.23±4.75	53.19±7.20	**63.89±2.92**
kitchen-complete	adv	act	30.00±4.56	29.38±4.27	30.62±1.25	**34.38±5.54**	28.75±3.23
		sta	**86.25±6.61**	82.50±2.04	84.38±6.57	83.12±6.25	80.62±3.15
		rew	84.38±5.54	81.88±2.39	**90.62±4.27**	82.50±4.56	80.62±4.27
	rnd	act	58.12±2.39	58.75±2.50	54.38±5.15	61.25±4.33	**64.38±2.39**
		sta	68.75±3.23	68.12±5.54	68.12±3.75	68.12±3.15	**70.00±5.40**
		rew	83.12±4.27	77.50±3.54	83.12±6.25	**85.62±5.15**	80.62±3.15
kitchen-mixed	adv	act	**55.00±6.12**	53.75±2.50	52.50±5.40	51.88±6.57	54.38±5.91
		sta	**56.25±4.79**	51.25±5.95	55.62±4.27	55.62±6.25	52.50±5.40
		rew	59.38±6.88	53.75±4.33	**60.00±4.56**	57.50±7.36	57.50±2.04
	rnd	act	56.88±5.54	56.88±5.54	56.88±5.15	**61.25±3.23**	55.00±4.56
		sta	**53.75±1.44**	51.25±1.44	51.88±3.75	51.88±2.39	51.25±4.79
		rew	56.25±9.46	56.25±5.20	**61.88±4.27**	55.62±2.39	52.50±2.04
kitchen-partial	adv	act	43.12±7.47	39.38±4.73	**45.62±16.25**	38.75±5.20	31.88±10.48
		sta	66.88±5.91	65.62±8.26	**71.25±4.33**	66.25±5.95	67.50±7.36
		rew	63.75±4.79	**65.00±5.40**	59.38±6.88	61.88±6.88	64.38±13.29
	rnd	act	60.00±6.12	61.25±7.77	**61.25±1.44**	58.75±5.95	58.75±4.33
		sta	63.12±5.54	**69.38±1.25**	58.75±5.20	60.62±5.54	61.88±6.25
		rew	**68.75±1.44**	66.88±3.15	61.88±3.15	68.75±6.29	66.25±4.79
Overall average			**70.51**	69.97	70.06	70.06	70.17

Table 6: Detailed robustness evaluation results under both training corruptions and testing attacks. Results are reported as mean ± sample standard deviation (over four random seeds). **Bold blue**: the highest scores (ties broken by lower standard deviation). "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption. CDA-DT achieves the highest scores on 40 out of 54 (74%) tasks, surpassing all baselines across all six corruption types.

Task	Corrupt.		BC	RBC	UWMSG	RIQL	DT	DeFog	RDT	CDA-DT (ours)
door (1%)	adv	act	6.12±4.44	17.52±12.41	-0.18±0.03	67.75±9.30	86.25±8.58	0.33±0.54	**102.30±0.48**	101.07±2.01
		sta	29.99±7.77	30.99±12.12	-0.20±0.00	78.51±12.32	103.35±0.42	104.57±0.40	105.16±0.26	**105.65±0.27**
		rew	94.35±9.50	101.60±1.26	-0.20±0.01	61.50±20.77	105.05±0.42	104.99±0.16	105.03±0.58	**105.13±0.22**
	rnd	act	37.97±10.14	40.95±8.60	-0.19±0.04	51.25±12.05	81.76±8.46	0.06±0.03	**98.39±3.84**	89.04±6.30
		sta	59.82±13.36	58.17±25.75	-0.07±0.16	94.59±8.47	97.08±4.69	104.98±0.17	104.83±0.13	**105.46±0.50**
		rew	94.35±9.50	101.60±1.26	-0.24±0.10	84.80±5.65	105.19±0.22	104.92±0.21	104.91±0.29	**105.24±0.39**
hammer (1%)	adv	act	42.64±7.23	47.48±12.47	1.06±0.79	82.73±8.01	87.00±17.01	24.96±0.26	**116.08±5.05**	114.17±1.19
		sta	47.14±10.56	58.99±17.49	0.19±0.08	91.36±11.52	100.80±8.39	116.70±5.91	126.66±0.42	**127.11±0.31**
		rew	111.80±5.84	119.15±7.57	0.26±0.02	80.81±19.93	125.58±0.09	125.31±0.73	127.44±0.66	**127.73±0.08**
	rnd	act	59.65±9.74	72.52±10.69	0.30±0.16	74.99±9.11	103.39±7.73	31.68±7.43	102.37±13.99	**118.96±6.75**
		sta	66.53±4.01	70.75±10.03	0.08±0.09	88.51±7.18	113.50±9.42	118.76±5.82	122.06±6.73	**126.69±0.14**
		rew	111.80±5.84	119.15±7.57	0.39±0.28	94.50±11.86	124.29±2.36	125.78±0.89	126.89±0.31	**127.32±0.29**
relocate (1%)	adv	act	13.49±5.19	16.33±7.73	-0.30±0.03	10.75±5.06	18.91±2.81	3.31±3.52	**66.49±4.00**	63.38±6.06
		sta	28.13±9.69	26.04±10.73	-0.30±0.03	26.22±11.13	80.76±6.70	46.63±7.13	81.93±7.90	**94.33±5.21**
		rew	56.15±8.67	58.72±12.40	-0.30±0.03	19.51±4.47	91.37±9.19	71.01±6.41	85.21±5.38	**99.62±3.24**
	rnd	act	31.61±16.22	38.65±11.17	-0.30±0.03	22.30±7.96	53.14±7.31	5.48±4.28	68.03±1.96	**77.42±6.40**
		sta	22.42±4.02	21.14±5.98	-0.30±0.03	23.50±5.50	68.54±6.95	29.98±2.99	74.98±12.07	**82.67±9.60**
		rew	56.15±8.67	58.72±12.40	-0.30±0.03	22.02±13.74	91.54±11.07	72.34±9.54	**96.66±5.58**	95.97±7.02
halfcheetah (10%)	adv	act	0.71±0.26	0.75±0.31	5.19±0.84	1.56±0.23	12.49±2.37	2.26±0.00	**16.06±1.15**	14.36±1.15
		sta	1.80±0.18	2.17±0.31	5.00±1.13	0.56±0.15	**9.08±1.10**	2.27±0.00	6.39±0.99	8.18±1.52
		rew	6.26±2.03	5.89±1.49	2.18±0.39	13.74±1.33	21.35±2.02	2.27±0.00	26.74±2.31	**33.92±1.35**
	rnd	act	1.18±0.28	1.28±0.35	5.73±1.42	2.80±0.30	10.45±1.39	2.26±0.00	14.72±2.40	**16.01±1.27**
		sta	1.33±0.59	1.46±0.53	3.36±0.66	0.39±0.11	10.13±1.93	2.27±0.00	11.92±1.62	**13.33±0.93**
		rew	6.26±2.03	5.89±1.49	3.10±0.58	11.30±1.54	23.77±0.66	2.27±0.00	23.35±2.59	**34.36±1.53**
hopper (10%)	adv	act	14.80±1.71	13.07±2.42	31.87±7.96	21.98±3.01	28.76±2.70	0.64±0.00	35.11±3.88	**43.23±5.88**
		sta	17.68±2.77	15.46±2.42	18.46±3.68	8.44±1.76	**34.26±2.90**	0.65±0.00	24.21±6.46	23.88±2.21
		rew	29.79±7.59	30.01±6.22	47.08±16.79	45.93±5.42	53.38±5.82	0.67±0.01	**56.39±6.58**	56.31±7.24
	rnd	act	17.25±2.54	16.46±0.93	29.30±3.78	25.98±1.72	28.45±2.33	0.64±0.00	33.42±3.40	**39.35±5.54**
		sta	11.49±1.52	12.42±3.04	21.24±2.03	9.27±1.14	31.81±1.29	0.65±0.00	30.26±4.40	**33.19±3.04**
		rew	29.79±7.59	30.01±6.22	27.66±4.23	40.45±2.32	38.61±5.10	0.67±0.01	30.52±3.44	**40.65±9.81**
walker2d (10%)	adv	act	2.82±0.20	3.22±0.82	6.10±0.77	3.41±1.06	16.07±2.15	-0.03±0.00	16.94±2.04	**19.68±1.79**
		sta	6.54±1.44	7.78±0.55	5.38±1.55	2.57±0.69	**25.40±1.76**	-0.02±0.00	14.95±1.10	22.75±6.14
		rew	18.02±2.11	20.24±7.52	24.58±8.09	15.18±4.18	46.64±3.89	-0.01±0.01	53.77±3.54	**58.26±4.95**
	rnd	act	7.15±2.31	6.25±2.61	9.12±1.74	8.26±1.65	24.33±3.45	-0.03±0.00	30.49±2.16	**32.27±0.92**
		sta	5.27±1.04	5.32±0.52	-0.53±0.28	3.63±1.84	18.90±1.74	-0.02±0.00	20.13±1.74	**21.88±3.46**
		rew	18.02±2.11	20.24±7.52	11.50±4.69	12.83±3.64	**39.86±4.09**	-0.01±0.00	35.16±6.80	30.57±8.54
kitchen-complete	adv	act	7.50±2.89	5.62±3.75	0.00±0.00	13.75±1.44	13.75±5.20	10.00±0.00	13.12±3.15	**16.25±2.50**
		sta	6.88±3.15	10.00±5.40	0.00±0.00	28.12±11.43	44.38±7.74	53.12±4.73	75.62±4.27	**79.38±5.15**
		rew	36.25±21.26	50.62±6.25	0.00±0.00	60.62±7.47	**91.25±4.33**	65.62±6.88	80.00±4.56	83.12±3.15
	rnd	act	8.12±6.25	5.62±1.25	0.00±0.00	11.25±3.23	13.75±3.23	6.88±1.25	21.25±2.50	**24.38±2.39**
		sta	10.00±4.08	9.38±2.39	0.00±0.00	34.38±1.25	50.62±3.75	53.12±6.57	64.38±10.48	**65.62±1.25**
		rew	36.25±21.26	50.62±6.25	0.00±0.00	65.62±5.54	**96.88±2.39**	66.25±1.44	84.38±4.27	83.12±5.15
kitchen-mixed	adv	act	11.25±6.29	21.25±5.20	0.00±0.00	6.25±1.44	6.25±2.50	4.38±1.25	26.25±4.33	**33.12±6.57**
		sta	16.88±5.15	16.88±6.57	0.00±0.00	20.00±5.00	45.00±0.00	24.38±14.77	50.00±3.54	**51.25±7.22**
		rew	51.88±2.39	57.50±5.40	0.00±0.00	48.75±4.79	58.75±4.79	23.75±13.62	55.62±4.73	**60.62±6.25**
	rnd	act	25.00±3.54	26.25±2.50	0.62±1.25	22.50±2.04	31.88±1.25	4.38±1.25	34.38±4.27	**35.62±6.57**
		sta	11.25±1.44	15.62±2.39	0.00±0.00	5.00±2.04	44.38±2.39	25.00±9.35	**50.62±1.25**	**50.62±1.25**
		rew	51.88±2.39	57.50±5.40	0.00±0.00	55.00±10.61	54.38±5.15	26.88±16.50	**61.88±1.25**	57.50±8.66
kitchen-partial	adv	act	11.88±5.15	18.12±5.54	5.00±8.42	0.00±0.00	8.75±3.23	7.50±2.04	24.38±2.39	**28.12±7.47**
		sta	23.12±2.39	20.62±5.15	0.00±0.00	15.00±2.04	40.00±10.21	23.12±3.75	60.62±6.88	**61.25±5.95**
		rew	45.00±2.04	49.38±3.75	0.00±0.00	46.88±9.44	58.12±6.57	22.50±3.54	60.00±5.40	**63.12±5.54**
	rnd	act	27.50±2.04	26.25±6.29	0.00±0.00	22.50±3.54	30.62±2.39	4.38±3.15	36.25±4.79	**37.50±2.89**
		sta	16.25±3.23	16.88±3.75	0.00±0.00	5.00±3.54	43.75±10.51	19.38±5.15	53.75±4.33	**56.88±1.25**
		rew	45.00±2.04	49.38±3.75	0.00±0.00	63.12±13.13	57.50±3.54	21.25±7.50	69.38±3.15	**70.00±2.04**
Cross-task average	adv	act	12.36	15.93	5.42	23.13	30.91	5.93	46.30	**48.15**
		sta	19.80	20.99	3.17	30.09	53.67	41.27	60.62	**63.75**
		rew	49.94	54.79	8.18	43.66	72.39	46.23	72.24	**76.43**
	rnd	act	23.94	26.03	4.95	26.87	41.97	6.19	48.81	**52.28**
		sta	22.71	23.46	2.64	29.36	53.19	39.35	59.21	**61.82**
		rew	49.94	54.79	4.68	49.96	70.22	46.71	70.35	**71.64**

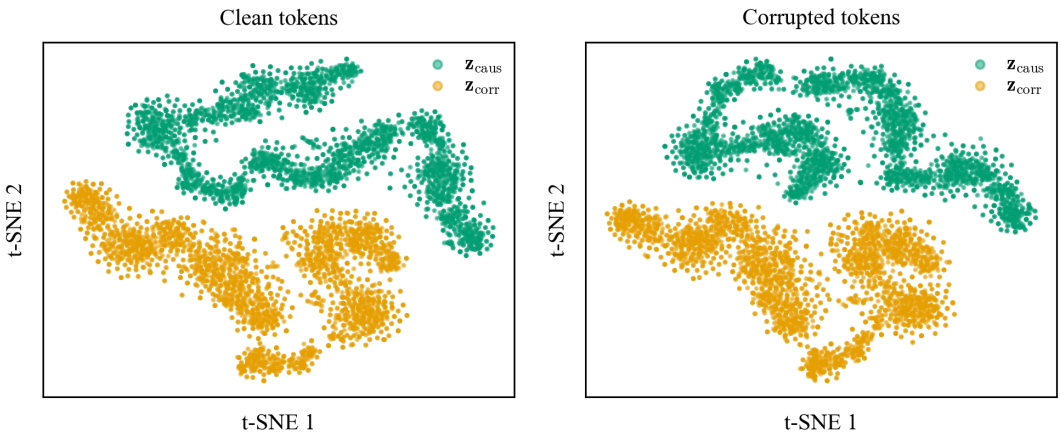

Figure 5: The t-SNE visualization of causal embeddings $\mathbf{z}_{\text{caus}}$ and corruption embeddings $\mathbf{z}_{\text{corr}}$ under clean and corrupted states on the "kitchen-complete-rnd-sta" task (seed 0).

temperatures exhibit a similar hierarchical pattern where deeper layers converge to lower temperatures that more aggressively filter out corrupted inputs.

A.9 Hyperparameter sensitivity analysis

To assess CDA-DT's robustness to its hyperparameters, we conduct sensitivity analyses on all key hyperparameters of CDA and ASA. Results are presented in Table 5. Relative score drops are computed from the displayed per-task mean scores as the percentage decrease in the mean over all 54 tasks relative to the default configuration, with rounding applied only to the final percentages.

For CDA's three loss weights (i.e., λ_{caus}, λ_{orth}, and λ_{rec} in Eq. (10)), we decrease (resp. increase) them by 50%, and the average score drops by only 0.76% (resp. 0.64%), indicating that CDA is robust to the choice of loss coefficients. We decrease (resp. increase) ASA's initial sparse temperatures by 50%, and the average score drops by only 0.63% (resp. 0.49%), showing that ASA's performance is robust to the initialization of sparse temperatures.

A.10 Detailed results under both training corruptions and testing attacks

Table 6 presents the per-task results under both training corruptions and testing attacks. The experimental setups are identical to those in the main experiments. CDA-DT achieves the highest scores on 40 out of 54 (74%) tasks. CDA-DT also achieves the highest cross-task average for each of the six corruption types.

A.11 Discreteness of kitchen evaluation scores

The kitchen environments in D4RL assign discrete scores: the normalized score of each rollout is an integer multiple of 25, so the mean over the four seeds reported in our tables is a multiple of 0.625. This quantization can make different methods yield identical summary statistics. For example, in Table 1, two methods both report 60.00 ± 6.12 on the "kitchen-partial-rnd-act" task, although their per-seed scores, $(57.5, 65, 65, 52.5)$ and $(55, 67.5, 55, 62.5)$, differ. Such coincidences thus arise from score discretization.

A.12 Comparison with CQL

CQL is neither a corruption-robust offline RL method nor a base algorithm of the evaluated baselines. Given this scope and the page limit of the main text, we present its results in this appendix.

Table 7 compares CQL with CDA-DT on the 54 tasks in Table 1. CDA-DT achieves a higher score on all 54 tasks and a higher cross-task average for each of the six corruption types.

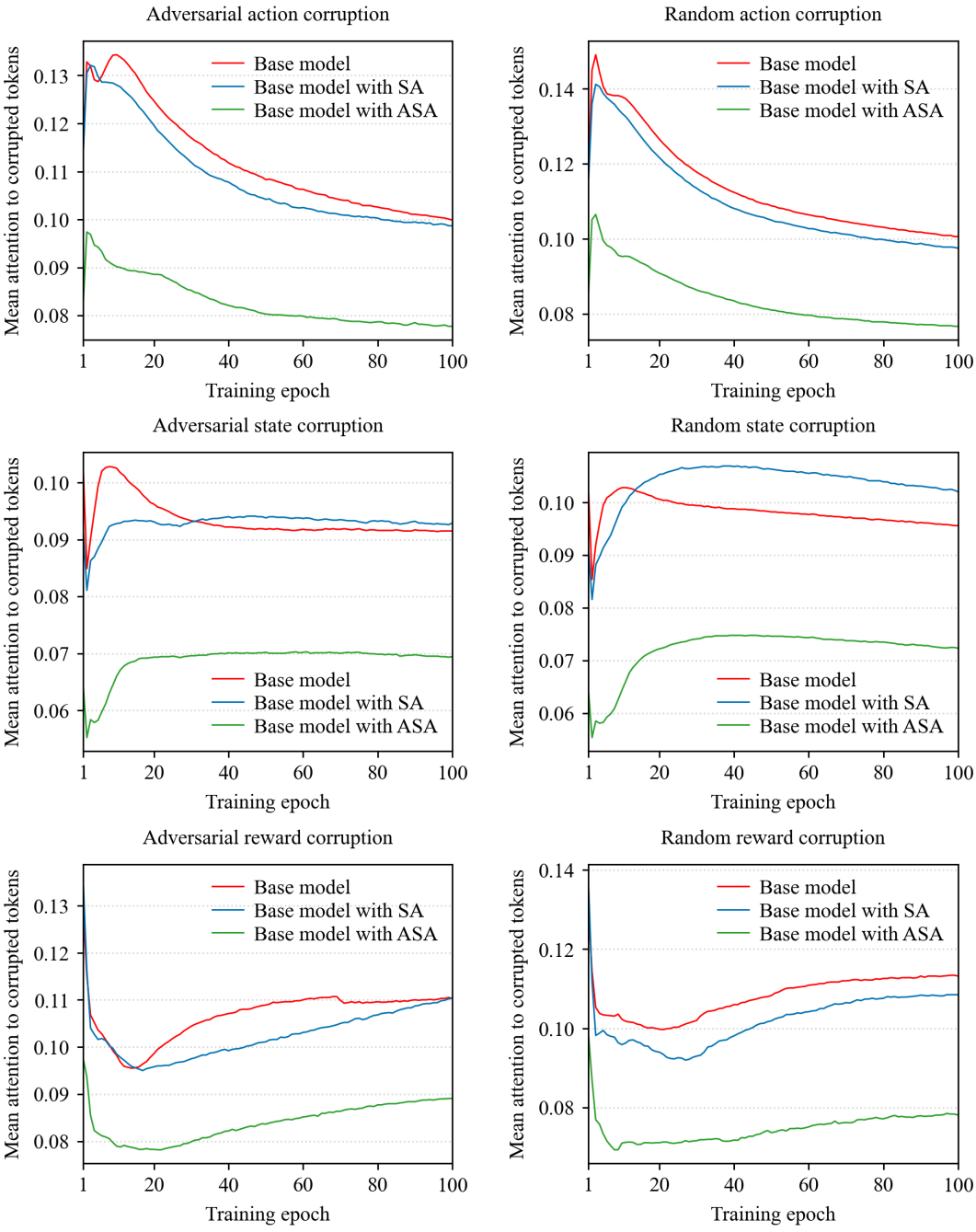

Figure 6: Per-corruption-type corrupted attention throughout training. The curves show the mean attention weight that non-corrupted query tokens assign to corrupted key tokens. "Base model": CDA-DT without either CDA or ASA. "+SA": base model with standard sparse attention. "+ASA": base model with our adaptive sparse attention. Results are aggregated over all tasks of each corruption type and four random seeds.

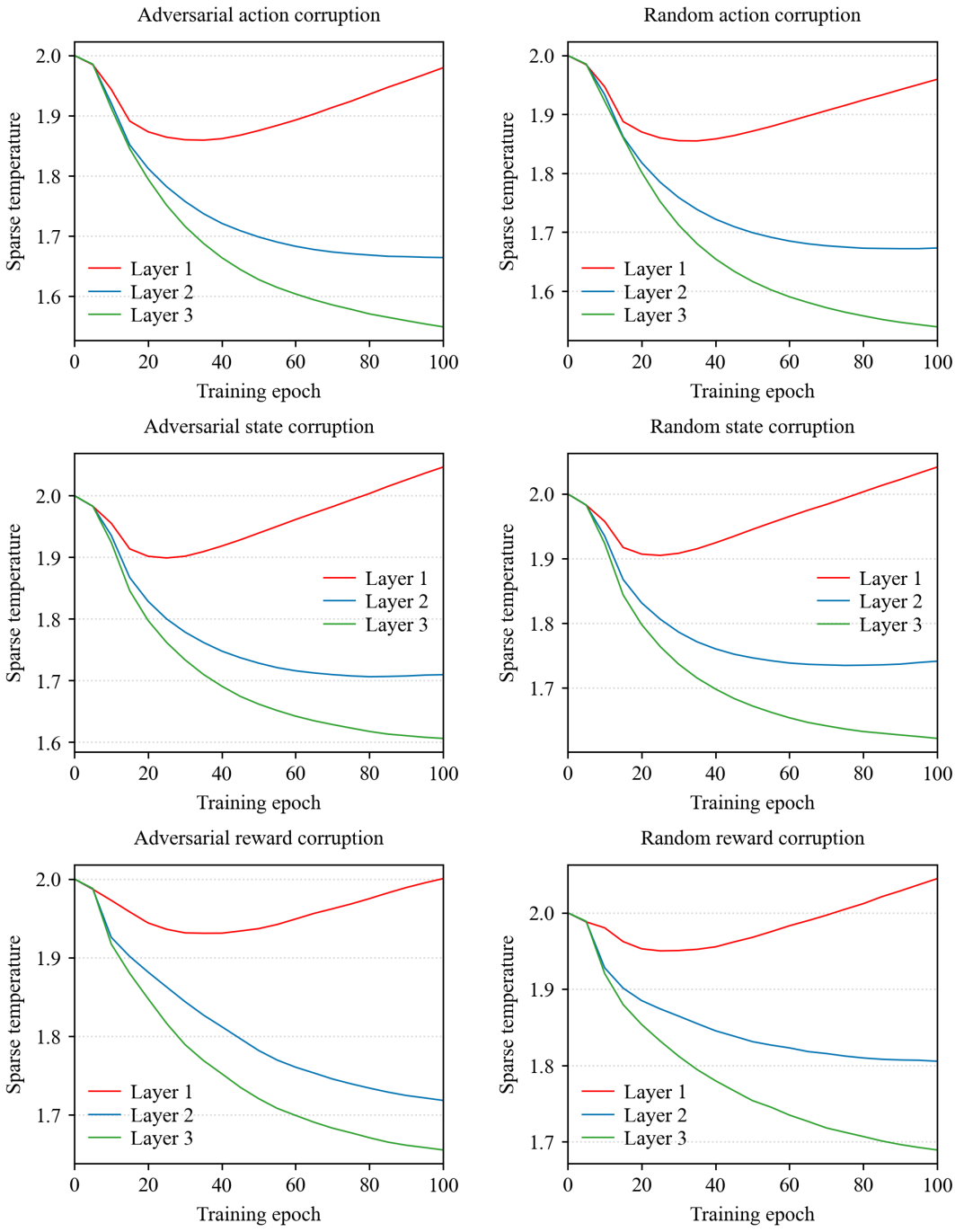

Figure 7: CDA-DT's sparse temperatures throughout training for each corruption type. The curves show each Transformer layer's mean sparse temperature after every five-epoch update. "Layer 1": the shallowest Transformer layer. "Layer 3": the deepest Transformer layer. Results are aggregated over all tasks of each corruption type and four random seeds.

Table 7: Comparison of CQL and CDA-DT under training data corruptions. Results are reported as mean ± sample standard deviation over four random seeds, in the same task order as Table 1. **Bold blue** denotes the higher score (ties broken by lower standard deviation). "adv": adversarial corruption. "rnd": random corruption. "act": action corruption. "sta": state corruption. "rew": reward corruption.

Task	Corrupt.		CQL	CDA-DT (ours)
door (1%)	adv	act	-0.21±0.10	**102.95±1.44**
		sta	-0.27±0.14	**105.67±0.37**
		rew	-0.30±0.04	**105.13±0.22**
	rnd	act	-0.29±0.09	**105.36±0.11**
		sta	-0.34±0.00	**105.50±0.43**
		rew	-0.32±0.04	**105.24±0.39**
hammer (1%)	adv	act	0.24±0.01	**122.83±1.82**
		sta	0.25±0.01	**127.02±0.32**
		rew	0.24±0.01	**127.73±0.08**
	rnd	act	0.25±0.01	**127.77±0.87**
		sta	0.23±0.05	**126.91±0.25**
		rew	0.24±0.01	**127.32±0.29**
relocate (1%)	adv	act	-0.16±0.03	**69.88±6.68**
		sta	-0.25±0.05	**93.09±3.85**
		rew	-0.26±0.08	**98.43±2.58**
	rnd	act	-0.28±0.04	**92.38±4.19**
		sta	-0.24±0.09	**84.38±7.45**
		rew	-0.24±0.08	**95.51±6.24**
halfcheetah (10%)	adv	act	22.05±5.04	**22.94±1.42**
		sta	13.45±6.96	**13.86±0.75**
		rew	34.89±1.69	**34.95±1.19**
	rnd	act	25.73±4.94	**26.73±0.98**
		sta	11.46±2.26	**18.36±2.85**
		rew	35.12±0.87	**35.40±0.77**
hopper (10%)	adv	act	1.83±0.01	**58.49±4.28**
		sta	2.29±1.39	**60.45±3.06**
		rew	2.29±0.53	**57.80±5.15**
	rnd	act	1.55±0.60	**53.03±4.46**
		sta	10.81±12.23	**56.09±4.11**
		rew	1.25±0.68	**58.69±4.26**
walker2d (10%)	adv	act	3.64±1.11	**32.07±1.91**
		sta	0.36±0.35	**44.04±9.46**
		rew	7.64±0.72	**55.29±7.36**
	rnd	act	4.47±3.32	**50.52±4.97**
		sta	0.05±0.13	**34.66±4.97**
		rew	9.62±4.34	**57.25±8.48**
kitchen-complete	adv	act	5.62±7.18	**30.00±4.56**
		sta	23.75±7.77	**86.25±6.61**
		rew	26.88±19.83	**84.38±5.54**
	rnd	act	23.12±3.75	**58.12±2.39**
		sta	22.50±8.42	**68.75±3.23**
		rew	0.00±0.00	**83.12±4.27**
kitchen-mixed	adv	act	0.00±0.00	**55.00±6.12**
		sta	0.00±0.00	**56.25±4.79**
		rew	0.00±0.00	**59.38±6.88**
	rnd	act	0.62±1.25	**56.88±5.54**
		sta	0.00±0.00	**53.75±1.44**
		rew	1.25±2.50	**56.25±9.46**
kitchen-partial	adv	act	0.00±0.00	**43.12±7.47**
		sta	0.00±0.00	**66.88±5.91**
		rew	6.25±12.50	**63.75±4.79**
	rnd	act	6.88±8.00	**60.00±6.12**
		sta	5.00±10.00	**63.12±5.54**
		rew	0.00±0.00	**68.75±1.44**
Cross-task average	adv	act	3.67	**59.70**
		sta	4.40	**72.61**
		rew	8.63	**76.32**
	rnd	act	6.89	**70.09**
		sta	5.50	**67.95**
		rew	5.21	**76.39**

A.13 Environment sizes and dimensions

Table 8 summarizes the dataset sizes and state/action dimensions of the nine environments used in our experiments. We report both transition and trajectory counts for the retained training datasets, together with the percentage of original trajectories retained.

Table 8: Dataset sizes and state/action dimensions for the nine environments. Following RDT, counts refer to the benchmark subsets used in our experiments. The retained trajectory percentage is relative to the original D4RL dataset.

Dataset	**Transitions**	**Trajectories**	**State dimension**	**Action dimension**	**Retained trajectories**
door-expert-v0	10,000	50	39	28	1%
hammer-expert-v0	10,000	50	46	26	1%
relocate-expert-v0	10,000	50	39	30	1%
halfcheetah-medium-replay-v2	20,000	20	17	6	10%
hopper-medium-replay-v2	32,921	204	11	3	10%
walker2d-medium-replay-v2	27,937	109	17	6	10%
kitchen-complete-v0	3,680	19	60	9	100%
kitchen-mixed-v0	136,950	613	60	9	100%
kitchen-partial-v0	136,950	613	60	9	100%

A.14 Limitations

Our method has several limitations. First, the dual-head embedders and auxiliary predictor networks introduce modest computational overhead. Second, while CDA-DT is highly effective against random and adversarial corruptions, its robustness to other data degradation types (e.g., missing data) has not been studied. Third, the update interval for the sparse temperatures (every five epochs) is fixed, and an adaptive schedule may further improve performance.