TEMPORAL-STRUCTURE ROBUSTNESS TRANSFER IN OFFLINE REINFORCEMENT LEARNING: UNCERTAINTY-GUIDED DECISION TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing corruption-robust offline RL methods largely focus on learning reliable policies from corrupted offline datasets, while robustness transfer under changes in the temporal structure of corruption remains less understood. We formulate temporal-structure robustness transfer (TS-RT), an offline RL setting in which policies are trained under temporally independent corruption and tested under previously unseen, temporally correlated corruption. To the best of our knowledge, this work provides the first systematic investigation of TS-RT in offline RL across observation, action, and reward corruption settings. To address TS-RT, we propose Uncertainty-Guided Decision Transformer (UDT), which learns a token-level uncertainty signal to regulate both causal information routing and the training influence of unreliable targets. For controlled evaluation, we introduce Drift-Attack, which generates temporally correlated test-time perturbations through burst–recover dynamics and momentum-driven drift within a matched-channel protocol. Across nine D4RL tasks and seven representative baselines, UDT achieves the highest overall normalized score of 63.52 and ranks first on 11 out of 15 reported aggregate evaluation metrics across corruption types, corruption channels, and environment groups. We provide code and trained checkpoints to facilitate reproducibility.

1 INTRODUCTION

Offline reinforcement learning (offline RL) learns decision-making policies from previously collected datasets, avoiding potentially costly or unsafe online interaction with the environment (Levine et al., 2020; Fujimoto et al., 2019; Kumar et al., 2020; Kostrikov et al., 2022). This makes offline RL attractive for real-world control, yet leaves learned policies critically dependent on logged-data reliability. Observations, actions, and rewards may be corrupted by sensor noise, actuator errors, logging failures, or malicious manipulation. This motivates corruption-robust offline RL methods based on robust value estimation (Zhang et al., 2022; Wu et al., 2022; Yang et al., 2024b), uncertainty-aware reweighting (Ye et al., 2023), data correction, and corruption-robust sequence modeling (Xu et al., 2025).

Robustness transfer across train–test shifts in the temporal correlation of corruption has not been systematically studied in offline RL. Prior work has studied robustness to corrupted offline data (Zhang et al., 2022; Ye et al., 2023; Yang et al., 2024b; Xu et al., 2025) and test-time perturbations (Xu et al., 2025; Yang & Xu, 2024; Ayabe et al., 2025). Temporally structured perturbations are explored in robust RL, while structured non-stationarity is considered in offline RL (Lee et al., 2020; Liang et al., 2024; Vogt-Lowell et al., 2026; Ackermann et al., 2024; Ada et al., 2025). This motivates the following question:

> *Does robustness learned from offline data under temporally independent corruption transfer to previously unseen, temporally correlated corruption at test time?*

We term this problem **temporal-structure robustness transfer (TS-RT)**. To the best of our knowledge, this work provides the **first** systematic investigation of TS-RT in offline RL across observation, action, and reward corruptions.

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.

The central challenge of TS-RT is that robustness learned under temporally independent corruption may fail to transfer when test-time perturbations become temporally correlated. Unlike temporally independent corruption, temporally correlated perturbations can induce errors that persist across consecutive timesteps (Efroni et al., 2022; Swamy et al., 2022); for sequence-based policies, such errors can remain in the causal context and influence multiple subsequent decisions (Chen et al., 2021).

To address TS-RT, we propose **Uncertainty-Guided Decision Transformer (UDT)**. Its core innovation is to make learned token-level uncertainty an explicit control signal within the sequence model. Existing corruption-robust methods primarily improve how a policy is learned from unreliable data, whereas UDT additionally regulates how unreliable trajectory information is used during sequence computation. In UDT, we introduce a shared token-level uncertainty signal that serves two complementary roles. It modulates causal self-attention to reduce the influence of unreliable historical tokens and reweights action and reward supervision through a detached weighting mechanism during training. By coupling reliability-aware information routing with robust learning, UDT enables continuous, token-wise adaptation without explicit corruption labels.

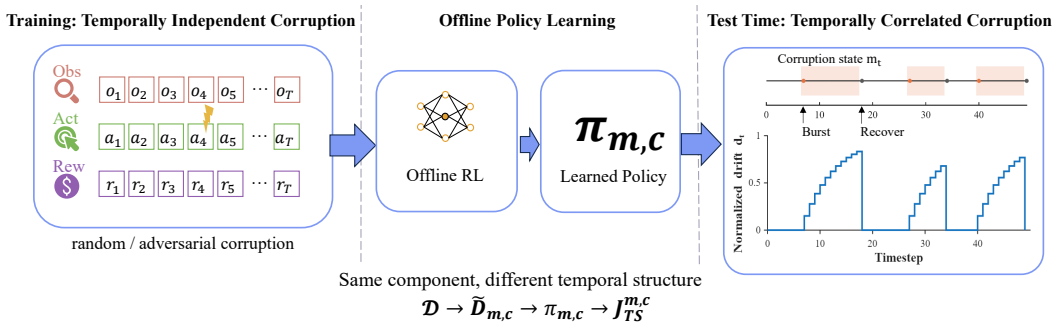

Figure 1: **Temporal-structure robustness transfer (TS-RT).** Policies are trained under temporally independent random or adversarial corruption and evaluated under matched-component, temporally correlated perturbations generated by Drift-Attack, with the corrupted component held fixed while the temporal dependence of corruption changes. The action channel is shown for illustration; the same protocol applies to observation and reward corruption.

For controlled evaluation of TS-RT, we introduce **Drift-Attack**, a test-time corruption process for assessing robustness to temporally correlated perturbations. We build on established training-time corruption protocols in robust offline RL (Yang et al., 2024b; Xu et al., 2025), where random or adversarial corruption is applied without explicit temporal dependence. To control for the corrupted component, we keep the corrupted channel fixed and evaluate policies under previously unseen, temporally correlated perturbations. Drift-Attack implements this shift through burst-recover dynamics and momentum-driven drift, with clean-reference statistics used to normalize perturbation scales across observations, actions, and rewards. Figure 1 illustrates the matched-component TS-RT protocol, in which temporally independent training corruption is replaced by temporally correlated test-time perturbations on the same components.

Our experiments demonstrate that UDT improves robustness transfer under TS-RT. Across nine D4RL tasks spanning MuJoCo, Kitchen, and Adroit, UDT outperforms seven representative baselines and achieves the highest overall normalized score of **63.52**, outperforming DT (58.52) and RDT (58.39) by 8.55% and 8.79%, respectively. Moreover, UDT ranks first on **11 out of 15** reported aggregate evaluation metrics across training corruption types, corrupted components, and benchmark families.

2 Related Work

Corruption-Robust Offline RL. Prior work addresses corrupted offline data through robust estimation and certification, uncertainty-based objectives, and dataset recovery (Zhang et al., 2022; Wu et al., 2022; Chen et al., 2024; Ye et al., 2023; Yang et al., 2024a;b; Liu et al., 2025). RDT further tackles data corruption through sequence modeling with embedding dropout, Gaussian-weighted

learning, and iterative data correction (Xu et al., 2025). Their primary focus is on robust learning from corrupted offline datasets; we instead study how such robustness transfers when the temporal dependence of corruption changes between training and test time.

Test-Time Robustness and Temporally Coupled Perturbations. Test-time robustness in offline RL has been studied under observation perturbations (Xu et al., 2025; Yang & Xu, 2024), action perturbations (Ayabe et al., 2025), and dynamics shifts (Qiao et al., 2026). Recent work also combines corrupted training observations with deployment-time observation delay (Zhao et al., 2026). More broadly, robust RL studies temporally coupled perturbations (Lee et al., 2020; Liang et al., 2024), while offline RL considers structured non-stationarity in offline datasets and time-varying test-time offsets (Ackermann et al., 2024; Ada et al., 2025). In contrast, TS-RT evaluates a matched-component shift: policies trained under temporally independent corruption are tested under previously unseen, temporally correlated corruption on the same component, covering observation, action, and reward channels.

Decision Transformers for Robustness and Uncertainty. Decision Transformer casts RL as conditional sequence modeling (Chen et al., 2021). Subsequent variants address frame dropping, corrupted offline data, and test-time adversaries (Hu et al., 2023; Xu et al., 2025; Tang et al., 2024). Uncertainty-based sequence modeling approaches have also been studied, including uncertainty-aware attention and uncertainty-weighted Decision Transformers (Heo et al., 2018; Li et al., 2025; Zhang et al., 2026). UDT uses a learned token-level uncertainty signal for causal information routing and detached loss reweighting, specifically targeting TS-RT.

3 Problem Formulation

We consider a finite-horizon offline reinforcement learning setting with horizon T and discount factor $\gamma \in [0, 1]$. The offline dataset consists of N trajectories,

$$\mathcal{D} = \{\tau_i\}_{i=1}^{N}, \qquad \tau_i = \left\{ \left(s_t^{(i)}, a_t^{(i)}, r_t^{(i)} \right) \right\}_{t=0}^{T_i - 1}, \tag{1}$$

where $T_i \leq T$, and $s_t^{(i)}$, $a_t^{(i)}$, and $r_t^{(i)}$ denote the observation, action, and reward at timestep t of trajectory i, respectively.

We denote the corrupted component by

$$c \in \{\mathrm{obs}, \mathrm{act}, \mathrm{rew}\},$$

with obs, act, and rew corresponding to observation, action, and reward corruption, respectively.

Temporally Independent Training Corruption. Let $\mathcal{P}_m^c$ denote a training-time corruption process applied to component c, where $m \in \{\mathrm{rand}, \mathrm{adv}\}$ indexes the corruption type, with rand and adv denoting random and adversarial corruption, respectively.

Let $\xi_{m,t}^c$ represent timestep-specific randomness and let x_t^c denote the value of component c at timestep t, with $x_t^{\mathrm{obs}} = s_t$, $x_t^{\mathrm{act}} = a_t$, and $x_t^{\mathrm{rew}} = r_t$. We define the pointwise action of $\mathcal{P}_m^c$ as

$$\mathcal{P}_m^c : \left(x_t^c, \xi_{m,t}^c\right) \mapsto \widetilde{x}_{m,t}^c, \qquad \delta_{m,t}^c = \Psi_m^c\left(x_t^c, \xi_{m,t}^c\right), \qquad \widetilde{x}_{m,t}^c = x_t^c + \delta_{m,t}^c. \tag{2}$$

Here, $\delta_{m,t}^c$ denotes the training-time perturbation, and Ψ_m^c specifies the perturbation mapping for corruption type m on component c, such as random noise or a timestep-wise adversarial perturbation. The corruption rule does not explicitly depend on a persistent corruption state or preceding perturbations. We refer to this pointwise structure as **temporally independent training corruption**.

Applying $\mathcal{P}_m^c$ pointwise to the offline dataset produces the complete corrupted dataset, from which an offline RL algorithm Alg learns

$$\widetilde{\mathcal{D}}_{m,c} = \mathcal{P}_m^c(\mathcal{D}), \qquad \pi_{m,c} = \mathsf{Alg}\left(\widetilde{\mathcal{D}}_{m,c}\right). \tag{3}$$

Here, $\pi_{m,c}$ denotes the policy associated with corruption condition (m, c).

Temporally Correlated Test-Time Corruption. Let $\mathcal{Q}^c$ denote a test-time corruption process applied to component c. This process is independent of the training corruption type m and is expressed by the following mapping:

$$\mathcal{Q}^c : \left(x_t^c, z_{t-1}^c, \omega_t^c\right) \mapsto \left(\widetilde{x}_t^c, z_t^c\right), \quad z_t^c = F^c\left(z_{t-1}^c, \omega_t^c\right), \quad \eta_t^c = G^c(x_t^c, z_t^c), \quad \widetilde{x}_t^c = x_t^c + \eta_t^c, \tag{4}$$

where z_t^c denotes the internal state of the corruption process, ω_t^c denotes the random input at timestep t, F^c denotes the state-transition mapping, and G^c denotes the perturbation mapping.

Following prior work on temporally coupled perturbations (Liang et al., 2024), we characterize temporal dependence through the persistent corruption state z_t^c. Since z_t^c explicitly depends on z_{t-1}^c, perturbation information can persist across timesteps. We refer to this explicit cross-timestep dependence as **temporally correlated test-time corruption**.

Temporal-Structure Robustness Transfer. We define **temporal-structure robustness transfer (TS-RT)** as the evaluation setting in which a policy learned from a complete offline dataset under temporally independent training corruption $\mathcal{P}_m^c$ is evaluated under a previously unseen, temporally correlated test-time corruption process $\mathcal{Q}^c$ applied to the same component c.

The resulting TS-RT protocol is

$$\mathcal{D} \xrightarrow{\mathcal{P}_m^c} \widetilde{\mathcal{D}}_{m,c} \xrightarrow{\mathsf{Alg}} \pi_{m,c} \xrightarrow[\text{test time}]{\mathcal{Q}^c} J_{\mathrm{TS}}^{m,c}. \tag{5}$$

In Section 4, we instantiate Alg with our proposed Uncertainty-Guided Decision Transformer.

TS-RT performance is defined as the expected return, computed using the uncorrupted environment rewards, when the policy is evaluated under $\mathcal{Q}^c$:

$$J_{\mathrm{TS}}^{m,c} = \mathbb{E}_{\pi_{m,c},\, \mathcal{Q}^c}\left[\sum_{t=0}^{T-1} \gamma^t r_t^{\mathrm{env}}\right], \tag{6}$$

where the expectation is over the policy, environment dynamics, and test-time corruption process, and r_t^{env} denotes the uncorrupted environment reward used for task evaluation.

4 Uncertainty-Guided Decision Transformer

We propose **Uncertainty-Guided Decision Transformer (UDT)**, which introduces a learned token-level uncertainty signal to regulate both causal information routing and the influence of unreliable targets during training. UDT is instantiated on the Robust Decision Transformer (RDT) backbone (Xu et al., 2025) and retains RDT's embedding dropout, reward prediction, Gaussian error-based weighting, and iterative data correction. It replaces deterministic token projections with paired content and uncertainty projections. UDT further introduces uncertainty-dependent attention modulation and reliability-aware loss weighting, reusing the same uncertainty signal in both mechanisms.

Under TS-RT, unreliable information may persist in the causal context and influence subsequent decisions. Training-time reweighting and data correction can mitigate corrupted supervision during optimization, but do not directly regulate how historical information is used at inference time. UDT therefore uses token-level uncertainty to modulate the attention logits of historical key tokens and to reweight action and reward prediction losses during training, with a stop-gradient applied to the weighting path. Importantly, UDT is trained only on the corrupted offline dataset $\widetilde{\mathcal{D}}_{m,c}$ and has no access to the test-time corruption process $\mathcal{Q}^c$ during policy optimization. The overall framework is illustrated in Figure 2.

4.1 Uncertainty-Aware Token Representation

UDT follows the return–observation–action tokenization of the underlying sequence model. For each timestep t, we consider token types

$$z \in \{\mathrm{ret}, \mathrm{obs}, \mathrm{act}\},$$

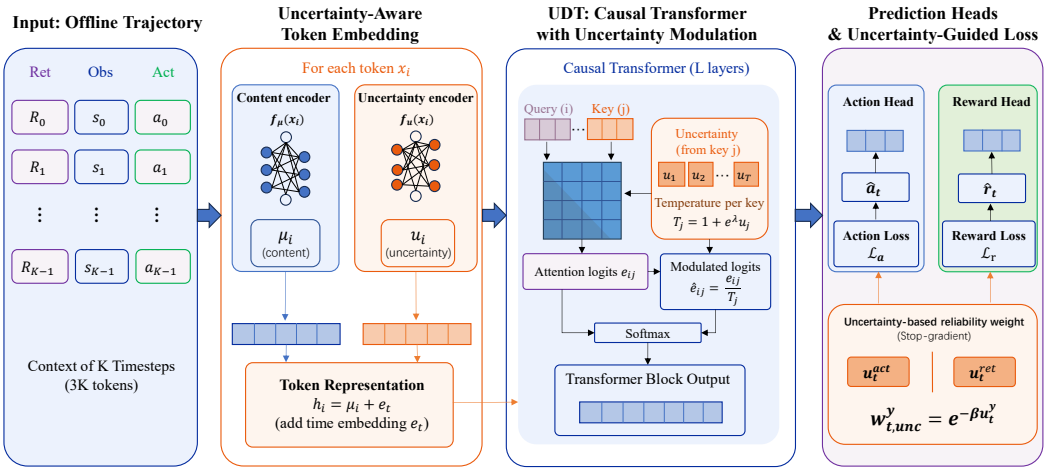

Figure 2: Overview of the **Uncertainty-Guided Decision Transformer (UDT)**. Each trajectory token is mapped to a content representation μ_i and a token-level uncertainty score u_i. The uncertainty score modulates causal attention and provides detached weights for the action and reward prediction losses.

where x_t^{ret}, x_t^{obs}, and x_t^{act} denote the return-to-go, observation, and action inputs, respectively. Instead of mapping each token through a single deterministic projection, UDT employs two spectrally normalized projection networks (Miyato et al., 2018),

$$\mu_t^z = f_\mu^z(x_t^z), \qquad \ell_t^z = f_u^z(x_t^z), \tag{7}$$

where $\mu_t^z, \ell_t^z \in \mathbb{R}^d$ and d denotes the embedding dimension.

The content branch produces μ_t^z, which is used as the token representation, while the uncertainty branch produces an uncertainty parameter vector ℓ_t^z. We transform the latter into a positive token-level uncertainty score,

$$u_t^z = \frac{1}{d}\sum_{k=1}^{d} \exp\left(\ell_{t,k}^z\right), \tag{8}$$

where $\ell_{t,k}^z$ denotes the k-th dimension of ℓ_t^z. We interpret larger u_t^z as lower estimated reliability.

A timestep embedding e_t is added to form $h_t^z = \mu_t^z + e_t$, while u_t^z is computed solely from the token value. For a context window of K timesteps, each timestep contributes one return-to-go, observation, and action token, yielding an interleaved sequence of $3K$ tokens. We use h_i and u_i, $i \in \{1, \ldots, 3K\}$, to denote the representation and uncertainty score of the i-th token, respectively.

4.2 Uncertainty-Guided Causal Attention

A corrupted token may remain in the context window and influence multiple subsequent decisions. UDT therefore uses the token-level uncertainty scores defined above to modulate the attention logits associated with historical key tokens.

For a given attention head, let $q_i, k_j \in \mathbb{R}^{d_k}$ denote the query of token i and the key of token j, respectively, where d_k denotes the dimensionality of the queries and keys. UDT modifies standard scaled dot-product attention (Vaswani et al., 2017) as

$$e_{ij} = \frac{q_i^\top k_j}{\sqrt{d_k}}, \qquad T_j = 1 + \eta u_j,\ \eta = \exp(\lambda) > 0, \qquad A_{ij} = \text{softmax}_j\left(\frac{e_{ij}}{T_j} + M_{ij}\right). \tag{9}$$

Here, λ controls the strength of uncertainty modulation, T_j is the resulting key-dependent temperature, and M_{ij} is the additive causal-padding mask.

Since $T_j \geq 1$, increasing u_j contracts the corresponding attention logit toward zero before softmax normalization. The uncertainty-based logit scaling provides continuous, token-dependent modulation of causal attention without hard removal of historical tokens. Because the scaling remains

differentiable with respect to u_j, gradients from the action and reward prediction objective propagate through the attention pathway to the uncertainty branch.

4.3 Reliability-Aware Training Objective

In addition to modulating causal attention, UDT reuses the learned uncertainty signal to reduce the training influence of potentially unreliable action and reward targets.

Following the Gaussian-weighted learning scheme of RDT (Xu et al., 2025), let $y \in \{a, r\}$ index the action and reward prediction targets, where the target associated with y has dimension d_y. We define the per-sample prediction error and the corresponding error-based weight as

$$\delta_t^y = \frac{1}{d_y} \|\widehat{y}_t - y_t\|_2^2, \qquad w_{t,\mathrm{err}}^y = \exp(-\beta_y \delta_t^y), \tag{10}$$

where $\beta_y \geq 0$ controls the strength of error-based down-weighting.

UDT further introduces an uncertainty-based reliability weight,

$$w_{t,\mathrm{unc}}^y = \exp(-\beta_u u_t^y), \qquad u_t^a = u_t^{\mathrm{act}}, \qquad u_t^r = u_t^{\mathrm{ret}}, \tag{11}$$

where $\beta_u > 0$ controls the sensitivity of reliability weighting to token uncertainty. Action prediction uses the uncertainty of the corresponding action token. Since return-to-go is constructed from the reward sequence, reward corruption directly affects the corresponding return tokens. We therefore use the uncertainty score u_t^{ret} of the return-to-go token as a proxy for the reliability of reward supervision.

The error-based and uncertainty-based weights are combined using stop-gradient,

$$\bar{w}_t^y = \mathrm{sg}\left(w_{t,\mathrm{err}}^y w_{t,\mathrm{unc}}^y\right), \tag{12}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The stop-gradient blocks gradients from the reliability weighting term to the uncertainty branch. Gradients instead reach the uncertainty branch through the differentiable attention pathway.

The weighted prediction loss over a context of K timesteps is

$$\mathcal{L}_y = \frac{1}{K} \sum_{t=1}^{K} v_t \, \bar{w}_t^y \, \delta_t^y, \tag{13}$$

where $v_t \in \{0, 1\}$ is the validity mask, with $v_t = 0$ for padded positions.

For $y = a$ and $y = r$, this gives the action loss $\mathcal{L}_a$ and reward loss $\mathcal{L}_r$, respectively.

The UDT prediction objective is

$$\mathcal{L}_{\mathrm{UDT}} = \mathcal{L}_a + \lambda_r \mathcal{L}_r, \tag{14}$$

where λ_r controls the contribution of reward prediction. Algorithm 1 in Appendix B summarizes the complete training procedure of UDT.

5 Drift-Attack

We instantiate the test-time corruption process $\mathcal{Q}^c$ in Eq. 5 as **Drift-Attack**. For each evaluation condition, test-time perturbations target the same component corrupted during training. The temporal structure of corruption changes from temporally independent to temporally correlated. The same mechanism is used for observation, action, and reward corruption.

For each component $c \in \{\mathrm{obs}, \mathrm{act}, \mathrm{rew}\}$, a binary corruption state $m_t^c \in \{0, 1\}$ follows a two-state Markov process:

$$\Pr(m_{t+1}^c = 1 \mid m_t^c = 0) = p_{\mathrm{burst}}, \qquad \Pr(m_{t+1}^c = 0 \mid m_t^c = 1) = p_{\mathrm{recover}}, \tag{15}$$

where p_{burst} and p_{recover} denote the activation and recovery probabilities, respectively.

For scale calibration, let x_{ref}^c denote s, a, and r^{env} for $c = \mathrm{obs}$, act, and rew, respectively. We define

$$\sigma_c = \mathrm{Std}_{\mathcal{D}}(x_{\mathrm{ref}}^c). \tag{16}$$

When corruption is active, the target perturbation is

$$b_t^c = \epsilon\, \sigma_c \odot v_t^c, \tag{17}$$

where ϵ controls the perturbation magnitude, and $v_t^c = \mathrm{sign}(s_t)$, $\mathrm{sign}(a_t)$, and $-\mathrm{sign}(r_t^{\mathrm{env}})$ for observation, action, and reward corruption, respectively.

To introduce temporal persistence, we apply exponential smoothing (Wawrzyński, 2015; Korenkevych et al., 2019):

$$d_t^c = \begin{cases} \mathrm{clip}\big(\alpha d_{t-1}^c + (1-\alpha) b_t^c, -3\sigma_c, 3\sigma_c\big), & m_t^c = 1, \\ 0, & m_t^c = 0, \end{cases} \tag{18}$$

where $\alpha \in [0, 1)$ controls temporal persistence. In the notation of Section 3, d_t^c corresponds to the test-time perturbation η_t^c generated by $\mathcal{Q}^c$.

The resulting perturbation is applied to the corresponding component:

$$\widetilde{s}_t = s_t + d_t^{\mathrm{obs}}, \qquad \widetilde{a}_t = \mathrm{clip}\left(a_t + d_t^{\mathrm{act}}, a_{\min}, a_{\max}\right), \qquad \widetilde{r}_t = r_t^{\mathrm{env}} + d_t^{\mathrm{rew}}. \tag{19}$$

Together, Eqs. 15–19 instantiate the test-time corruption process $\mathcal{Q}^c$ in Eq. 5. For reward corruption, the perturbed reward also updates the subsequent return-to-go:

$$\widetilde{R}_{t+1} = \widetilde{R}_t - \widetilde{r}_t. \tag{20}$$

The original environment reward r_t^{env} is retained for performance evaluation.

6 Experiments

We empirically evaluate the robustness-transfer behavior of existing offline RL methods and the effectiveness of UDT under TS-RT by asking three questions: **(1)** How do existing offline RL methods perform under TS-RT? **(2)** Does UDT consistently improve robustness transfer across training corruption types, corrupted components, and tasks? **(3)** How do its two uncertainty-guided mechanisms contribute to robustness transfer and robustness to persistent context contamination?

6.1 Evaluation Protocol

Benchmarks and baselines. We evaluate on nine D4RL tasks spanning MuJoCo, Kitchen, and Adroit (Fu et al., 2020), following the limited-data setting of RDT (Xu et al., 2025). We compare UDT with seven representative offline RL and corruption-robust baselines: BC, CQL, UWMSG, RIQL, DeFog, DT, and RDT (Kumar et al., 2020; Ye et al., 2023; Yang et al., 2024b; Hu et al., 2023; Chen et al., 2021; Xu et al., 2025).

TS-RT protocol. Following RDT (Xu et al., 2025), policies are trained with random or adversarial temporally independent corruption on observations, actions, or rewards, using a corruption ratio of 30% and a corruption scale of $\epsilon = 1.0$. At test time, each policy is evaluated under Drift-Attack on the same corrupted component, where the test-time perturbations introduce temporal correlation absent from the training corruption.

We report D4RL normalized scores computed from the original environment return. To summarize performance across the TS-RT benchmark, we compute category-wise averages by training corruption type, corrupted component, and task, together with an overall macro-average across all evaluation conditions. All results are averaged over three independent training seeds. Additional experimental details are provided in Appendix A.

6.2 Main Results under TS-RT

Table 1 reports the category-wise aggregate scores under TS-RT. We group the results by training corruption type, corrupted component, and task. All results are reported as mean and standard deviation over three independent training seeds. Full results for all evaluation conditions are provided in Appendix C.

Table 1: Normalized scores under TS-RT, reported as mean ± standard deviation over three training seeds. Rand. and Adv. denote random and adversarial training corruption, respectively. Best results are shown in bold according to the unrounded means.

Metric	BC	CQL	UWMSG	RIQL	DeFog	DT	RDT	UDT
Training corruption								
Rand.	40.2±1.0	4.5±0.2	5.0±0.5	41.6±2.4	49.5±1.0	61.8±1.4	59.9±1.0	**65.5±1.1**
Adv.	33.5±1.8	5.0±0.2	7.0±0.6	36.1±0.5	43.3±0.7	55.2±0.1	56.9±0.6	**61.6±0.8**
Corrupted component								
Obs.	35.9±1.3	3.0±0.2	3.5±0.5	33.5±1.7	47.4±0.9	**60.2±1.4**	51.0±1.1	56.6±2.7
Act.	24.3±1.7	4.3±0.5	7.7±0.9	34.7±2.3	40.7±0.5	45.6±0.5	54.1±0.4	**59.3±1.9**
Rew.	50.2±2.6	6.9±0.3	7.0±0.4	48.4±0.3	51.1±2.2	69.8±1.5	70.1±1.8	**74.7±1.0**
MuJoCo								
HalfCheetah	3.4±0.2	23.9±1.0	5.9±0.6	6.6±0.5	14.9±1.2	18.4±1.1	19.7±1.5	**27.3±0.3**
Hopper	23.3±0.7	2.7±0.7	35.6±2.2	30.0±2.8	39.2±5.1	45.9±2.5	47.7±1.9	**53.6±1.7**
Walker2d	13.5±0.3	5.1±0.2	11.3±0.8	11.6±0.9	22.0±2.7	36.5±0.8	39.6±0.9	**45.7±2.3**
Kitchen								
Complete	20.4±5.0	8.8±1.2	0.0±0.0	35.6±1.7	44.4±3.2	**60.7±0.9**	43.3±2.1	44.4±2.5
Mixed	30.7±1.5	0.4±0.0	0.0±0.0	27.5±3.7	12.9±2.1	38.5±0.7	34.7±1.9	**41.4±4.8**
Partial	32.8±1.4	2.1±1.7	1.4±2.0	29.4±2.9	13.3±1.6	36.5±2.9	34.7±5.1	**51.9±2.4**
Adroit								
Door	75.5±1.6	-0.3±0.0	-0.2±0.0	85.4±3.2	104.6±0.3	103.3±1.6	**104.6±0.2**	104.2±1.0
Hammer	88.8±1.8	0.3±0.0	0.5±0.1	97.7±6.1	116.0±2.5	116.5±1.7	120.5±3.8	**124.9±3.4**
Relocate	43.0±4.2	-0.2±0.0	-0.3±0.0	25.8±3.1	50.1±3.1	70.3±3.0	**80.6±1.0**	78.3±3.1
Overall								
Overall	36.8±1.3	4.7±0.1	6.0±0.5	38.8±1.3	46.4±0.8	58.5±0.7	58.4±0.5	**63.5±0.6**

Overall TS-RT performance. UDT achieves the highest overall normalized score of 63.5, outperforming DT (58.5) and RDT (58.4) by 8.5% and 8.8%, respectively. Across the 15 reported averages, UDT achieves the highest score in 11. It ranks first under both training corruption types, two of the three corrupted components, six of the nine tasks, and in the overall average. The advantage is particularly pronounced under adversarial training corruption, where UDT reaches 61.6, exceeding RDT (56.9) by 8.3%. These results show that UDT achieves the strongest overall TS-RT performance across both random and adversarial training corruption settings.

Component- and task-level performance. UDT attains the highest aggregate performance under both action and reward corruption, with scores of 59.3 and 74.7, outperforming RDT by 9.6% and 6.6%, respectively. At the task level, UDT ranks first on all three MuJoCo tasks, as well as Kitchen-Mixed, Kitchen-Partial, and Hammer. The largest task-level margin is observed on Kitchen-Partial, where UDT outperforms the strongest baseline by 42.2%, followed by a 15.4% gain on Walker2d. The remaining cases reveal complementary strengths of the baselines: DT performs best under observation corruption and on Kitchen-Complete, where UDT is approximately 5.9% and 26.8% lower, respectively, while RDT leads on Door and Relocate, where UDT is approximately 0.4% and 2.8% lower, respectively. Overall, UDT's advantages span multiple corrupted components and task families, rather than being concentrated in a small subset of the benchmark.

6.3 Ablation Study

We evaluate two ablated variants of UDT: **w/o Reweighting**, which removes reliability-aware loss reweighting, and **w/o Routing**, which removes uncertainty-guided causal attention. All other settings remain unchanged. We report averages across training corruption types, corrupted components, and the overall score. The complete results for seeds 0, 1, and 2 are provided in Appendix D.

The full UDT achieves the best overall TS-RT performance. Removing reweighting causes the largest drop under observation corruption, whereas removing routing most strongly degrades reward corruption. At the same time, individual ablations can perform marginally better on specific components. Nevertheless, combining both mechanisms yields the strongest aggregate robustness, indicating complementary rather than uniformly monotonic contributions.

Table 2: Ablation of uncertainty-guided causal attention and reliability-aware loss reweighting under TS-RT. Results are mean ± standard deviation over three seeds, with the best results in bold.

Setting	Rand.	Adv.	Obs.	Act.	Rew.	Overall
UDT	**65.5±1.1**	**61.6±0.8**	56.6±2.7	**59.3±1.9**	**74.7±1.0**	**63.5±0.6**
w/o Reweighting	64.0±0.9	60.9±1.7	55.3±3.3	58.2±1.4	73.9±2.2	62.4±0.4
w/o Routing	64.0±0.4	60.5±0.3	**56.8±0.7**	58.3±0.6	71.7±1.7	62.3±0.3

6.4 Persistent Context Contamination Analysis

We further examine how persistent corruption affects sequence models by contaminating their historical context. Using clean trajectories, we inject contiguous observation-corruption bursts of length $L \in \{1, 5, 10, 15\}$ with fixed per-token corruption strength and measure the post-burst action deviation $D_m(L)$. To quantify how residual effects grow with corruption persistence, we report the normalized residual growth

$$\bar{D}_m(L) = \frac{D_m(L)}{D_m(1)}. \tag{21}$$

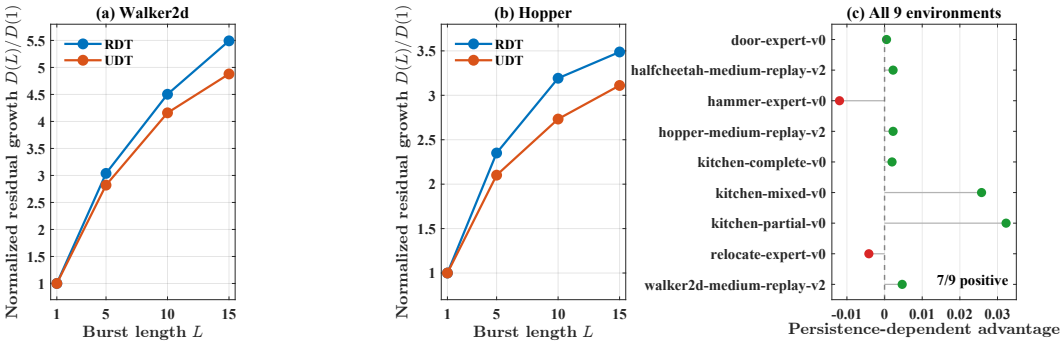

Figure 3: Persistent context contamination analysis. Panels (a)–(b) show normalized post-burst residual action deviation as corruption burst length increases. Panel (c) summarizes the persistence-dependent advantage of UDT over RDT across all nine tasks, with positive values indicating a larger UDT advantage at longer corruption bursts.

As shown in Figure 3, residual action deviation grows with corruption persistence for both RDT and UDT, but more slowly for UDT. UDT shows a persistence-dependent advantage in 7 of 9 tasks, indicating lower sensitivity than RDT to prolonged causal-context contamination. This is consistent with the ablation results, suggesting that uncertainty-guided routing reduces the influence of unreliable historical information.

7 Conclusion

We provide the first systematic study of **temporal-structure robustness transfer (TS-RT)** in offline reinforcement learning, a setting in which policies learned under temporally independent corruption are evaluated under previously unseen, temporally correlated test-time corruption across observations, actions, and rewards. To address TS-RT, we propose **Uncertainty-Guided Decision Transformer (UDT)**, which uses learned token-level uncertainty to regulate both causal information routing and unreliable supervision. We further introduce **Drift-Attack** for controlled evaluation of temporal-structure shifts. Across nine D4RL tasks, UDT achieves the strongest aggregate TS-RT performance, while ablation and persistent-context analyses support the effectiveness of its uncertainty-guided mechanisms. These findings highlight uncertainty-guided sequence modeling as an effective approach to improving robustness transfer under temporal-structure shifts in offline RL.

AI USE STATEMENT

In this work, we used generative AI tools to assist with feedback on research methodology and experimental design, the formulation and refinement of mathematical descriptions, interpretation of experimental results, and implementation-related code review and debugging. We additionally used generative AI tools for literature discovery and organization, manuscript drafting and language editing, paper structuring, and the organization and presentation of figures and experimental results.

We did not use generative AI tools to generate synthetic datasets or formal proofs, and other required-disclosure tasks not mentioned above were not applicable to this work. All AI-assisted outputs were reviewed and verified by the authors, including checking related-work claims against original sources and validating code and experimental results. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the robustness of offline reinforcement learning under data corruption and test-time perturbations. All experiments are conducted in standard simulated control environments and do not involve human subjects or private user data. Drift-Attack is introduced solely for robustness evaluation and for understanding policy behavior under temporally correlated corruption. We do not evaluate attacks on real-world physical systems.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate reproduction of our results. The problem setting and evaluation protocol are formally defined in Sections 3 and 5, while the architecture and training objective of UDT are described in Section 4. Experimental settings, including datasets, training corruption, model and optimization hyperparameters, Drift-Attack parameters, training seeds, and evaluation procedures, are provided in Appendix A. Full task-level results and ablation results are also reported in the appendix. The source code is provided as supplementary material, and the trained checkpoints are available in an anonymous repository.

REFERENCES

Johannes Ackermann, Takayuki Osa, and Masashi Sugiyama. Offline reinforcement learning from datasets with structured non-stationarity. *Reinforcement Learning Journal*, 5:2140–2161, 2024.

Suzan Ece Ada, Georg Martius, Emre Ugur, and Erhan Oztop. Forecasting in offline reinforcement learning for non-stationary environments. In *Advances in Neural Information Processing Systems*, volume 38, pp. 8910–8932, 2025. doi: 10.52202/085713-0273.

Shingo Ayabe, Takuto Otomo, Hiroshi Kera, and Kazuhiko Kawamoto. Robustness evaluation of offline reinforcement learning for robot control against action perturbations. *International Journal of Advanced Robotic Systems*, 22(4), 2025. doi: 10.1177/17298806251360454.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision Transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097, 2021.

Yiding Chen, Xuezhou Zhang, Qiaomin Xie, and Xiaojin Zhu. Exact policy recovery in offline RL with both heavy-tailed rewards and data corruption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11416–11424, 2024. doi: 10.1609/aaai.v38i10.29022.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Provable RL with exogenous distractors via multistep inverse dynamics. In *International Conference on Learning Representations*, 2022.

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, 2019.

Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. Uncertainty-aware attention for reliable interpretation and prediction. In *Advances in Neural Information Processing Systems*, volume 31, pp. 917–926, 2018.

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

Dmytro Korenkevych, A. Rupam Mahmood, Gautham Vasan, and James Bergstra. Autoregressive policies for continuous control deep reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 2754–2762, 2019. doi: 10.24963/ijcai.2019/382.

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. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Xian Yeow Lee, Sambit Ghadai, Kai Liang Tan, Chinmay Hegde, and Soumik Sarkar. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4577–4584, 2020. doi: 10.1609/aaai.v34i04.5887.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Zenan Li, Fan Nie, Qiao Sun, Fang Da, and Hang Zhao. Uncertainty-aware Decision Transformer for stochastic driving environments. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 364–386, 2025.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, Xiangyu Liu, Benjamin Eysenbach, Tuomas Sandholm, Furong Huang, and Stephen McAleer. Game-theoretic robust reinforcement learning handles temporally-coupled perturbations. In *International Conference on Learning Representations*, 2024.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5443.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Zhongjian Qiao, Rui Yang, Jiafei Lyu, Xiu Li, Zhongxiang Dai, Zhuoran Yang, Siyang Gao, and Shuang Qiu. Dual-robust cross-domain offline reinforcement learning against dynamics shifts. In *International Conference on Learning Representations*, 2026.

Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, pp. 20877–20890, 2022.

Xiaohang Tang, Afonso Marques, Parameswaran Kamalaruban, and Ilija Bogunovic. Adversarially robust Decision Transformer. In *Advances in Neural Information Processing Systems*, volume 37, pp. 29697–29722, 2024. doi: 10.52202/079017-0934.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kevin Vogt-Lowell, Theodoros Tsiligkaridis, Rodney Lafuente-Mercado, Surabhi Ghatti, Shanghua Gao, Marinka Zitnik, and Daniela Rus. When sensors fail: Temporal sequence models for robust PPO under sensor drift. In *International Conference on Learning Representations*, 2026.

Paweł Wawrzyński. Control policy with autocorrelated noise in reinforcement learning for robotics. *International Journal of Machine Learning and Computing*, 5(2):91–95, 2015. doi: 10.7763/IJMLC.2015.V5.489.

Fan Wu, Linyi Li, Chejian Xu, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. COPA: Certifying robust policies for offline reinforcement learning against poisoning attacks. In *International Conference on Learning Representations*, 2022.

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 *International Conference on Learning Representations*, 2025.

Rui Yang, Jie Wang, Guoping Wu, and Bin Li. Uncertainty-based offline variational bayesian reinforcement learning for robustness under diverse data corruptions. In *Advances in Neural Information Processing Systems*, volume 37, pp. 39748–39783, 2024a. doi: 10.52202/079017-1255.

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*, 2024b.

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*, 2024.

Chenlu Ye, Rui Yang, Quanquan Gu, and Tong Zhang. Corruption-robust offline reinforcement learning with general function approximation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1570.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 5757–5773, 2022.

Zhihao Zhang, Chengyang Peng, Minghao Zhu, Ekim Yurtsever, and Keith A. Redmill. An uncertainty-weighted Decision Transformer for navigation in dense, complex driving scenarios. In *IEEE Intelligent Vehicles Symposium*, pp. 23–30, 2026. doi: 10.1109/IV66570.2026.11624079.

Yibo Zhao, Zijian Cao, Zhihao Zhang, and Yanfang Fu. ORBIT: Robust offline reinforcement learning via integrated observation recovery with belief-based decision-making under observation corruption and delay. *Electronics*, 15(14):3214, 2026. doi: 10.3390/electronics15143214.

A Experimental Details

A.1 Datasets and Training Corruption

We follow the limited-data setting of RDT (Xu et al., 2025). We evaluate on nine D4RL tasks spanning MuJoCo, Kitchen, and Adroit. The dataset configurations are summarized in Table 3. Observations are normalized using statistics from the corresponding training dataset.

Table 3: Dataset and evaluation configurations.

Task	Dataset	Ratio	Target returns	Reward scale
HalfCheetah	halfcheetah-medium-replay-v2	10%	12000, 6000	10^{-3}
Hopper	hopper-medium-replay-v2	10%	3600, 1800	10^{-3}
Walker2d	walker2d-medium-replay-v2	10%	5000, 2500	10^{-3}
Kitchen-Complete	kitchen-complete-v0	100%	400, 500	1
Kitchen-Mixed	kitchen-mixed-v0	100%	400, 500	1
Kitchen-Partial	kitchen-partial-v0	100%	400, 500	1
Door	door-expert-v0	1%	2900, 1450	1
Hammer	hammer-expert-v0	1%	12800, 6400	1
Relocate	relocate-expert-v0	1%	4300, 2150	1

Following RDT (Xu et al., 2025), training corruption is applied pointwise over time to observations, actions, or rewards, without a persistent corruption state. We use a corruption ratio of 30% and a corruption scale of $\epsilon = 1.0$.

A.2 Training Details

The common model and optimization settings of UDT are listed in Table 4.

Table 4: Model and optimization hyperparameters.

Hyperparameter	Value
Transformer blocks	3
Attention heads	1
Embedding dimension	128
Context length	20
Batch size	64
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Gradient clipping	0.25
Attention dropout	0.0
Residual dropout	0.1
Training epochs	100
Updates per epoch	1000

For the UDT-specific reliability weighting, we set $\beta_u = 2.0$ for all tasks. The learnable modulation parameter λ is initialized to 0, corresponding to $\exp(\lambda) = 1$ at initialization, and is optimized jointly with the model parameters. UDT retains the robust training mechanisms of RDT. These include embedding dropout, Gaussian error-based weighting, and iterative data correction. Task-specific settings are summarized in Table 5.

Iterative data correction starts after epoch 50 when enabled.

A.3 Drift-Attack Parameters

The parameters used for Drift-Attack are listed in Table 6.

The clean-reference standard deviation σ_c is used for scale calibration, as defined in Section 5. After perturbation, actions are clipped to the valid environment range.

Table 5: Task-specific robust training hyperparameters. p denotes embedding dropout and ζ the data-correction threshold. "–" indicates no iterative data correction.

Task group	Comp.	p	(β_a, β_r)	ζ	λ_r
MuJoCo	Obs.	0.2	$(1.0, 1.0)$	–	1.0
	Act.	0.2	$(1.0, 1.0)$	6.0	1.0
	Rew.	0.2	$(1.0, 1.0)$	6.0	1.0
Kitchen	Obs.	0.1	$(30.0, 30.0)$	–	0.1
	Act.	0.1	$(30.0, 30.0)$	5.0	0.1
	Rew.	0.1	$(30.0, 30.0)$	6.0	0.1
Adroit	Obs.	0.1	$(0.1, 0.0)$	–	1.0
	Act.	0.1	$(10.0, 0.0)$	6.0	1.0
	Rew.	0.1	$(0.1, 0.1)$	6.0	1.0

Table 6: Drift-Attack parameters.

Parameter	Value
p_{burst}	0.1
p_{recover}	0.3
α	0.85
ϵ	1.0
Drift clipping	$[-3\sigma_c, 3\sigma_c]$

A.4 Evaluation

Each policy is evaluated for 10 episodes for each target return in Table 3. We report the higher mean normalized score across the two target returns. Performance is computed from the original environment return.

For reward corruption, the corrupted reward affects the return-to-go supplied to the policy. It is not used to compute the evaluation score.

The main results and ablation study use three training seeds, 0, 1, and 2. We report the mean and standard deviation across seeds.

Aggregate results are computed by training corruption type, corrupted component, and task. We also report an overall macro-average across all 54 evaluation conditions. Full results are provided in Appendix C.

A.5 Persistent Context Contamination Analysis

For the analysis in Section 6.4, we use clean trajectories and inject contiguous observation-corruption bursts of length $L \in \{1, 5, 10, 15\}$ with a fixed per-token corruption strength. For each model m, we measure the post-burst action deviation $D_m(L)$ and report the normalized residual growth $\bar{D}_m(L) = D_m(L)/D_m(1)$. The analysis compares RDT and UDT across all nine tasks.

B UDT TRAINING PROCEDURE

Algorithm 1 summarizes the training procedure of Uncertainty-Guided Decision Transformer (UDT).

Algorithm 1 Training Procedure of Uncertainty-Guided Decision Transformer

Require: Corrupted offline dataset $\widetilde{\mathcal{D}}_{m,c}$, context length K, coefficients $\beta_a, \beta_r, \beta_u, \lambda_r$
Ensure: Trained policy $\pi_{m,c}$

Initialize UDT parameters θ and modulation parameter $\lambda \leftarrow 0$
for each training iteration **do**
  Sample a batch of K-step trajectory segments from $\widetilde{\mathcal{D}}_{m,c}$
  Let $v_t \in \{0, 1\}$ denote the validity mask for non-padded positions
  for $z \in \{\mathrm{ret}, \mathrm{obs}, \mathrm{act}\}$ **do**
    $\mu_t^z \leftarrow f_\mu^z(x_t^z)$, $\ell_t^z \leftarrow f_u^z(x_t^z)$
    $u_t^z \leftarrow \frac{1}{d}\sum_{k=1}^{d} \exp(\ell_{t,k}^z), \quad h_t^z \leftarrow \mu_t^z + e_t$
  end for
  Interleave token representations and uncertainty scores
  Apply embedding normalization and dropout
  for each causal Transformer layer **do**
    $e_{ij} \leftarrow q_i^\top k_j / \sqrt{d_k}, \quad T_j \leftarrow 1 + \exp(\lambda) u_j$
    $A_{ij} \leftarrow \mathrm{softmax}_j(e_{ij}/T_j + M_{ij})$
  end for
  Predict actions $\widehat{a}_t$ and rewards $\widehat{r}_t$
  Set $u_t^a \leftarrow u_t^{\mathrm{act}}$ and $u_t^r \leftarrow u_t^{\mathrm{ret}}$
  for $y \in \{a, r\}$ **do**
    $\delta_t^y \leftarrow \frac{1}{d_y}\|\widehat{y}_t - y_t\|_2^2$
    $w_{t,\mathrm{err}}^y \leftarrow \exp(-\beta_y \delta_t^y), \quad w_{t,\mathrm{unc}}^y \leftarrow \exp(-\beta_u u_t^y)$
    $\bar{w}_t^y \leftarrow \mathrm{sg}\big(w_{t,\mathrm{err}}^y w_{t,\mathrm{unc}}^y\big)$
    $\mathcal{L}_y \leftarrow \frac{1}{K}\sum_{t=1}^{K} v_t \, \bar{w}_t^y \, \delta_t^y$
  end for
  $\mathcal{L}_{\mathrm{UDT}} \leftarrow \mathcal{L}_a + \lambda_r \mathcal{L}_r$
  Update θ and λ using $\nabla \mathcal{L}_{\mathrm{UDT}}$
  Apply iterative data correction following RDT when scheduled
end for
return $\pi_{m,c}$

C DETAILED TS-RT RESULTS

C.1 MAIN BASELINE COMPARISON

Tables 7 and 8 report the detailed TS-RT results under random and adversarial training corruption, respectively. For each task and corrupted component, we report the mean and standard deviation over three independent training seeds. The component-level averages are computed across the nine tasks, and the final row reports the macro-average over all 27 evaluation conditions. Best results among the compared methods are shown in bold according to the unrounded means.

Table 7: Detailed TS-RT normalized scores under random training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	BC	CQL	UWMSG	RIQL	DeFog	DT	RDT	UDT
Observation corruption								
HalfCheetah (10%)	3.1±0.7	10.4±0.5	4.5±0.7	6.2±1.4	12.1±1.2	13.1±0.8	12.5±0.8	**15.7±0.9**

Continued on next page

Table 7 – continued from previous page

Task	BC	CQL	UWMSG	RIQL	DeFog	DT	RDT	UDT
Hopper (10%)	18.1±1.7	7.4±4.0	22.8±8.1	18.9±3.8	35.2±1.7	43.4±1.9	45.4±1.3	**49.2±5.0**
Walker2d (10%)	9.5±1.4	0.1±0.2	-0.6±0.2	11.1±2.0	19.8±2.9	29.0±0.9	36.1±1.6	**37.8±6.3**
Kitchen-Complete	12.5±3.5	9.2±3.1	0.0±0.0	16.7±3.1	60.0±10.8	**61.7±4.2**	25.8±8.2	23.3±1.4
Kitchen-Partial	20.0±2.0	0.0±0.0	0.0±0.0	8.3±2.4	17.5±7.4	51.7±6.2	14.2±5.1	**60.0±5.0**
Kitchen-Mixed	23.3±3.1	0.0±0.0	0.0±0.0	8.3±3.1	23.3±4.2	43.3±1.2	16.7±4.2	**50.0±10.0**
Door (1%)	94.2±5.2	-0.3±0.0	0.0±0.1	93.7±9.2	104.4±0.9	104.8±0.5	**104.9±0.5**	104.2±1.0
Hammer (1%)	104.7±7.7	0.2±0.0	0.1±0.1	104.9±9.7	125.6±0.9	119.6±2.6	122.5±6.6	**127.4±0.9**
Relocate (1%)	30.0±3.6	-0.3±0.0	-0.3±0.0	28.2±2.1	35.3±1.4	**76.4±6.5**	75.1±4.6	68.7±6.3
Average	35.0	3.0	3.0	32.9	48.1	**60.3**	50.4	59.6
Action corruption								
HalfCheetah (10%)	1.8±0.4	21.6±4.8	11.3±5.0	4.0±0.8	16.1±1.0	19.0±2.2	22.9±1.7	**32.6±1.5**
Hopper (10%)	27.1±2.1	1.4±0.6	42.1±2.0	26.8±2.2	37.6±0.6	44.5±2.6	44.4±2.5	**53.3±5.1**
Walker2d (10%)	11.5±3.6	3.6±3.0	13.2±2.7	13.8±1.1	28.1±4.1	33.0±0.7	45.8±2.2	**50.9±8.1**
Kitchen-Complete	5.8±1.2	15.8±1.2	0.0±0.0	35.8±4.2	24.2±5.1	40.0±7.4	**45.8±2.4**	31.7±6.3
Kitchen-Partial	40.8±5.1	4.2±5.9	0.0±0.0	42.5±3.5	28.3±3.1	39.2±4.2	41.7±6.2	**48.3±5.2**
Kitchen-Mixed	39.2±1.2	0.8±1.2	0.0±0.0	**42.5±7.4**	25.0±7.1	35.8±1.2	37.5±5.4	36.7±1.4
Door (1%)	76.6±7.7	-0.3±0.0	-0.2±0.1	81.7±3.0	**105.6±0.5**	103.7±1.5	105.2±0.1	101.7±6.2
Hammer (1%)	74.2±5.9	0.3±0.0	0.5±0.5	85.3±20.1	**124.0±1.8**	112.9±2.5	117.0±12.1	117.0±16.4
Relocate (1%)	40.5±8.0	-0.3±0.0	-0.3±0.0	34.9±6.5	56.2±2.7	69.8±1.6	75.3±5.5	**77.9±1.7**
Average	35.3	5.2	7.4	40.8	49.4	55.3	59.5	**61.1**
Reward corruption								
HalfCheetah (10%)	5.8±0.5	35.6±1.4	3.4±0.4	8.9±0.6	17.9±1.1	22.6±2.1	25.5±1.4	**37.0±0.5**
Hopper (10%)	29.0±2.5	1.0±0.6	26.7±4.5	42.6±2.7	53.1±13.6	49.9±0.8	47.2±3.8	**59.5±0.3**
Walker2d (10%)	20.3±2.2	11.5±4.6	12.8±4.1	11.4±4.2	28.5±5.0	48.6±0.6	48.7±2.9	**59.1±11.4**
Kitchen-Complete	38.3±15.0	0.0±0.0	0.0±0.0	61.7±3.1	53.3±15.5	**96.7±2.4**	71.7±5.1	85.0±17.3
Kitchen-Partial	45.0±0.0	0.0±0.0	0.0±0.0	**67.5±0.0**	5.0±4.1	37.5±13.4	54.2±12.0	63.3±9.5
Kitchen-Mixed	41.7±5.1	0.0±0.0	0.0±0.0	42.5±8.2	3.3±2.4	49.2±1.2	**55.8±4.2**	50.0±2.5
Door (1%)	97.9±3.9	-0.3±0.0	-0.2±0.1	96.4±2.6	104.6±0.2	105.1±0.2	104.9±0.3	**105.5±0.1**
Hammer (1%)	106.3±5.0	0.2±0.0	0.2±0.1	106.0±7.0	125.5±0.2	125.7±0.8	**126.9±0.2**	126.3±1.8
Relocate (1%)	67.5±8.0	-0.2±0.1	-0.3±0.0	22.8±2.5	67.2±5.0	92.5±9.7	94.4±6.7	**95.6±6.4**
Average	50.2	5.3	4.7	51.1	50.9	69.8	69.9	**75.7**
Average over all tasks	40.2	4.5	5.0	41.6	49.5	61.8	59.9	**65.5**

Table 8: Detailed TS-RT normalized scores under adversarial training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	BC	CQL	UWMSG	RIQL	DeFog	DT	RDT	UDT
Observation corruption								
HalfCheetah (10%)	3.6±0.2	**17.3±0.9**	8.0±1.8	5.0±1.0	10.5±0.5	13.3±1.2	11.2±2.4	13.3±0.9
Hopper (10%)	17.6±2.3	1.8±0.5	22.3±1.8	19.6±0.7	32.9±3.0	43.5±3.7	42.5±4.7	**50.1±0.4**
Walker2d (10%)	13.7±3.0	1.2±1.4	5.8±2.4	10.8±1.2	18.5±0.6	37.6±0.7	30.8±1.8	**38.5±0.6**
Kitchen-Complete	20.0±2.0	5.8±3.1	0.0±0.0	20.0±2.0	55.8±3.1	**57.5±5.4**	26.7±5.9	19.2±3.8
Kitchen-Partial	30.8±4.2	0.0±0.0	0.0±0.0	13.3±3.1	25.0±2.0	**39.2±4.2**	18.3±2.4	30.0±10.9
Kitchen-Mixed	25.8±2.4	1.7±1.2	0.0±0.0	19.2±4.2	15.0±10.8	**40.8±4.2**	17.5±6.1	17.5±10.0
Door (1%)	78.3±5.1	-0.2±0.1	-0.2±0.0	87.2±4.3	104.3±0.6	104.8±0.2	**105.2±0.0**	104.8±0.1
Hammer (1%)	99.3±10.1	0.3±0.0	0.2±0.0	102.9±5.5	119.0±4.9	126.0±0.7	127.1±0.5	**127.1±1.4**
Relocate (1%)	42.3±12.0	-0.2±0.0	-0.3±0.0	27.8±6.0	38.5±8.0	78.2±4.2	**86.0±5.4**	82.6±9.1
Average	36.8	3.1	4.0	34.0	46.6	**60.1**	51.7	53.7
Action corruption								
HalfCheetah (10%)	0.6±0.3	22.4±3.4	6.3±1.1	1.4±0.6	15.4±3.0	20.0±0.9	19.2±1.7	**30.1±1.3**
Hopper (10%)	19.0±3.8	1.8±0.0	49.2±10.7	28.4±7.7	32.3±2.9	42.3±5.3	**50.5±2.5**	50.5±2.7
Walker2d (10%)	5.4±0.1	3.3±0.7	6.3±0.2	5.9±1.0	12.3±1.1	22.0±3.6	22.0±6.2	**35.8±3.1**
Kitchen-Complete	7.5±3.5	2.5±3.5	0.0±0.0	20.0±4.1	15.0±3.5	17.5±3.5	18.3±3.1	**24.2±3.8**
Kitchen-Partial	15.0±2.0	0.0±0.0	8.3±11.8	0.0±0.0	0.8±1.2	14.2±5.1	26.7±3.1	**51.7±6.3**
Kitchen-Mixed	12.5±5.4	0.0±0.0	0.0±0.0	5.0±2.0	5.0±0.0	5.0±0.0	31.7±1.2	**41.7±3.8**
Door (1%)	8.3±4.4	-0.2±0.0	-0.2±0.0	81.5±9.3	**104.2±0.2**	96.0±8.8	103.1±0.5	103.4±0.5
Hammer (1%)	42.0±19.4	0.3±0.0	2.0±1.1	100.2±5.2	76.7±10.0	88.8±8.6	102.0±17.0	**124.2±0.4**
Relocate (1%)	10.3±1.6	-0.2±0.0	-0.3±0.0	14.4±1.9	26.2±6.4	16.4±5.7	**64.4±6.7**	55.2±11.3
Average	13.4	3.3	8.0	28.5	32.0	35.8	48.6	**57.4**
Reward corruption								
HalfCheetah (10%)	5.8±0.5	**36.1±1.5**	2.0±0.3	14.1±1.2	17.4±1.8	22.5±1.4	27.1±4.1	34.8±0.6
Hopper (10%)	29.0±2.5	2.4±0.4	50.8±7.4	43.9±6.8	43.8±12.5	52.1±2.3	56.1±0.8	**58.8±7.5**
Walker2d (10%)	20.3±2.2	10.7±5.2	30.2±4.1	16.4±1.8	24.8±7.3	48.6±0.7	**54.4±2.9**	52.5±3.5
Kitchen-Complete	38.3±15.0	19.2±11.2	0.0±0.0	59.2±8.2	58.3±3.1	**90.8±5.9**	71.7±4.7	83.3±2.9
Kitchen-Partial	45.0±0.0	8.3±11.8	0.0±0.0	45.0±10.8	3.3±3.1	37.5±6.1	53.3±17.0	**58.3±8.8**
Kitchen-Mixed	41.7±5.1	0.0±0.0	0.0±0.0	47.5±2.0	5.8±5.1	**56.7±2.4**	49.2±1.2	52.5±12.5
Door (1%)	97.9±3.9	-0.3±0.0	-0.2±0.0	72.0±21.6	104.8±0.1	105.3±0.1	104.6±0.3	**105.5±0.4**
Hammer (1%)	106.3±5.0	0.2±0.0	0.2±0.1	86.8±12.5	125.4±1.0	125.8±0.7	**127.3±0.4**	127.2±0.8
Relocate (1%)	67.5±8.0	-0.3±0.1	-0.3±0.0	26.5±2.0	77.5±7.0	88.8±2.9	88.2±3.3	**89.6±1.7**
Average	50.2	8.5	9.2	45.7	51.3	69.8	70.2	**73.6**
Average over all tasks	33.5	5.0	7.0	36.1	43.3	55.2	56.9	**61.6**

C.2 Additional Comparison with RBC

RBC is additionally included as a reference baseline following RDT (Xu et al., 2025). To provide a direct comparison under the TS-RT protocol, we report the mean and standard deviation over the same three training seeds for RBC and UDT under both random and adversarial training corruption. For the component-level and overall rows, we first compute the corresponding macro-average for each seed and then report the mean and standard deviation across the three seed-level averages. The higher mean in each comparison is shown in bold.

Table 9: Comparison between RBC and UDT under random training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	RBC	UDT
Observation corruption		
HalfCheetah (10%)	3.3±0.5	**15.7±0.9**
Hopper (10%)	16.3±1.8	**49.2±5.0**
Walker2d (10%)	9.3±2.1	**37.8±6.3**
Kitchen-Complete	13.3±5.1	**23.3±1.4**
Kitchen-Partial	27.5±3.5	**60.0±5.0**
Kitchen-Mixed	27.5±2.0	**50.0±10.0**
Door (1%)	93.6±7.3	**104.2±1.0**
Hammer (1%)	105.5±2.3	**127.4±0.9**
Relocate (1%)	26.2±8.2	**68.7±6.3**
Average	35.8±1.2	**59.6±2.6**
Action corruption		
HalfCheetah (10%)	1.3±0.3	**32.6±1.5**
Hopper (10%)	23.8±1.8	**53.3±5.1**
Walker2d (10%)	10.8±2.6	**50.9±8.1**
Kitchen-Complete	10.8±6.2	**31.7±6.3**
Kitchen-Partial	38.3±4.7	**48.3±5.2**
Kitchen-Mixed	**46.7±6.2**	36.7±1.4
Door (1%)	82.5±12.7	**101.7±6.2**
Hammer (1%)	91.2±15.8	**117.0±16.4**
Relocate (1%)	38.5±6.0	**77.9±1.7**
Average	38.2±2.9	**61.1±1.6**
Reward corruption		
HalfCheetah (10%)	4.0±0.7	**37.0±0.5**
Hopper (10%)	32.7±8.7	**59.5±0.3**
Walker2d (10%)	16.2±0.8	**59.1±11.4**
Kitchen-Complete	51.7±18.5	**85.0±17.3**
Kitchen-Partial	41.7±4.2	**63.3±9.5**
Kitchen-Mixed	45.0±2.0	**50.0±2.5**
Door (1%)	103.1±0.9	**105.5±0.1**
Hammer (1%)	110.9±10.8	**126.3±1.8**
Relocate (1%)	66.5±17.9	**95.6±6.4**
Average	52.4±5.8	**75.7±1.1**
Average over all tasks	42.1±3.3	**65.5±1.1**

Table 10: Comparison between RBC and UDT under adversarial training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	RBC	UDT
Observation corruption		
HalfCheetah (10%)	4.1±0.2	**13.3±0.9**
Hopper (10%)	19.6±1.6	**50.1±0.4**
Walker2d (10%)	11.4±2.6	**38.5±0.6**
Kitchen-Complete	15.8±1.2	**19.2±3.8**
Kitchen-Partial	**30.0±2.0**	30.0±10.9
Kitchen-Mixed	**25.8±4.7**	17.5±10.0
Door (1%)	70.8±7.3	**104.8±0.1**
Hammer (1%)	104.7±1.6	**127.1±1.4**
Relocate (1%)	42.0±8.0	**82.6±9.1**
Average	36.0±1.3	**53.7±2.9**
Action corruption		
HalfCheetah (10%)	0.8±0.2	**30.1±1.3**
Hopper (10%)	15.4±2.5	**50.5±2.7**
Walker2d (10%)	3.7±0.4	**35.8±3.1**
Kitchen-Complete	9.2±4.7	**24.2±3.8**
Kitchen-Partial	14.2±2.4	**51.7±6.3**
Kitchen-Mixed	30.0±4.1	**41.7±3.8**
Door (1%)	16.8±8.1	**103.4±0.5**
Hammer (1%)	66.9±5.5	**124.2±0.4**
Relocate (1%)	12.5±4.3	**55.2±11.3**
Average	18.8±2.2	**57.4±2.9**
Reward corruption		
HalfCheetah (10%)	4.0±0.7	**34.8±0.6**
Hopper (10%)	32.7±8.7	**58.8±7.5**
Walker2d (10%)	16.2±0.8	**52.5±3.5**
Kitchen-Complete	51.7±18.5	**83.3±2.9**
Kitchen-Partial	41.7±4.2	**58.3±8.8**
Kitchen-Mixed	45.0±2.0	**52.5±12.5**
Door (1%)	103.1±0.9	**105.5±0.4**
Hammer (1%)	110.9±10.8	**127.2±0.8**
Relocate (1%)	66.5±17.9	**89.6±1.7**
Average	52.4±5.8	**73.6±2.1**
Average over all tasks	35.7±2.0	**61.6±0.8**

Comparison with RBC. At the aggregate level, UDT consistently outperforms RBC across all corrupted components under both training corruption types.

D Full Ablation Results

Tables 11 and 12 report the complete per-condition ablation results under random and adversarial training corruption, respectively. Each entry is reported as the mean ± standard deviation over training seeds 0, 1, and 2. Each table contains 27 evaluation conditions formed by nine tasks and three corrupted components. The best mean in each condition is shown in bold.

Table 11: Complete per-condition ablation results under random training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	Comp.	UDT	w/o Reweighting	w/o Routing
HalfCheetah	Obs.	15.70±0.91	**18.35±3.66**	13.16±2.44
	Act.	32.59±1.46	**33.52±1.29**	30.06±1.39
	Rew.	37.01±0.46	**37.33±1.17**	35.66±1.23
Hopper	Obs.	**49.25±5.04**	48.04±2.98	44.97±3.12
	Act.	**53.26±5.07**	48.45±3.39	50.33±1.08
	Rew.	**59.50±0.32**	54.06±6.25	57.02±2.58
Walker2d	Obs.	**37.78±6.27**	33.08±2.87	32.40±3.62
	Act.	**50.89±8.11**	47.94±4.79	47.32±6.72
	Rew.	**59.05±11.38**	57.46±2.23	57.39±5.45
Complete	Obs.	23.33±1.44	**25.00±9.01**	15.83±12.33
	Act.	31.67±6.29	33.33±2.89	**36.67±3.82**
	Rew.	85.00±17.32	**85.83±1.44**	85.00±4.33
Mixed	Obs.	50.00±10.00	43.33±19.09	**55.00±7.50**
	Act.	36.67±1.44	37.50±2.50	**38.33±3.82**
	Rew.	50.00±2.50	**58.33±6.29**	53.33±8.04
Partial	Obs.	**60.00±5.00**	55.00±2.50	55.83±3.82
	Act.	48.33±5.20	**50.83±3.82**	45.00±2.50
	Rew.	**63.33±9.46**	53.33±7.64	37.50±6.61
Door	Obs.	104.25±1.00	104.32±0.31	**105.11±0.07**
	Act.	101.69±6.23	104.58±0.99	**105.17±1.41**
	Rew.	105.55±0.08	104.16±0.50	**105.60±0.23**
Hammer	Obs.	**127.37±0.95**	124.60±4.79	126.93±0.61
	Act.	117.02±16.40	124.40±2.60	**124.94±1.41**
	Rew.	126.28±1.80	**126.94±0.61**	126.64±0.72
Relocate	Obs.	68.75±6.29	63.16±3.87	**79.58±7.15**
	Act.	**77.91±1.72**	68.16±1.61	76.10±4.03
	Rew.	**95.59±6.43**	86.33±5.25	86.78±7.60

Table 12: Complete per-condition ablation results under adversarial training corruption. Results are reported as mean ± standard deviation over three training seeds.

Task	Comp.	UDT	w/o Reweighting	w/o Routing
HalfCheetah	Obs.	**13.31±0.89**	11.44±2.57	12.79±0.68
	Act.	**30.11±1.29**	29.14±2.31	27.40±0.16
	Rew.	34.80±0.58	35.43±3.00	**36.27±0.97**
Hopper	Obs.	50.12±0.44	48.58±1.82	**50.79±1.94**
	Act.	50.50±2.68	49.72±4.69	**51.12±3.79**
	Rew.	**58.83±7.55**	57.30±0.96	54.35±3.18
Walker2d	Obs.	38.46±0.60	34.83±6.73	**39.21±5.40**
	Act.	**35.79±3.15**	32.53±2.93	29.76±1.68
	Rew.	**52.52±3.45**	49.61±3.04	49.89±1.24
Complete	Obs.	19.17±3.82	**25.83±1.44**	20.00±10.90
	Act.	**24.17±3.82**	23.33±6.29	22.50±5.00
	Rew.	83.33±2.89	**84.17±8.78**	83.33±5.20
Mixed	Obs.	17.50±10.00	**24.17±11.81**	20.00±10.00
	Act.	**41.67±3.82**	40.00±4.33	36.67±3.82
	Rew.	52.50±12.50	**60.83±8.78**	56.67±16.65
Partial	Obs.	30.00±10.90	30.83±20.05	**35.00±5.00**
	Act.	**51.67±6.29**	50.83±6.29	50.00±4.33
	Rew.	**58.33±8.78**	52.50±11.46	43.33±7.64
Door	Obs.	**104.81±0.15**	104.39±0.39	101.40±6.16
	Act.	**103.36±0.46**	102.13±0.87	102.35±1.24
	Rew.	**105.49±0.37**	105.25±0.61	105.05±0.67
Hammer	Obs.	**127.10±1.36**	126.91±2.24	126.99±0.77
	Act.	**124.16±0.38**	123.41±0.94	118.40±6.49
	Rew.	**127.18±0.76**	126.77±1.21	126.43±0.07
Relocate	Obs.	82.57±9.06	72.71±5.42	**86.61±14.02**
	Act.	55.20±11.26	47.90±7.94	**57.40±7.31**
	Rew.	89.61±1.70	**93.75±8.92**	90.95±4.42