GENERATIVE SUPPORT REALIGNMENT FOR CROSS-DOMAIN OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-domain offline reinforcement learning learns a target policy from pre-collected source and target datasets with different dynamics. When target data are scarce, effectively compensating for their limited coverage using source data remains challenging due to the discrepancy between domains. We propose *Target-aligned Coverage Expansion (TCE)*, which leverages source states to generatively realign and expand the limited target support while controlling the generation error induced by this expansion. We further derive a performance gap bound that characterizes the interplay between generation error and source–target dynamics gap, providing theoretical guidance for effective coverage expansion and source utilization. Across diverse cross-domain environments, TCE consistently outperforms state-of-the-art baselines, while further analysis empirically supports the guidance provided by our theoretical findings.

1 INTRODUCTION

Cross-domain reinforcement learning (cross-domain RL) aims to transfer a policy learned in a source domain to a target domain with different environment dynamics. This setting arises in real-world applications such as heterogeneous robot control, simulation-to-real autonomous driving, and medical decision making (Gottesman et al., 2018; Yurtsever et al., 2020). A variety of RL approaches have been proposed to address such domain shifts (Eysenbach et al., 2020; Kim et al., 2020), but most assume online interaction with the source or target environment to collect additional data during training. In practice, such interaction can be costly, unsafe, or even infeasible (Levine et al., 2020). Cross-domain offline RL addresses this limitation by learning exclusively from pre-collected datasets from both domains (Liu et al., 2022). This problem becomes particularly challenging when target data are scarce, making effective use of source data crucial for learning in the target domain.

Existing cross-domain offline RL methods primarily exploit source data in two ways. One line of work selects source transitions similar to the target domain through contrastive representation learning (Wen et al., 2024). The resulting dataset can then be used to train policies with standard offline RL algorithms such as CQL (Kumar et al., 2020) or IQL (Kostrikov et al., 2022). Recent methods further consider smaller target-data budgets by selectively incorporating target-near source transitions using optimal transport (Lyu et al., 2025). Another line of work generates additional training data by leveraging source data (Le Pham Van et al., 2025; Guo et al., 2026). While these approaches improve the use of source information, effectively compensating for scarce target data remains challenging due to source–target dynamics gap and generation error.

This raises a central question: *how can source data best compensate for limited target coverage while keeping the resulting performance gap small?* To address this question, we derive a theoretical performance bound that disentangles the dynamics gap from source transitions and the generation error from target-aligned generation. Based on this analysis, we propose **Target-aligned Coverage Expansion (TCE)**, which aims to expand the limited target coverage while keeping the resulting performance gap small. To this end, TCE exploits the broader state coverage of source data to expand the support of target-aligned generation while controlling the error induced by this expansion, and additionally incorporates target-near source transitions when favorable under the bound. Extensive experiments across diverse domain shifts demonstrate consistent improvements over existing methods, and further analysis supports the theoretical implications and identifies when direct source incorporation provides additional benefits.

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.

2 Related Work

Offline Reinforcement Learning. Offline RL learns policies entirely from pre-collected datasets without additional environment interaction. Representative methods such as CQL (Kumar et al., 2020) and IQL (Kostrikov et al., 2022) achieve strong performance on standard offline RL benchmarks (Fu et al., 2020), while subsequent studies further improve offline learning through advanced penalty and constraint mechanisms (Lyu et al., 2022; Yeom et al., 2024). Other approaches address distributional shift through behavior regularization (Wu et al., 2019) or data filtering based on mutual information (Poole et al., 2019; Guo et al., 2022). More recently, generative approaches have been explored to augment limited offline datasets (Li et al., 2024; Luo et al., 2025; Chae et al., 2026). However, learning from data collected under different environment dynamics introduces an additional challenge, motivating cross-domain offline RL.

Cross-Domain Reinforcement Learning. Cross-domain RL addresses substantial dynamics shifts across distinct domains, beyond the relatively small task variations considered in meta-learning (Finn et al., 2017; Kim et al., 2025; Han et al., 2026), using approaches such as reward correction (Eysenbach et al., 2020) and value-guided data filtering (Xu et al., 2023). Cross-domain imitation learning similarly transfers demonstrations across domains (Kim et al., 2020; Fickinger et al., 2022; Choi et al., 2023), including adaptation to varying environment dynamics (Chae et al., 2022). However, online approaches require target interaction (Lyu et al., 2024a), limiting their applicability to purely offline settings. Offline cross-domain RL (Wen et al., 2024) has therefore explored source selection (Lyu et al., 2025), transition editing (Niu et al., 2024), robust adaptation (Qiao et al., 2025), and data generation (Le Pham Van et al., 2025; Guo et al., 2026). While these methods primarily mitigate dynamics mismatch or augment available data, TCE uses source states to expand limited target support and generates target-aligned transitions over the expanded support, controlling the expansion to limit the resulting performance gap.

Score-Based Models and Diffusion Processes. Two approaches to score-based generative modeling have enabled high-quality sample generation: denoising score matching, which estimates gradients of the data log-density across multiple noise scales (Song & Ermon, 2019; 2020), and diffusion models, which progressively corrupt and then denoise data through a sequence of intermediate steps (Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021). The stochastic differential equations (SDEs) framework unifies both approaches and enables continuous-time sampling (Song et al., 2020; Karras et al., 2022). In this paper, we leverage the SDE formalism for two complementary generation processes: expanding state coverage beyond the limited target support and generating target-aligned state pairs over the expanded states.

3 Background

3.1 Cross-Domain Offline RL Setup

We define a Markov Decision Process (MDP) as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P_{\mathcal{M}}, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $P_{\mathcal{M}}$ the transition dynamics, R the reward function, and γ the discount factor. In the cross-domain setting, we assume access to a source domain $\mathcal{M}_{\mathrm{src}} = (\mathcal{S}, \mathcal{A}, P_{\mathrm{src}}, R, \gamma)$ and a target domain $\mathcal{M}_{\mathrm{tar}} = (\mathcal{S}, \mathcal{A}, P_{\mathrm{tar}}, R, \gamma)$, which share the same state and action spaces as well as the reward function but differ in their transition dynamics, i.e., $P_{\mathrm{src}} \neq P_{\mathrm{tar}}$. In the cross-domain offline setting, the agent cannot interact with either domain and must rely solely on pre-collected transitions (s_t, a_t, r_t, s_{t+1}), where $s_t \in \mathcal{S}$ denotes the state, $a_t \in \mathcal{A}$ the action, $r_t = R(s_t, a_t)$ the reward, and $s_{t+1} \sim P(\cdot|s_t, a_t)$ the next state with $P = P_{\mathrm{src}}$ or $P = P_{\mathrm{tar}}$. We denote the datasets collected from the source and target domains as $\mathcal{D}_{\mathrm{src}}$ and $\mathcal{D}_{\mathrm{tar}}$, respectively, under the practical constraint that $|\mathcal{D}_{\mathrm{tar}}| \ll |\mathcal{D}_{\mathrm{src}}|$, making direct policy learning on the target domain challenging.

3.2 Score-based Generative Models with SDEs

Generative models aim to learn the data distribution $p_{\text{data}}(x)$ and generate realistic samples. Representative approaches include Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), Variational Auto-Encoders (VAEs) (Kingma & Welling, 2013), and diffusion models (Ho et al., 2020). Among these, score-based generative models formulated with stochastic differential equations (SDEs) (Song et al., 2020) provide a continuous-time view of diffusion, support flexible noise schedules, and enable efficient sampling and likelihood computation via the probability-flow ODE. Given a clean sample x^0, we perturb it with Gaussian noise as $x^\tau = x^0 + \sigma(\tau)z$, where $z \sim \mathcal{N}(0, I)$, $\tau \in [0, 1]$ denotes the continuous noise level, and $\sigma(\tau)$ is the noise scale. In this paper, we consider a conditional

input c and train a score network $q_\theta(x, \tau \mid c)$ to estimate the conditional score $\nabla_x \log p_\tau(x \mid c)$ via denoising score matching:

$$\mathcal{L}_{\text{score}}(\theta) = \mathbb{E}_{\tau,(x^0,c)}\left[\zeta(\tau) \, \| \, q_\theta(x^\tau, \tau \mid c) + z/\sigma(\tau)\|_2^2\right], \tag{1}$$

where $\zeta(\tau)$ is a time-dependent weight. At sampling time, starting from $x^1 \sim \mathcal{N}(0, \sigma(1)^2 I)$, samples are generated by solving the discretized reverse SDE using the Predictor–Corrector sampler with discretized noise levels τ^k ($1 = \tau^K > \cdots > \tau^0 = 0$):

$$x^{k-1} = x^k + \left[f(x^k, \tau^k) - g(\tau^k)^2 q_\theta(x^k, \tau^k \mid c)\right]\Delta\tau^k + g(\tau^k)\sqrt{|\Delta\tau^k|}\,\xi^k, \quad \xi^k \sim \mathcal{N}(0, I). \tag{2}$$

where f and g denote the drift and diffusion coefficients, and $\Delta\tau^k = \tau^{k-1} - \tau^k$ is the step size. In this work, we additionally apply a Langevin corrector (Song & Ermon, 2019) after each predictor step to improve sample quality. When the conditional input c is absent, the formulation simplifies to an unconditional score network $q_\theta(x^\tau, \tau)$. Implementation details are provided in Appendix C.

4 METHODOLOGY

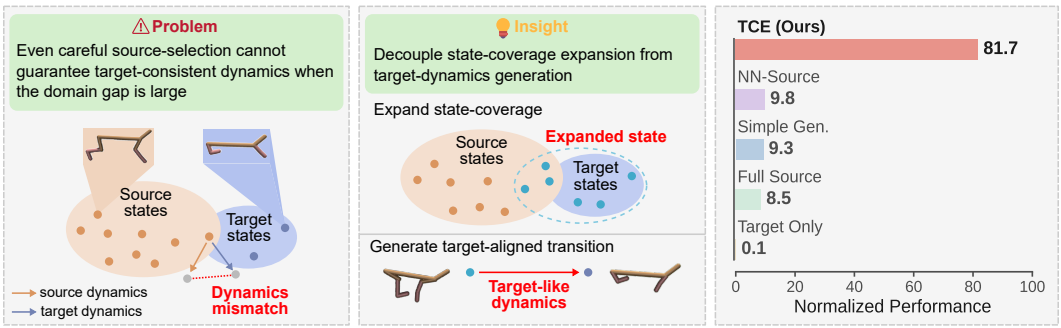


Figure 1: Motivation for target-aligned coverage expansion. (a) Source and target HalfCheetah domains with a large morphology gap. (b) TCE expands limited target-state coverage using source states and generates target-aligned transitions over the expanded support. (c) With coverage expansion, TCE (Ours) substantially outperforms target-only training (Target Only), full source reuse (Full Source), nearest-neighbor-based source selection (NN-Source), and simple generation (Simple Gen.).

4.1 MOTIVATION

A key challenge in cross-domain offline RL is to effectively exploit abundant source data when only limited target data are available. This challenge is particularly pronounced under large dynamics gaps, as illustrated by the substantial morphology difference between the source and target HalfCheetah domains in Fig. 1(a). Directly incorporating source transitions, even those close to the target data, can introduce considerable dynamics gap. In contrast, simple generation from scarce target data avoids source dynamics but may remain concentrated around the observed target support, providing limited improvement in state coverage. These limitations motivate using source data not simply as additional transitions, but to expand the limited target state support for generation.

Building on this idea, TCE expands the target support toward nearby source states and generates target-aligned transitions over the expanded regions, as illustrated in Fig. 1(b). The extent of expansion is controlled to increase coverage without excessively increasing the resulting performance gap. In this example, the source–target dynamics gap is sufficiently large that source transitions are excluded, while their states remain valuable for expanding the generation support. As shown in Fig. 1(c), this strategy yields a substantial performance improvement, whereas target-only learning, direct source reuse, NN-based selection of target-near source transitions, and simple generation provide only limited gains. This illustrates how source data can remain useful even under large dynamics gaps by expanding the support of target-aligned generation beyond that provided by scarce target data.

4.2 THEORETICAL ANALYSIS

To formalize the motivation above, we analyze the performance gap induced by two sources of transition discrepancy: directly incorporating source transitions and generating target-aligned transitions beyond the limited target data. This decomposition allows us to characterize how the

dynamics gap and generation error jointly affect target performance, providing a basis for expanding target coverage while controlling the resulting gap. For an MDP $\mathcal{M}$ and policy π, let $\rho^{\pi}_{\mathcal{M}}(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P^{\pi}_{\mathcal{M}}(s_t = s, a_t = a)$ denote the normalized discounted occupancy measure, where $P^{\pi}_{\mathcal{M}}$ is the visitation probability under π, and define the discounted return as $J_{\mathcal{M}}(\pi) := \frac{1}{1-\gamma}\mathbb{E}_{\rho^{\pi}_{\mathcal{M}}}[r(s,a)]$. We assume that the source and target MDPs share the state and action space, reward function, discount factor, and initial-state distribution, with $|r(s,a)| \le r_{\max}$.

Let P_{src}, P_{tar}, and $\hat{P}_{\mathrm{tar}}$ denote the source, target, and learned target-aligned transition kernels conditioned on (s,a), respectively. Define $P_{\mathrm{mix}} := \lambda P_{\mathrm{src}} + (1-\lambda)\hat{P}_{\mathrm{tar}}$, where $\lambda \in [0,1]$ is the mixture coefficient, and let $\mathcal{M}_{\mathrm{mix}}$ denote the induced MDP. Adapting the standard telescoping analysis of Xu et al. (2023) to our target-aligned generative setting, we obtain the following performance-gap bound.

Theorem 4.1. *Let $J_{\mathrm{mix}}(\pi)$ and $\rho^{\pi}_{\mathrm{mix}}$ denote the return and state-action occupancy measure under $\mathcal{M}_{\mathrm{mix}}$. Then,*

$$J_{\mathrm{mix}}(\pi) - J_{\mathrm{tar}}(\pi) \le \frac{2\gamma r_{\max}}{(1-\gamma)^2}\Big(\lambda\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\big[D_{\mathrm{TV}}(P_{\mathrm{src}} \,\|\, P_{\mathrm{tar}})\big] + (1-\lambda)\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\big[D_{\mathrm{TV}}(\hat{P}_{\mathrm{tar}} \,\|\, P_{\mathrm{tar}})\big]\Big). \quad (3)$$

where D_{TV} denotes total variation distance and the expectations are over $(s,a) \sim \rho^{\pi}_{\mathrm{mix}}$.

In addition, letting V^{π}_{tar} denote the value function of policy π in the target MDP, the performance gap can be bounded in terms of the corresponding value discrepancies as

$$J_{\mathrm{mix}}(\pi) - J_{\mathrm{tar}}(\pi) \le \frac{\gamma}{1-\gamma}\Big(\lambda\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}[|\mathbb{E}_{P_{\mathrm{src}}}[V^{\pi}_{\mathrm{tar}}(s')] - \mathbb{E}_{P_{\mathrm{tar}}}[V^{\pi}_{\mathrm{tar}}(s')]|] + (1-\lambda)\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\Big[\Big|\mathbb{E}_{\hat{P}_{\mathrm{tar}}}[V^{\pi}_{\mathrm{tar}}(s')] - \mathbb{E}_{P_{\mathrm{tar}}}[V^{\pi}_{\mathrm{tar}}(s')]\Big|\Big]\Big). \quad (4)$$

Proof. The proof of Theorem 4.1 is provided in Appendix B.

Theorem 4.1 decomposes the performance gap over the expanded occupancy $\rho^{\pi}_{\mathrm{mix}}$ into two terms: *the source–target dynamics gap* induced by P_{src} and *the generation error* induced by $\hat{P}_{\mathrm{tar}}$, weighted by λ and $1-\lambda$, respectively. Thus, effective coverage expansion requires not only broadening the target support, but also keeping the resulting performance gap small. This leads to two key implications for TCE. First, target-aligned generation should expand coverage as much as possible without inducing excessive generation error. Second, the source–target dynamics gap determines whether selected source transitions should be additionally incorporated. When this gap is small, target-near source transitions can complement the generated data; when it is large, their direct use becomes less favorable, and source data are instead used primarily for coverage expansion. These principles motivate the design of TCE presented next and are further examined across different dynamics gaps in our experiments.

4.3 Target-Aligned Generation over Expanded State Support

We first address the primary objective suggested by Theorem 4.1: expanding the limited target-state coverage without excessively increasing the generation-induced performance gap. TCE separates state-support expansion from target-transition generation using two score networks: a *coverage-expansion score network* $q^{\mathrm{cov}}_{\theta}$ for expanding the state distribution toward source-supported regions, and a *target-transition score network* $q^{\mathrm{tran}}_{\theta}$ for generating target-aligned state pairs (s,s') over the expanded support. The action and reward are subsequently inferred from each generated state pair rather than jointly generating the full transition (s,a,r,s') conditioned on s. As shown empirically in Section 5, this state-pair-first design provides better generalization under scarce target data.

To control the extent of coverage expansion, we introduce a coverage coefficient $\eta \in [0,1]$ and progressively incorporate source states according to their proximity to the target data. For a source transition $(s,s') \in \mathcal{D}_{\mathrm{src}}$, we define its distance to the target dataset as

$$d_{\mathrm{NN}}(s,s') := \min_{(\tilde{s},\tilde{s}') \in \mathcal{D}_{\mathrm{tar}}} \|s - \tilde{s}\| + \|s' - \tilde{s}'\|. \quad (5)$$

For $\eta \in (0,1]$, let d_{η} denote the η-quantile of these distances and define

$$\mathcal{D}^{\eta}_{\mathrm{src}} := \{(s,s') \in \mathcal{D}_{\mathrm{src}} : d_{\mathrm{NN}}(s,s') \le d_{\eta}\}. \quad (6)$$

When $\eta = 0$, only target states are used, whereas increasing η progressively expands the support toward source-covered regions, reaching the full source state support at $\eta = 1$. Accordingly, $q^{\mathrm{cov}}_{\theta}$ is

trained on states from $\mathcal{D}^{\eta}_{\mathrm{src}} \cup \mathcal{D}_{\mathrm{tar}}$ to model the expanded state distribution, while $q^{\mathrm{tran}}_{\theta}$ is trained exclusively on $\mathcal{D}_{\mathrm{tar}}$ to generate target-aligned state pairs over this expanded support.

Building on the score-matching objective in Eq. (1), we train the two score networks as

$$\underbrace{\mathbb{E}_{\tau,\, s_t \sim \mathcal{D}^{\eta}_{\mathrm{src}} \cup \mathcal{D}_{\mathrm{tar}}} \left[\zeta(\tau) \left\| q^{\mathrm{cov}}_{\theta}(s^{\tau}_t, \tau) + \tfrac{z}{\sigma(\tau)} \right\|_2^2 \right]}_{\textbf{coverage-expansion score loss}} + \underbrace{\mathbb{E}_{\tau,\, (s_t, s_{t+1}) \sim \mathcal{D}_{\mathrm{tar}}} \left[\zeta(\tau) \left\| q^{\mathrm{tran}}_{\theta}(s^{\tau}_{t+1}, \tau \mid s_t) + \tfrac{z}{\sigma(\tau)} \right\|_2^2 \right]}_{\textbf{target-transition score loss}}, \quad (7)$$

where $z \sim \mathcal{N}(0, I)$ and $s^{\tau} = s + \sigma(\tau) z$.

After training, we first sample a state from the expanded support by solving the reverse SDE with the coverage-expansion score network:

$$s^{k-1} = s^k + \left[f(s^k, \tau^k) - g(\tau^k)^2 q^{\mathrm{cov}}_{\theta}(s^k, \tau^k) \right] \Delta\tau^k + g(\tau^k) \sqrt{|\Delta\tau^k|}\, \xi^k, \quad \xi^k \sim \mathcal{N}(0, I), \quad (8)$$

which is integrated from $k = K$ to 0 to obtain $\hat{s}_t = s^0$. Conditioned on $\hat{s}_t$, we then solve the reverse SDE with the target-transition score network:

$$s^{k-1} = s^k + \left[f(s^k, \tau^k) - g(\tau^k)^2 q^{\mathrm{tran}}_{\theta}(s^k, \tau^k \mid \hat{s}_t) \right] \Delta\tau^k + g(\tau^k) \sqrt{|\Delta\tau^k|}\, \xi^k, \quad \xi^k \sim \mathcal{N}(0, I), \quad (9)$$

yielding a target-aligned state pair $(\hat{s}_t, \hat{s}_{t+1})$, where $\hat{s}_{t+1} = s^0$. We then generate the corresponding action and reward using separately trained inverse dynamics and reward models, Inv_{ψ} and $\hat{R}_{\phi}$, as $\hat{a}_t \sim \mathrm{Inv}_{\psi}(\cdot \mid \hat{s}_t, \hat{s}_{t+1})$ and $\hat{r}_t = \hat{R}_{\phi}(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1})$, respectively.

This procedure yields the *coverage-expanded, target-aligned dataset* $\mathcal{D}^{\eta}_{\mathrm{gen}} = \{(\hat{s}_t, \hat{a}_t, \hat{r}_t, \hat{s}_{t+1})\}$. The coefficient η directly controls its coverage: increasing η expands $\mathcal{D}^{\eta}_{\mathrm{gen}}$ toward broader source-supported regions, but may also increase the error of $\hat{P}_{\mathrm{tar}}$ farther from the observed target support. Thus, η governs the trade-off between coverage expansion and the generation-induced performance gap characterized in Theorem 4.1.

4.4 TCE Dataset Construction and Offline Learning

While $\mathcal{D}^{\eta}_{\mathrm{gen}}$ expands target coverage without directly inheriting source dynamics, target-near source transitions may also provide useful data for offline learning. Following the NN criterion in Eq. (6), we define $\mathcal{D}^{\lambda}_{\mathrm{src}}$ as the closest λ fraction of source transitions. Under a fixed augmentation budget, we construct the TCE training dataset as

$$\mathcal{D}^{\eta,\lambda}_{\mathrm{train}} = \mathcal{D}_{\mathrm{tar}} \cup \mathcal{D}^{\lambda}_{\mathrm{src}} \cup \mathcal{D}^{\eta}_{\mathrm{gen}},$$

where $|\mathcal{D}^{\lambda}_{\mathrm{src}}| = \lambda |\mathcal{D}_{\mathrm{src}}|$ and $|\mathcal{D}^{\eta}_{\mathrm{gen}}| = (1-\lambda)|\mathcal{D}_{\mathrm{src}}|$. Here, $\lambda \in [0, 1]$ corresponds to the mixture coefficient in Theorem 4.1 and controls the allocation between direct source utilization and target-aligned generation. A larger λ increases the contribution of source transitions and thus the influence of the source–target dynamics gap, whereas a smaller λ places greater emphasis on generated transitions and their generation error. The training dataset can therefore be adjusted according to the relative magnitudes of the dynamics gap and generation error.

Accordingly, we consider two variants of TCE, as illustrated in Fig. 2. **TCE(Gen)** uses only coverage-expanded target-aligned generation ($\lambda = 0, \eta > 0$), whereas **TCE(Mix)** additionally incorporates target-near source transitions ($\lambda > 0$, $\eta \geq 0$). TCE(Gen) is used under large source–target dynamics gaps, while TCE(Mix) allows direct source utilization when the dynamics gap of selected source transitions is sufficiently small. The setting $\eta = \lambda = 0$ corresponds to target-only generation without coverage expansion, which we denote as **Simple Augmentation (Simple Aug.)**.

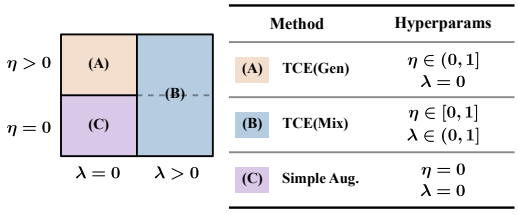


Figure 2: TCE configurations controlled by the coverage coefficient η and mixture coefficient λ.

Using $\mathcal{D}^{\eta,\lambda}_{\mathrm{train}}$, we perform offline policy learning with Implicit Q-Learning (IQL) (Kostrikov et al., 2022). Following Lyu et al. (2025), we additionally regularize the policy toward the target-domain behavior policy:

$$\mathcal{L}_{\pi} = \mathcal{L}^{\mathrm{IQL}}_{\pi} + \beta\, \mathbb{E}_{s \sim \mathcal{D}_{\mathrm{tar}}} \left[D_{\mathrm{KL}}(\hat{\pi}_{\mathrm{b}}(\cdot|s) \,\|\, \pi(\cdot|s)) \right], \quad (10)$$

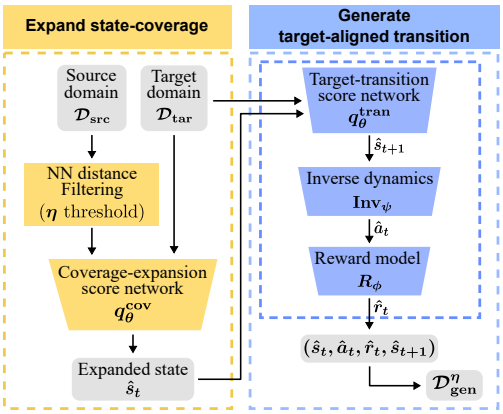


Figure 3: Overview of the TCE generation.

Algorithm 1 TCE Framework

Input: $\mathcal{D}_{\text{tar}}, \mathcal{D}_{\text{src}}, \eta, \lambda$
Train models: Train q_θ^{cov} on $\mathcal{D}_{\text{src}}^{\eta} \cup \mathcal{D}_{\text{tar}}$,
and q_θ^{tran} on $\mathcal{D}_{\text{tar}}$ via Eq. (7)
Train Inv_ψ on $\mathcal{D}_{\text{tar}}$, and $\hat{R}_\phi$ on $\mathcal{D}_{\text{src}} \cup \mathcal{D}_{\text{tar}}$
Generate samples:
Generate $\hat{s}_t \sim q_\theta^{\text{cov}}$, $\hat{s}_{t+1} \sim q_\theta^{\text{tran}}(\cdot \mid \hat{s}_t)$,
$\hat{a}_t \sim \text{Inv}_\psi$, and $\hat{r}_t = \hat{R}_\phi(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1})$ to form $\mathcal{D}_{\text{gen}}^{\eta}$, where $|\mathcal{D}_{\text{gen}}^{\eta}| = (1-\lambda)|\mathcal{D}_{\text{src}}|$
Construct training data:
Let $\mathcal{D}_{\text{src}}^{\lambda}$ be the $\lambda|\mathcal{D}_{\text{src}}|$ source transitions with the smallest d_{NN}
$\mathcal{D}_{\text{train}}^{\eta,\lambda} = \mathcal{D}_{\text{tar}} \cup \mathcal{D}_{\text{src}}^{\lambda} \cup \mathcal{D}_{\text{gen}}^{\eta}$
Offline RL: Train π on $\mathcal{D}_{\text{train}}^{\eta,\lambda}$

where $\hat{\pi}_{\text{b}}$ denotes the empirical behavior policy estimated from the target dataset. The overall TCE framework is illustrated in Fig. 3, with the complete procedure summarized in Algorithm 1. Further implementation details are provided in Appendix C.

5 Experiments

In this section, we evaluate TCE across diverse cross-domain setups and compare it with recent cross-domain offline RL baselines. We further perform ablations to quantify the contribution of each component and study sensitivity to key hyperparameters.

5.1 Experimental Setup

We evaluate TCE on cross-domain MuJoCo continuous-control tasks (Todorov et al., 2012) and Adroit manipulation tasks from the ODRL benchmark (Lyu et al., 2024b). In MuJoCo, the source and target domains share the same agent type (HalfCheetah, Hopper, Walker2d, or Ant) but differ through morphology, kinematic, or gravity shifts. Morphology shifts alter body-segment dimensions, kinematic shifts restrict joint rotation ranges, and gravity shifts halve the gravitational acceleration while preserving the agent structure. We use pre-collected D4RL datasets (Fu et al., 2020) of varying quality: `medium`, `medium-expert`, and `expert` for the target domain, and `medium`, `medium-replay`, and `medium-expert` for the source domain. This results in 36 cross-domain tasks for each shift setting, with 0.2M–2M source transitions and at most 5K target transitions. In Adroit, we consider morphological (`shrink-finger`) and kinematic (`broken-joint`) shifts at medium and hard target constraint levels (M/H). We use expert-only datasets containing 0.5M source transitions and 5K target transitions. We report normalized returns, where 0 and 100 correspond to random and expert policies, respectively.

For comparison, we evaluate TCE against cross-domain offline RL baselines: **IQL*** (Kostrikov et al., 2022), which trains IQL on source-target data; **DARA** (Liu et al., 2022), which applies classifier-based reward correction to mitigate dynamics mismatch; **BOSA** (Liu et al., 2024), which constrains policies to dataset support; **IGDF** (Wen et al., 2024), which filters source transitions via representation learning; **MOBODY** (Guo et al., 2026), a generation-based approach that learns target dynamics to augment data for policy optimization; and **OTDF** (Lyu et al., 2025), which filters source data using optimal transport distances. For Adroit, we use the official ODRL benchmark (Lyu et al., 2024b).

For IQL*, DARA, BOSA, IGDF, and OTDF, we report results from (Lyu et al., 2025), where each algorithm is implemented with hyperparameter tuning. MOBODY is evaluated using its official implementation. For TCE(Gen) and TCE(Mix), we use the coverage coefficient $\eta = 0.2$ unless otherwise specified. TCE(Gen) fixes $\lambda = 0$, whereas TCE(Mix) selects λ from $\{0.1, 0.5, 0.9\}$. More details, including hyperparameter settings, are provided in Appendix D.

5.2 Performance Comparison

Morphology Shifts. Table 1 shows that TCE consistently outperforms the baselines across morphology shifts, with at least one TCE variant achieving the highest mean score on 33 of 36 tasks. TCE also achieves a total score of 2065.2, substantially higher than the strongest baseline, OTDF, at 1274.3. In particular, TCE shows pronounced gains on `expert`-target tasks, where existing methods

Table 1: Performance comparison on morphology-shift tasks. Abbrev.: half=HalfCheetah, hopp=Hopper, walk=Walker2d, ant=Ant; m=`medium`, m-r=`medium-replay`, e=`expert`, m-e=`medium-expert`. Src./Tgt. denote source/target domains. Results are reported as mean ± standard deviation over 5 seeds; best results are in **bold** and second-best results are underlined.

Src.	Tgt.	IQL*	DARA	BOSA	IGDF	MOBODY	OTDF	TCE(Gen)	TCE(Mix)
half-m	m	30.0±1.6	26.6±3.3	19.3±3.5	41.6±0.5	42.5±1.2	39.1±2.3	**44.1±0.2**	**44.1±0.2**
half-m	m-e	31.8±1.1	32.0±0.7	33.6±1.1	29.6±2.2	26.4±2.5	35.6±0.7	**44.7±0.6**	43.3±0.5
half-m	e	8.5±1.0	9.3±1.6	7.9±0.8	10.0±0.8	9.3±0.8	10.7±1.2	**81.7±9.2**	61.5±4.5
half-m-r	m	30.8±4.4	35.6±0.7	35.0±4.6	28.0±2.0	41.1±0.6	40.0±1.2	**44.2±1.0**	43.3±2.3
half-m-r	m-e	12.9±2.2	16.9±4.1	19.9±5.5	12.0±3.7	22.3±2.1	34.4±0.7	**44.2±1.3**	43.1±0.1
half-m-r	e	5.9±1.7	3.7±2.7	2.4±1.9	5.3±2.3	4.3±2.4	8.2±2.7	**80.2±0.8**	36.9±8.6
half-m-e	m	41.5±0.1	40.3±1.2	41.3±0.3	40.9±0.4	42.3±1.7	41.4±0.3	**44.1±1.3**	**44.1±2.0**
half-m-e	m-e	25.8±2.0	30.6±2.8	32.1±0.8	26.2±1.8	27.5±2.8	35.1±0.6	**44.8±0.5**	43.8±2.0
half-m-e	e	7.8±1.3	8.3±1.3	9.1±0.8	7.5±0.9	9.9±1.8	9.8±1.0	**83.2±5.1**	62.9±8.3
hopp-m	m	13.5±0.2	13.5±0.4	13.2±0.3	13.4±0.2	13.3±0.5	11.0±0.9	**50.0±2.0**	5.0±3.7
hopp-m	m-e	13.4±0.1	13.6±0.2	11.2±4.6	13.3±0.4	13.0±0.9	12.6±0.8	**16.1±5.3**	9.2±1.8
hopp-m	e	13.5±0.2	13.6±0.3	13.3±0.4	13.9±0.1	13.4±0.4	10.7±4.7	**99.8±0.1**	77.3±15.7
hopp-m-r	m	10.8±1.1	10.2±1.0	1.2±0.0	12.0±4.4	5.6±3.4	8.7±2.8	**31.1±3.2**	27.0±14.5
hopp-m-r	m-e	11.6±1.6	10.4±0.9	1.3±0.2	8.2±2.8	11.6±2.3	9.7±2.7	**15.8±4.1**	3.4±0.1
hopp-m-r	e	9.8±0.5	9.0±0.3	1.3±0.1	11.4±1.5	9.4±1.8	10.7±2.4	**99.8±0.1**	75.4±26.1
hopp-m-e	m	12.6±1.4	13.0±0.5	15.7±7.2	12.7±0.8	13.6±0.9	7.9±3.2	**50.5±7.6**	22.0±17.3
hopp-m-e	m-e	14.1±1.3	13.8±0.6	12.0±1.4	13.3±1.2	13.2±2.7	9.6±3.5	**18.8±6.5**	10.2±1.3
hopp-m-e	e	13.8±0.5	12.3±1.8	10.5±5.0	12.8±0.9	13.1±2.2	5.9±4.0	**97.9±0.3**	66.1±23.1
walk-m	m	23.0±4.7	23.3±3.3	6.2±2.9	27.5±9.5	36.6±9.7	50.5±5.8	43.2±3.8	**55.2±5.3**
walk-m	m-e	21.5±8.6	22.2±7.6	7.2±2.9	20.7±5.9	20.6±9.4	**44.3±23.8**	40.4±2.5	22.7±9.7
walk-m	e	20.3±2.8	17.3±3.4	15.8±8.7	15.8±4.5	31.5±15.6	55.3±8.3	74.2±1.6	**86.8±11.8**
walk-m-r	m	11.3±3.0	10.9±4.6	5.4±4.0	13.4±7.2	27.0±13.4	37.4±5.1	44.7±2.7	**45.6±10.4**
walk-m-r	m-e	7.0±1.5	4.5±1.1	4.0±2.2	6.9±2.2	17.5±12.8	33.8±6.9	**43.4±2.6**	17.2±3.1
walk-m-r	e	6.3±0.9	4.5±1.1	3.8±3.4	5.5±2.2	24.5±15.1	41.5±6.8	**63.2±7.4**	52.6±0.2
walk-m-e	m	24.1±7.4	31.7±6.6	18.7±6.5	27.5±2.3	37.7±18.5	**49.9±4.6**	43.5±3.4	47.4±3.6
walk-m-e	m-e	27.0±5.5	23.3±5.5	11.1±0.9	25.3±6.4	24.1±12.1	**40.5±11.0**	37.4±6.1	21.7±7.9
walk-m-e	e	22.4±3.3	25.2±5.7	9.9±3.9	24.7±2.4	26.2±9.6	45.7±6.9	**62.5±9.6**	47.6±10.9
ant-m	m	38.7±3.8	41.3±1.8	18.2±1.9	40.9±1.7	40.6±0.8	39.4±1.7	**41.5±0.8**	40.8±1.2
ant-m	m-e	47.0±5.1	43.3±2.0	45.3±7.0	44.4±1.7	54.7±2.5	58.3±8.9	72.7±2.5	**74.9±4.0**
ant-m	e	36.2±3.5	48.5±4.2	72.2±10.5	41.4±4.2	58.0±7.4	85.4±4.4	94.0±4.5	**96.4±3.2**
ant-m-r	m	38.2±2.9	38.9±2.7	20.2±3.7	39.7±1.2	40.8±2.8	41.2±0.9	**41.5±1.5**	40.0±1.6
ant-m-r	m-e	38.1±3.5	33.4±5.5	15.2±1.6	37.3±2.4	49.4±3.6	50.8±4.5	**71.0±8.4**	63.1±9.8
ant-m-r	e	24.1±1.9	24.5±2.6	16.0±1.7	23.6±1.4	32.4±8.7	67.2±7.5	**96.1±0.4**	**96.1±0.6**
ant-m-e	m	32.9±5.1	40.2±1.5	28.1±5.6	36.1±4.4	39.5±3.5	39.9±2.9	**41.9±0.6**	41.4±1.8
ant-m-e	m-e	35.7±3.9	36.5±8.7	14.8±15.9	30.7±10.8	50.3±9.2	65.7±4.5	66.4±1.2	**69.2±4.9**
ant-m-e	e	36.1±8.5	34.6±5.8	53.9±5.0	35.2±6.6	44.9±6.5	86.4±2.2	**96.6±0.5**	94.5±2.4
Total Score		798.0	816.8	646.3	808.7	988.1	1274.3	**2065.2**	1731.8

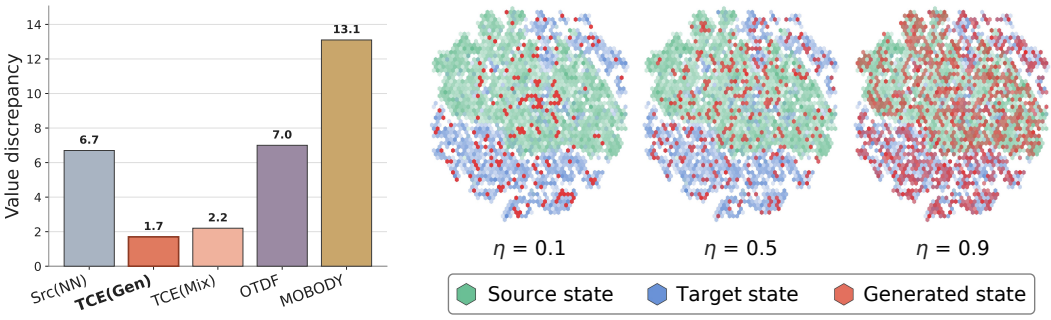
often struggle under limited target coverage. Morphology changes induce relatively large dynamics gaps by directly altering the physical structure of the agent. In these settings, the dynamics gap is generally large relative to the generation error, and TCE(Gen) outperforms TCE(Mix) on most tasks. We further analyze this relationship in Section 5.3.

Kinematic and Gravity Shifts. Full results for kinematic and gravity shifts are provided in Appendix E. Kinematic shifts exhibit a similar trend to morphology shifts due to their relatively large dynamics gaps: TCE achieves the highest mean score on 30 of 36 tasks, with TCE(Gen) generally outperforming TCE(Mix). In contrast, gravity shifts induce smaller dynamics gaps while preserving agent morphology and joint ranges. TCE remains competitive across these tasks, but TCE(Mix) generally outperforms TCE(Gen), indicating greater benefits from directly incorporating target-near source transitions.

To further examine the effect of the dynamics gap, we progressively vary the target gravity while fixing the source gravity in Appendix E.3. TCE(Mix) performs better when the gravity difference is small, whereas TCE(Gen) becomes increasingly favorable as the difference grows. This transition supports our finding that *direct source reuse is more beneficial under small dynamics gaps, whereas large dynamics gaps favor target-aligned generation.*

Table 2: Performance comparison on Adroit tasks from the ODRL benchmark.

Env	Shift	Level	IQL*	DARA	BOSA	IGDF	MOBODY	OTDF	TCE(Gen)	TCE(Mix)
Pen	broken-joint	M	15.5±3.4	33.8±2.8	20.4±2.5	29.8±2.8	41.3±3.5	36.6±2.9	**59.6±3.7**	30.5±0.4
		H	24.1±1.6	13.3±3.3	19.8±1.3	14.0±1.9	34.5±3.1	40.1±4.1	**44.8±2.2**	23.6±1.6
	shrink-finger	M	18.6±2.2	16.2±3.3	17.4±1.3	22.2±0.9	24.9±2.0	34.5±5.0	**38.6±3.1**	25.0±12.0
		H	16.4±1.7	24.5±2.8	9.5±1.6	23.5±7.2	29.7±1.5	32.4±3.9	**51.9±4.2**	20.4±3.7
Door	broken-joint	M	50.8±2.4	50.9±6.2	47.8±5.4	73.2±8.8	79.4±1.8	80.4±1.6	**82.8±1.3**	82.6±1.1
		H	50.7±7.2	36.2±4.5	8.8±2.1	53.4±7.6	54.3±2.3	78.5±2.1	**91.8±0.9**	81.0±0.2
	shrink-finger	M	28.3±5.2	58.6±8.4	0.8±0.2	73.2±9.1	50.6±0.9	70.2±1.1	**77.2±3.0**	71.4±5.3
		H	14.1±7.7	8.8±1.9	9.3±1.0	35.0±6.5	42.6±2.4	56.5±2.0	**66.9±0.8**	57.5±4.8
Total Score			218.5	242.3	133.8	324.3	357.3	429.2	**513.6**	392.0



(a) Value discrepancy under morphology shifts.

(b) Coverage expansion with varying η under morphology shifts.

Figure 4: Analysis of value discrepancy and coverage expansion.

Adroit Tasks. Across the Adroit settings in Table 2, TCE(Gen) achieves the highest mean score in every setting, outperforming all baselines and TCE(Mix). Its advantage over TCE(Mix) is particularly pronounced on Pen, while the two variants remain comparable in some Door settings. In these high-dimensional manipulation tasks, changes in finger morphology and joint constraints can induce larger source–target dynamics gaps, making direct source reuse less favorable and providing greater benefits to TCE(Gen). These results demonstrate the effectiveness of target-aligned generation in challenging high-dimensional manipulation settings.

5.3 Analyzing the Source–Generation Trade-off

To better understand why TCE achieves strong performance, we investigate two questions motivated by our theoretical analysis: (1) Can the performance trends of TCE be explained by the dynamics gap and generation error identified in Theorem 4.1? (2) How does generation error change as TCE expands coverage, and how far can coverage be expanded before the error outweighs its benefit?

Analysis of the Theoretical Gap. To address the first question, we empirically evaluate the value-discrepancy formulation of the performance gap in Theorem 4.1 under morphology shifts, using a reference target value function trained on additional target data solely for evaluation. We consider Src(NN), consisting of source transitions nearest to the target data, along with TCE(Gen), TCE(Mix), OTDF, and MOBODY. As shown in Fig. 4(a), Src(NN) exhibits a large value discrepancy despite its proximity to the target data, indicating that state proximity does not necessarily imply a small dynamics gap. In contrast, TCE(Gen) achieves a substantially smaller discrepancy, supporting its advantage over direct source reuse when the dynamics gap is large relative to the generation error. TCE(Mix) exhibits a larger discrepancy due to the additional use of source transitions, while OTDF and MOBODY similarly show larger discrepancies. These results demonstrate that our theoretical analysis captures the relative reliability of different ways of utilizing source information and explains why target-aligned generation is particularly effective under large dynamics gaps.

Coverage–Generation Error Trade-off. While reducing the discrepancy introduced by source utilization is beneficial, minimizing discrepancy alone is insufficient when target coverage is limited. We therefore examine how coverage expansion affects generation error and policy performance. As shown in Fig. 4(b), increasing η progressively expands generated samples from the limited target-state region toward the broader source-state region. Table 3 shows that this expansion gradually increases action, reward, and transition errors relative to ground-truth target transitions obtained

Table 3: Generation errors, value discrepancies, and average normalized scores across coverage coefficients η on 36 MuJoCo morphology-shift tasks.

Model	η	Error			Value Discrepancy	Average Score
		Action	Reward	Trans.		
TCE(Gen)	0	**0.038±0.009**	**0.024±0.010**	**0.016±0.005**	**1.47±0.77**	54.6±25.1
	0.1	0.062±0.011	0.025±0.012	0.024±0.013	2.56±0.82	55.8±24.1
	0.2	0.066±0.014	0.025±0.013	0.029±0.012	3.01±0.89	**57.4±24.6**
	0.5	0.085±0.021	0.052±0.014	0.046±0.015	3.71±1.15	52.1±25.5
	0.9	0.118±0.020	0.069±0.017	0.052±0.019	3.85±0.97	50.9±26.1
TCE(Full-Trans.)	0.2	0.180±0.045	0.105±0.015	0.159±0.052	8.22±1.50	46.5±22.6

from the target simulator, along with the resulting value discrepancy. We also compare TCE(Gen) with TCE(Full-Trans.), which directly generates (a, r, s') given s. As discussed in Section 4.3, full-transition generation generalizes less effectively under scarce target data, resulting in larger generation errors and lower policy performance than our state-pair-first design.

With $\eta = 0$, generation achieves the smallest errors and value discrepancy but remains confined to the limited target coverage. Moderate expansion improves performance despite increased generation error and value discrepancy, indicating that the benefit of broader coverage can outweigh the increased error. With excessive expansion, however, the increased generation error outweighs the benefit of additional coverage, causing performance to decline. These results reveal a clear trade-off between coverage expansion and generation error, highlighting the importance of expanding beyond limited target support while keeping the resulting generation error sufficiently small.

5.4 Further Ablation and Analysis

Component Evaluation. Table 4 further evaluates the selection strategy and policy regularization under morphology shifts. We compare TCE(Gen-OT) and TCE(Mix-OT), which replace NN selection with optimal-transport (OT) selection (Lyu et al., 2025), and TCE(w/o Reg.), which removes policy regularization. The OT variants perform comparably to their NN-based counterparts, with the differences between NN and OT being smaller than those between TCE(Gen) and TCE(Mix). This indicates that the relative benefit of generation versus direct source reuse is more influential than the specific selection strategy in these settings. TCE(w/o Reg.) shows only a modest performance decrease, indicating a limited contribution from policy regularization.

Table 4: Component evaluation using normalized scores averaged over 36 morphology-shift tasks.

Setting	Avg. Score
TCE(Gen)	**57.4±24.6**
TCE(Gen-OT)	55.6±24.2
TCE(Mix)	48.1±25.5
TCE(Mix-OT)	47.5±25.1
TCE(w/o Reg.)	55.8±26.8
IQL*	22.2±11.9

Further Analysis. We additionally provide a computational comparison in Appendix G, showing that TCE(Gen) maintains peak GPU memory usage comparable to the baselines despite additional preprocessing. We further analyze the sensitivity to η and λ across additional domain shifts in Appendix H.1, with results consistent with our main coverage–error and source–generation analyses. Appendix H.2 examines the effect of target-data size, showing that additional target data reduces inverse-dynamics error, while performance gains diminish beyond 5K transitions. Finally, we analyze the effect of denoising steps in Appendix H.3, where TCE(Gen) exhibits only modest performance variation across the tested values. Limitations and future directions are discussed in Appendix A.

6 Conclusion

We presented Target-aligned Coverage Expansion (TCE), which leverages source states to expand the limited target coverage through target-aligned generation while controlling generation error. Our performance-gap analysis characterizes the interplay between the source–target dynamics gap and generation error, providing guidance for effective coverage expansion and source utilization. Experiments across diverse dynamics shifts demonstrate strong performance and validate this trade-off: moderate expansion improves performance despite increased generation error, while direct source reuse becomes beneficial under smaller dynamics gaps. These findings highlight the importance of balancing coverage expansion against generation error and demonstrate that source states can effectively guide target-aligned generation even when large dynamics gaps limit direct source reuse.

AI USE STATEMENT

Generative AI tools were used solely to improve the clarity and phrasing of the manuscript. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final content.

ETHICS STATEMENT

This work does not involve human subjects or personally identifiable information, and all experiments were conducted in simulated environments using offline reinforcement learning benchmarks. Because the proposed method augments training data with generated transitions, applications in safety-critical domains require validation in the real environment before a learned policy is deployed.

REPRODUCIBILITY STATEMENT

The assumptions underlying our theoretical results are stated in Section 4.2, and the complete proof is provided in Appendix B. Implementation details and experimental settings are described in Appendices C and D, respectively. Our experiments build on the D4RL and ODRL benchmarks, and the dataset compositions and environment-specific construction procedures are documented in the appendix. Our code is submitted as anonymized supplementary material.

REFERENCES

Jongseong Chae, Seungyul Han, Whiyoung Jung, Myungsik Cho, Sungho Choi, and Youngchul Sung. Robust imitation learning against variations in environment dynamics. In International Conference on Machine Learning, pp. 2828–2852. PMLR, 2022.

Jongseong Chae, Jongeui Park, Yongjae Shin, Gyeongmin Kim, Seungyul Han, and Youngchul Sung. Flow actor-critic for offline reinforcement learning. In The Fourteenth International Conference on Learning Representations, 2026.

Sungho Choi, Seungyul Han, Woojun Kim, Jongseong Chae, Whiyoung Jung, and Youngchul Sung. Domain adaptive imitation learning with visual observation. Advances in Neural Information Processing Systems, 36:44067–44104, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. Advances in neural information processing systems, 34:8780–8794, 2021.

Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Ruslan Salakhutdinov, and Sergey Levine. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In 4th Lifelong Machine Learning Workshop at ICML 2020, 2020.

Arnaud Fickinger, Samuel Cohen, Stuart Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. In International Conference on Learning Representations, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, pp. 1126–1135. PMLR, 2017.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. CoRR, abs/2004.07219, 2020.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

Omer Gottesman, Fredrik D. Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, Jiayu Yao, Isaac Lage, Christopher Mosch, Li-Wei H. Lehman, Matthieu Komorowski, Aldo Faisal, Leo Anthony Celi, David A. Sontag, and Finale Doshi-Velez. Evaluating reinforcement learning algorithms in observational health settings. CoRR, abs/1805.12298, 2018.

Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Jing Huang, Larry Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. Advances in Neural Information Processing Systems, 35:28319–28334, 2022.

Yihong Guo, Yu Yang, Pan Xu, and Anqi Liu. MOBODY: Model-based off-dynamics offline reinforcement learning. In The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=7c0YS3cuno.

Seungyul Han, Sanghyeon Lee, Sangjun Bae, and Yisak Park. Self-improving skill learning for robust skill-based meta-reinforcement learning. In International conference on learning representations, volume 2026, pp. 108189–108225, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems, 35:26565–26577, 2022.

Jeongmo Kim, Yisak Park, Minung Kim, and Seungyul Han. Task-aware virtual training: Enhancing generalization in meta-reinforcement learning for out-of-distribution tasks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 30720–30748. PMLR, 13–19 Jul 2025.

Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. Domain adaptive imitation learning. In International Conference on Machine Learning, pp. 5286–5295. PMLR, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

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.

Linh Le Pham Van, Minh Hoang Nguyen, Duc Kieu, Hung Le, Sunil Gupta, et al. Dmc: Nearest neighbor guidance diffusion model for offline cross-domain reinforcement learning. In ECAI 2025, pp. 2331–2338. IOS Press, 2025.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. CoRR, abs/2005.01643, 2020.

Guanghe Li, Yixiang Shan, Zhengbang Zhu, Ting Long, and Weinan Zhang. Diffstitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching. arXiv preprint arXiv:2402.02439, 2024.

Jinxin Liu, Zhang Hongyin, and Donglin Wang. DARA: Dynamics-aware reward augmentation in offline reinforcement learning. In International Conference on Learning Representations, 2022.

Jinxin Liu, Ziqi Zhang, Zhenyu Wei, Zifeng Zhuang, Yachen Kang, Sibo Gai, and Donglin Wang. Beyond OOD state actions: Supported cross-domain offline reinforcement learning. the AAAI Conference on Artificial Intelligence, 2024.

Yunhao Luo, Utkarsh A Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition. arXiv preprint arXiv:2503.05153, 2025.

Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline reinforcement learning. Advances in Neural Information Processing Systems, 35:1711–1724, 2022.

Jiafei Lyu, Chenjia Bai, Jingwen Yang, Zongqing Lu, and Xiu Li. Cross-domain policy adaptation by capturing representation mismatch. arXiv preprint arXiv:2405.15369, 2024a.

Jiafei Lyu, Kang Xu, Jiacheng Xu, Jing-Wen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, Xiu Li, et al. Odrl: A benchmark for off-dynamics reinforcement learning. Advances in Neural Information Processing Systems, 37:59859–59911, 2024b.

Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In The Thirteenth International Conference on Learning Representations, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In International conference on machine learning, pp. 8162–8171. PMLR, 2021.

Haoyi Niu, Qimao Chen, Tenglong Liu, Jianxiong Li, Guyue Zhou, Yi ZHANG, Jianming HU, and Xianyuan Zhan. xted: Cross-domain adaptation via diffusion-based trajectory editing. In NeurIPS 2024 Workshop on Open-World Agents, 2024.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In ICML, volume 97 of Proceedings of Machine Learning Research, pp. 5171–5180. PMLR, 2019.

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. arXiv preprint arXiv:2512.02486, 2025.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. Advances in neural information processing systems, 33:12438–12448, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems, pp. 5026–5033. IEEE, 2012.

Zhi Wang, Li Zhang, Wenhao Wu, Yuanheng Zhu, Dongbin Zhao, and Chunlin Chen. Meta-dt: Offline meta-rl as conditional sequence modeling with world model disentanglement. Advances in Neural Information Processing Systems, 37:44845–44870, 2024.

Xiaoyu Wen, Chenjia Bai, Kang Xu, Xudong Yu, Yang Zhang, Xuelong Li, and Zhen Wang. Contrastive representation for data filtering in cross-domain offline reinforcement learning. arXiv preprint arXiv:2405.06192, 2024.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. CoRR, abs/1911.11361, 2019.

Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-domain policy adaptation via value-guided data filtering. In Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Junghyuk Yeom, Yonghyeon Jo, Jungmo Kim, Sanghyeon Lee, and Seungyul Han. Exclusively penalized q-learning for offline reinforcement learning. Advances in Neural Information Processing Systems, 37:113405–113435, 2024.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. IEEE access, 8:58443–58469, 2020.

A Limitations and Future Work

Although TCE consistently outperforms baselines in our experiments, it incurs additional training and sampling costs. However, Appendix G shows that its peak GPU memory usage remains comparable to the baselines. TCE also introduces two hyperparameters for coverage expansion and source–generation mixing, but our analysis provides guidance for their selection by considering the coverage–error trade-off and the source–target dynamics gap. In addition, while TCE is effective under the large dynamics gaps evaluated in this work, it does not yet address extreme domain differences where source and target state distributions have little overlap or their state or action spaces differ. Extending TCE to such settings remains an important direction for future work.

B Proof of Theorem 4.1

We begin by introducing the *Telescoping Lemma* (Xu et al., 2023), a fundamental result that decomposes the performance difference between two MDPs arising from their dynamic mismatch. This lemma serves as a cornerstone for our analysis.

Lemma B.1 (Telescoping Lemma, Lemma C.1 in Xu et al. (2023))**.** *Let $\mathcal{M}_1 := (\mathcal{S}, \mathcal{A}, P_1, r, \gamma)$ and $\mathcal{M}_2 := (\mathcal{S}, \mathcal{A}, P_2, r, \gamma)$ be two MDPs differing only in their transition dynamics P_1 and P_2. For any policy π, defining the value discrepancy term as*

$$G^{\pi}_{\mathcal{M}_1,\mathcal{M}_2}(s,a) := \mathbb{E}_{s'\sim P_1}\left[V^{\pi}_{\mathcal{M}_2}(s')\right] - \mathbb{E}_{s'\sim P_2}\left[V^{\pi}_{\mathcal{M}_2}(s')\right],$$

allows us to express the performance gap as:

$$J_{\mathcal{M}_1}(\pi) - J_{\mathcal{M}_2}(\pi) = \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim\rho^{\pi}_{\mathcal{M}_1}}\left[G^{\pi}_{\mathcal{M}_1,\mathcal{M}_2}(s,a)\right].$$

The proof of Lemma B.1 is provided in Xu et al. (2023). Building upon this result, we restate and prove Theorem 4.1, which bounds the performance gap between the mixture and target domains through transition and value discrepancies.

Theorem 4.1 *Let $J_{\mathrm{mix}}(\pi)$ and $\rho^{\pi}_{\mathrm{mix}}$ denote the expected return and the state-action occupancy measure in the mixture MDP, respectively. Then, the performance gap between the mixture MDP and the target MDP can be bounded in terms of transition dynamics as follows:*

$$J_{\mathrm{mix}}(\pi) - J_{\mathrm{tar}}(\pi) \le \frac{2\gamma r_{\max}}{(1-\gamma)^2}\Big(\lambda\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[D_{\mathrm{TV}}(P_{\mathrm{src}}\,\|\,P_{\mathrm{tar}})\right] + (1-\lambda)\,\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[D_{\mathrm{TV}}(\hat{P}_{\mathrm{tar}}\,\|\,P_{\mathrm{tar}})\right]\Big), \quad (11)$$

where $|r(s,a)| \le r_{\max}$ and $D_{\mathrm{TV}}(P\,\|\,Q)$ denotes the total variation distance between P and Q.

Proof of the total-variation bound. To simplify notation, let P_{mix}, P_{src}, and P_{tar} denote $P_{\mathrm{mix}}(\cdot \mid s,a)$, $P_{\mathrm{src}}(\cdot \mid s,a)$, and $P_{\mathrm{tar}}(\cdot \mid s,a)$ respectively. Similarly, $\rho^{\pi}_{\mathrm{mix}}$ denotes $\rho^{\pi}_{\mathrm{mix}}(s,a)$, and V^{π}_{tar} represents $V^{\pi}_{\mathrm{tar}}(s')$.

Applying Lemma B.1 with $\mathcal{M}_1 = \mathrm{mix}$ and $\mathcal{M}_2 = \mathrm{tar}$, we have:

$$\begin{aligned}
J_{\mathrm{mix}}(\pi) - J_{\mathrm{tar}}(\pi) &= \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[\int P_{\mathrm{mix}} V^{\pi}_{\mathrm{tar}} ds' - \int P_{\mathrm{tar}} V^{\pi}_{\mathrm{tar}} ds'\right] && \text{(by Lemma B.1)}\\
&\le \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[\int |P_{\mathrm{mix}} - P_{\mathrm{tar}}|\cdot|V^{\pi}_{\mathrm{tar}}|\,ds'\right] && \text{(Triangle Inequality)}\\
&\le \frac{\gamma}{1-\gamma}\cdot\frac{r_{\max}}{1-\gamma}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[\int |P_{\mathrm{mix}} - P_{\mathrm{tar}}|\,ds'\right]\\
&= \frac{2\gamma r_{\max}}{(1-\gamma)^2}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[\frac{1}{2}\int |P_{\mathrm{mix}} - P_{\mathrm{tar}}| ds'\right]\\
&= \frac{2\gamma r_{\max}}{(1-\gamma)^2}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[D_{\mathrm{TV}}(P_{\mathrm{mix}}\|P_{\mathrm{tar}})\right]\\
&= \frac{2\gamma r_{\max}}{(1-\gamma)^2}\mathbb{E}_{\rho^{\pi}_{\mathrm{mix}}}\left[D_{\mathrm{TV}}((\lambda P_{\mathrm{src}} + (1-\lambda)\hat{P}_{\mathrm{tar}})\|P_{\mathrm{tar}})\right]
\end{aligned}$$

$$\leq \frac{2\gamma r_{\max}}{(1-\gamma)^2}\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\lambda D_{\text{TV}}(P_{\text{src}}\|P_{\text{tar}}) + (1-\lambda)D_{\text{TV}}(\hat{P}_{\text{tar}}\|P_{\text{tar}})\Big]$$
$$= \frac{2\gamma r_{\max}}{(1-\gamma)^2}\Big(\lambda\,\mathbb{E}_{\rho^\pi_{\text{mix}}}\big[D_{\text{TV}}(P_{\text{src}}\|P_{\text{tar}})\big] + (1-\lambda)\,\mathbb{E}_{\rho^\pi_{\text{mix}}}\big[D_{\text{TV}}(\hat{P}_{\text{tar}}\|P_{\text{tar}})\big]\Big)$$

where the inequality follows from the convexity of total variation distance and the final equality follows from linearity of expectation, yielding the desired bound. □

We next restate and prove the value-discrepancy bound in Theorem 4.1.

Letting V^π_{tar} denote the target-domain value function, the performance gap is bounded by the corresponding value discrepancies as

$$J_{\text{mix}}(\pi) - J_{\text{tar}}(\pi) \leq \frac{\gamma}{1-\gamma}\Big(\lambda\,\mathbb{E}_{\rho^\pi_{\text{mix}}}[|\mathbb{E}_{P_{\text{src}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]|] + (1-\lambda)\,\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\Big|\mathbb{E}_{\hat{P}_{\text{tar}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\Big|\Big]\Big).$$

Proof of the value-discrepancy bound. To simplify notation, let P_{mix}, P_{src}, and P_{tar} denote $P_{\text{mix}}(\cdot \mid s,a)$, $P_{\text{src}}(\cdot \mid s,a)$, and $P_{\text{tar}}(\cdot \mid s,a)$ respectively. Similarly, ρ^π_{mix} denotes $\rho^\pi_{\text{mix}}(s,a)$, and V^π_{tar} represents $V^\pi_{\text{tar}}(s')$.

Applying Lemma B.1 with $\mathcal{M}_1 = \text{mix}$ and $\mathcal{M}_2 = \text{tar}$, we have:

$$J_{\text{mix}}(\pi) - J_{\text{tar}}(\pi) = \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\left[\int P_{\text{mix}}V^\pi_{\text{tar}}ds' - \int P_{\text{tar}}V^\pi_{\text{tar}}ds'\right] \qquad \text{(by Lemma B.1)}$$
$$= \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\left[\mathbb{E}_{P_{\text{mix}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\right]$$
$$\leq \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\left[\left|\mathbb{E}_{P_{\text{mix}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\right|\right]$$
$$= \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\left[\left|\lambda\mathbb{E}_{P_{\text{src}}}[V^\pi_{\text{tar}}(s')] + (1-\lambda)\mathbb{E}_{\hat{P}_{\text{tar}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\right|\right]$$
$$= \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\Big|\lambda(\mathbb{E}_{P_{\text{src}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]) + (1-\lambda)(\mathbb{E}_{\hat{P}_{\text{tar}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')])\Big|\Big]$$
$$\leq \frac{\gamma}{1-\gamma}\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\lambda\big|\mathbb{E}_{P_{\text{src}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\big| + (1-\lambda)\big|\mathbb{E}_{\hat{P}_{\text{tar}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\big|\Big]$$
$$= \frac{\gamma}{1-\gamma}\left(\lambda\,\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\big|\mathbb{E}_{P_{\text{src}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\big|\Big] + (1-\lambda)\,\mathbb{E}_{\rho^\pi_{\text{mix}}}\Big[\big|\mathbb{E}_{\hat{P}_{\text{tar}}}[V^\pi_{\text{tar}}(s')] - \mathbb{E}_{P_{\text{tar}}}[V^\pi_{\text{tar}}(s')]\big|\Big]\right)$$

where the inequality follows from the triangle inequality and the final equality follows from linearity of expectation, proving the claim. □

Theorem 4.1 highlights that effective coverage expansion requires not only broadening the target support, but also controlling the resulting performance gap. This motivates expanding the support of target-aligned generation without inducing excessive generation error and using the source–target dynamics gap to guide direct source incorporation. When this gap is small, target-near source transitions can complement the generated data; when it is large, their direct use becomes less favorable, and source data are instead used primarily for coverage expansion.

C Detailed Implementation and Algorithm of TCE

This section summarizes core components of our proposed TCE method: Appendix C.1 provides the configuration of the score-based generative models. Appendix C.2 details the auxiliary models, dataset construction, and the offline policy learning. Appendix C.3 provides network architecture and configuration details supporting the overall implementation.

C.1 Details of Score-based Generative Model with SDEs

Our generative framework follows the score-based modeling with SDEs described in Section 3.2. We adopt the variance exploding (VE) SDE formulation with drift $f(x, \tau) = 0$ and diffusion coefficient $g(\tau) = \sqrt{d(\sigma^2)/d\tau}$. In our implementation, we use a VP-style variance schedule while retaining the VE reverse-time dynamics and predictor-corrector sampling, which can be viewed as a parameterization choice within the score-SDE framework. The noise scale $\sigma(\tau)$ is defined using a quadratic schedule for the exponent $\mathcal{B}(\tau)$:

$$\sigma(\tau) = \sqrt{1 - \exp\left(-\mathcal{B}(\tau)\right)}, \qquad \mathcal{B}(\tau) = \alpha_{\min}\tau + \frac{1}{2}(\alpha_{\max} - \alpha_{\min})\tau^2, \tag{12}$$

where we fix $\alpha_{\min} = 0.1$ and $\alpha_{\max} = 20$ to ensure a smooth noise injection process. This choice is better suited to our low-dimensional, data-scarce offline RL setting than the canonical geometric VE schedule, as it gives smoother noise growth and more stable score estimation under limited data.

The score network is optimized via the denoising score matching loss (Eq. (1) in main text) using the weighting function $\zeta(\tau) = \sigma(\tau)^2$. In the absence of conditional input c, this formulation simplifies to an unconditional score network $q_\theta(x^\tau, \tau)$.

For sample generation, we employ the predictor-corrector (PC) scheme with $K = 500$ discretization steps. This method combines a numerical SDE solver (Predictor) with Langevin dynamics (Corrector) to improve robustness and sample quality. The predictor integrates the reverse SDE:

$$\text{(Predictor)} \quad x^{k-1} = x^k - g(\tau^k)^2 q_\theta(x^k, \tau^k \mid c)\Delta\tau^k + g(\tau^k)\sqrt{|\Delta\tau^k|}\,\xi^k, \quad \xi^k \sim \mathcal{N}(0, I). \tag{13}$$

Following the predictor step, a Langevin corrector step refines the sample based on the score function:

$$\text{(Corrector)} \quad x^{k-1} \leftarrow x^{k-1} + \epsilon^k q_\theta(x^{k-1}, \tau^{k-1} \mid c) + \sqrt{2\epsilon^k}\xi^k, \quad \xi^k \sim \mathcal{N}(0, I), \tag{14}$$

where ϵ^k is a step-size parameter for the corrector. This refinement helps correct errors accumulated during the numerical integration, yielding higher-quality transitions.

C.2 Detailed Implementation of TCE

In this section, we provide implementation details for the auxiliary models, the dataset construction, and the offline policy learning. The training and sampling for the score-based generative models follow the methodology described in Section 4.3.

Auxiliary Models and Labeling. Since the score-based models generate only state transitions $(\hat{s}_t, \hat{s}_{t+1})$, we require auxiliary models to recover the missing actions and rewards. We train an inverse dynamics model Inv_ψ and a reward model $\hat{R}_\phi$ using the mean squared error (MSE) loss:

$$\mathcal{L}_{\text{aux}}(\psi, \phi) = \underbrace{\mathbb{E}_{\mathcal{D}_{\text{tar}}}\|\mathrm{Inv}_\psi(s_t, s_{t+1}) - a_t\|_2^2}_{\textbf{inverse dynamics loss}} + \underbrace{\mathbb{E}_{\mathcal{D}_{\text{tar}}\cup\mathcal{D}_{\text{src}}}\|\hat{R}_\phi(s_t, a_t, s_{t+1}) - r_t\|_2^2}_{\textbf{reward loss}}. \tag{15}$$

The inverse dynamics model is trained solely on $\mathcal{D}_{\text{tar}}$ to ensure consistency with target dynamics, while the reward model is trained on $\mathcal{D}_{\text{tar}} \cup \mathcal{D}_{\text{src}}$ to leverage the shared reward function. We then utilize these models to label the generated state transitions $(\hat{s}_t, \hat{s}_{t+1})$ with actions and rewards, forming the fully labeled dataset $\mathcal{D}^\eta_{\text{gen}}$.

Dataset Construction and IQL Training. For a mixture coefficient λ, we retain the closest $\lambda|\mathcal{D}_{\text{src}}|$ source transitions under the same d_{NN} ranking and generate $(1-\lambda)|\mathcal{D}_{\text{src}}|$ target-aligned transitions. The resulting dataset is $\mathcal{D}^{\eta,\lambda}_{\text{train}} = \mathcal{D}_{\text{tar}} \cup \mathcal{D}^\lambda_{\text{src}} \cup \mathcal{D}^\eta_{\text{gen}}$. TCE(Gen) uses $\lambda = 0$, whereas TCE(Mix) uses $\lambda > 0$; both therefore have the same total number of policy-training transitions.

For policy learning, we employ Implicit Q-Learning (IQL) (Kostrikov et al., 2022) on $\mathcal{D}_{\text{train}}^{\eta,\lambda}$. For brevity, we write this dataset as $\mathcal{D}_{\text{train}}$ in the objectives below. The training objectives for the value function V_φ, Q-function Q_φ, and policy π_ω are defined as follows:

$$\mathcal{L}_V(\varphi) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_{\text{train}}}\left[L_2^{\tau_V}\left(Q_{\varphi'}(s_t,a_t) - V_\varphi(s_t)\right)\right], \tag{16}$$

$$\mathcal{L}_Q(\varphi) = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}_{\text{train}}}\left[\left(r_t + \gamma V_{\varphi^-}(s_{t+1}) - Q_\varphi(s_t,a_t)\right)^2\right], \tag{17}$$

$$\mathcal{L}_\pi(\omega) = -\mathbb{E}_{(s_t,a_t)\sim\mathcal{D}_{\text{train}}}\Big[\exp\left(\kappa\cdot\text{Adv}(s_t,a_t)\right)\,\log\pi_\omega(a_t|s_t)\Big] + \beta\,\mathbb{E}_{s_t\sim\mathcal{D}_{\text{tar}}}\Big[D_{\text{KL}}\big(\hat{\pi}_b(\cdot|s_t)\,\|\,\pi_\omega(\cdot|s_t)\big)\Big], \tag{18}$$

where $L_2^{\tau_V}(u) = |\tau_V - \mathbb{1}[u<0]|\,u^2$ is the expectile loss with $\tau_V = 0.7$; φ' and φ^- denote the target network and stop-gradient parameters, respectively; $\text{Adv}(s_t,a_t) = Q_\varphi(s_t,a_t) - V_\varphi(s_t)$ is the advantage weighted by temperature $\kappa = 3$; and $\hat{\pi}_b$ is the empirical behavior policy cloned from $\mathcal{D}_{\text{tar}}$, with β controlling the KL-divergence penalty.

C.3 Network Architecture and Configurations

We employ four core neural network architectures to model coverage-expansion scores, target-transition scores, inverse dynamics, and reward estimation. As summarized in Table 5, all networks leverage MLP backbones with residual connections to ensure effective gradient flow. Details of environment-specific state and action dimensions (d_s and d_a) are provided in Table 6.

Coverage-expansion score network (q_θ^{cov}) estimates the score of perturbed states conditioned on diffusion time. It takes the state s_t and time τ as inputs. The time embedding from a 2-layer MLP is concatenated with s_t and passed through residual dense blocks to output a score in $\mathbb{R}^{d_s}$.

Target-transition score network (q_θ^{tran}) predicts the score of the next state s_{t+1} conditioned on s_t and τ. Like the coverage-expansion score network, it embeds τ and s_t with separate MLPs. These embeddings are concatenated with s_{t+1} and passed through residual blocks to output a score in $\mathbb{R}^{d_s}$.

Inverse dynamics (Inv_ψ) and reward models ($\hat{R}_\phi$) map state transition pairs $[s_t, s_{t+1}]$ to the action space $\mathbb{R}^{d_a}$ and scalar rewards, respectively. Both networks share the same architecture, starting with a dense layer followed by residual blocks.

Table 5: Detailed architectures of the score and dynamics models.

Notation: MLP$(x;[h_1,\ldots,h_k])$ denotes an MLP with hidden sizes h_i and SiLU activations. Linear(d) is a fully connected layer with output dimension d. $[\text{ResBlock}(d)]_{\times N}$ denotes a stack of N residual MLP blocks.

Network	Inputs	Architecture Specification
Coverage-expansion score (q_θ^{cov})	State s_t Time τ	$h_\tau = \text{MLP}(\tau;[128,128])$ $h_{\text{in}} = \text{Concat}[s_t, h_\tau]$ $\text{MLP}(h_{\text{in}};[256]) \to [\text{ResBlock}(256)]_{\times 4} \to \text{Linear}(d_s)$
Target-transition score (q_θ^{tran})	State s_t Time τ Next state s_{t+1}	$h_\tau = \text{MLP}(\tau;[128,128])$ $h_s = \text{MLP}(s_t;[128,128])$ $h_{\text{in}} = \text{Concat}[s_{t+1}, h_\tau, h_s]$ $\text{MLP}(h_{\text{in}};[256]) \to [\text{ResBlock}(256)]_{\times 4} \to \text{Linear}(d_s)$
Inverse dynamics model (Inv_ψ)	State s_t Next state s_{t+1}	$h_{\text{in}} = \text{Concat}[s_t, s_{t+1}]$ $\text{MLP}(h_{\text{in}};[256]) \to [\text{ResBlock}(256)]_{\times 3} \to \text{Linear}(d_a)$
Reward model ($\hat{R}_\phi$)	State s_t Next state s_{t+1}	$h_{\text{in}} = \text{Concat}[s_t, s_{t+1}]$ $\text{MLP}(h_{\text{in}};[256]) \to [\text{ResBlock}(256)]_{\times 3} \to \text{Linear}(1)$

Table 6: State dimension (d_s) and action dimension (d_a) of each agent.

Agent Type	State Dimension (d_s)	Action Dimension (d_a)
HalfCheetah, Walker2d	17	6
Hopper	11	3
Ant	111	8
Pen	45	24
Door	39	24

D Detailed Experimental Setup

This section outlines the experimental setup in detail. It first summarizes the baselines used for comparison in Appendix D.1, then describes the environment and offline dataset configurations and domain shifts considered in Appendix D.2, and finally presents the hyperparameter choices for model training and algorithm parameters in Appendix D.3.

D.1 Baselines Explanation

This section summarizes the baseline algorithms used for comparison with TCE approach.

IQL* (Kostrikov et al., 2022) trains IQL on the union of source and target datasets, providing a baseline for direct source-data utilization without explicit dynamics alignment. **Official Code:** `https://github.com/ikostrikov/implicit_q_learning`.

DARA (Liu et al., 2022) mitigates the effect of dynamics mismatch by training domain classifiers on state-action-next-state and state-action pairs to quantify domain discrepancy. This discrepancy is used to adjust source rewards, encouraging source data to better align with target dynamics. For this method, we follow the implementation provided in Lyu et al. (2025).

BOSA (Liu et al., 2024) introduces supported value estimation to constrain critic updates to plausible transitions under target dynamics. The actor updates only consider supported actions, preventing exploitation of unsupported out-of-distribution transitions. For this method, we follow the implementation provided in Lyu et al. (2025).

IGDF (Wen et al., 2024) learns cross-domain contrastive representations that distinguish source from target transitions. The resulting scores filter source data during critic training to ensure only reliable source transitions contribute. **Official Code:** `https://github.com/BattleWen/IGDF`.

OTDF (Lyu et al., 2025) applies optimal transport to align source and target transitions by computing deviation scores and selectively weighting source samples. The policy update also integrates CVAE-based support regularization to ensure the learned policy remains consistent with the target action. **Official Code:** `https://github.com/dmksjfl/OTDF`.

MOBODY (Guo et al., 2026) is a model-based off-dynamics offline RL method that learns target dynamics from source and limited target data, then performs policy optimization via rollouts from the learned target dynamics. It models source and target with separate action encoders and a shared latent transition, and uses target-Q-weighted behavior cloning for policy regularization. **Official Code:** `https://github.com/guoyihonggyh/MOBODY-Model-Based-Off-Dynamics-Offline-Reinforcement-Learning`.

D.2 Environmental Setup

This section details the experimental setup used to evaluate our approach. We adopt the cross-domain continuous control setup, including environments, dataset compositions, and domain shift configurations, proposed by Lyu et al. (2025). The benchmark is based on MuJoCo environments (Todorov et al., 2012) and features four agent types: HalfCheetah, Hopper, Walker2d, and Ant. In addition, we evaluate Adroit dexterous-manipulation tasks from the ODRL benchmark (Lyu et al., 2024b), covering higher-dimensional domains with more pronounced structural shifts.

Offline Datasets. The offline datasets consist of pre-collected data from both source and target domains, primarily drawn from the D4RL benchmark (Fu et al., 2020) for MuJoCo tasks and from the ODRL benchmark (Lyu et al., 2024b) for Adroit tasks.

Source Domain. For MuJoCo, source data comprises three quality levels: `medium`, `medium-replay`, and `medium-expert`. The `medium` dataset contains 1M samples generated by a partially trained SAC policy. The `medium-replay` dataset includes all samples in the SAC replay buffer up to the point of medium performance (approx. 0.2M to 0.4M samples), thus mixing low- to medium-quality experiences. Finally, the `medium-expert` datasets combine 50% expert and 50% suboptimal data, with total sizes of approximately 2M transitions. For Adroit, following ODRL, the source domain consists of expert-demonstration data with roughly 0.5M transitions per task.

Target Domain. The target domain is used to assess policy adaptation under distributional shifts. For MuJoCo, three types of dynamics changes are introduced to the agents: **Morphology** shift (modifying the agent's physical structure), **Kinematic** shift (restricting joint rotations to simulate malfunctions), and **Gravity** shift (altering gravitational acceleration). MuJoCo target datasets are limited to at most 5K transitions per dataset and are provided across `medium`, `medium-expert`, and `expert` quality levels. For Adroit, following ODRL, we evaluate two tasks, `Pen` and `Door`, under two target-domain shift types: **Morphology** shift (`shrink-finger`, reducing finger capsule sizes) and **Kinematic** shift (`broken-joint`, restricting finger joint ranges). Each target dataset is restricted to 5K transitions. The labels `M` and `H` denote medium and hard shift levels. Table 7 summarizes the dataset sizes by domain and quality.

Table 7: Dataset sizes by domain and data quality for MuJoCo and Adroit tasks.

Environment	Source Domain		Target Domain	Morphology	Kinematic	Gravity
HalfCheetah	`medium`	1M	`medium`	5K	5K	5K
	`medium-replay`	0.2M	`medium-expert`	5K	5K	5K
	`medium-expert`	2M	`expert`	5K	5K	5K
Hopper	`medium`	1M	`medium`	5K	5K	5K
	`medium-replay`	0.4M	`medium-expert`	4.3K	5K	4.3K
	`medium-expert`	2M	`expert`	5K	5K	5K
Walker2d	`medium`	1M	`medium`	5K	5K	5K
	`medium-replay`	0.3M	`medium-expert`	3.5K	4.4K	4.8K
	`medium-expert`	2M	`expert`	5K	5K	5K
Ant	`medium`	1M	`medium`	5K	5K	5K
	`medium-replay`	0.3M	`medium-expert`	5K	5K	3.1K
	`medium-expert`	2M	`expert`	5K	5K	5K
Pen	`expert`	0.5M	`expert`	5K	5K	–
Door	`expert`	0.5M	`expert`	5K	5K	–

Evaluation Metric. All results are presented as normalized scores to fairly compare across environments with varying return scales:

$$NS = \frac{J - J_r}{J_e - J_r} \times 100,$$

where J is the return of the evaluated policy, and J_r and J_e represent returns from random and expert policies in the target domain, respectively. Reference scores proposed by Lyu et al. (2025)

for each agent and target domain scenario are shown in Table 8. For Adroit, the morphology rows below correspond to the `shrink-finger` shift and the kinematic rows correspond to the `broken-joint` shift.

Table 8: Reference minimum score J_r and maximum score J_e by agent and domain shifts.

Agent Type	Domain Shifts	Reference Min Score J_r	Reference Max Score J_e
HalfCheetah	Morphology	-280.18	9713.59
	Kinematic	-280.18	7065.03
	Gravity	-280.18	9509.15
Hopper	Morphology	-26.34	3152.75
	Kinematic	-26.34	2842.73
	Gravity	-26.34	3234.3
Walker2d	Morphology	10.8	4398.43
	Kinematic	10.8	3257.51
	Gravity	10.8	5154.71
Ant	Morphology	-325.6	5722.01
	Kinematic	-325.6	5122.57
	Gravity	-325.6	4317.07
Pen	Morphology	-12.17	6408.38
	Kinematic	-12.17	6408.38
Door	Morphology	-52.34	2880.57
	Kinematic	-52.34	2880.57

Domain Shifts. To simulate realistic but significant distributional differences between source and target domains, we consider separate domain-shift settings for MuJoCo and Adroit. These shifts directly impact the agent's control dynamics and pose challenges to policy generalization. Fig. 5 provides visual examples of the source domain and the corresponding morphology-shifted target domains, highlighting the structural changes induced by these perturbations.

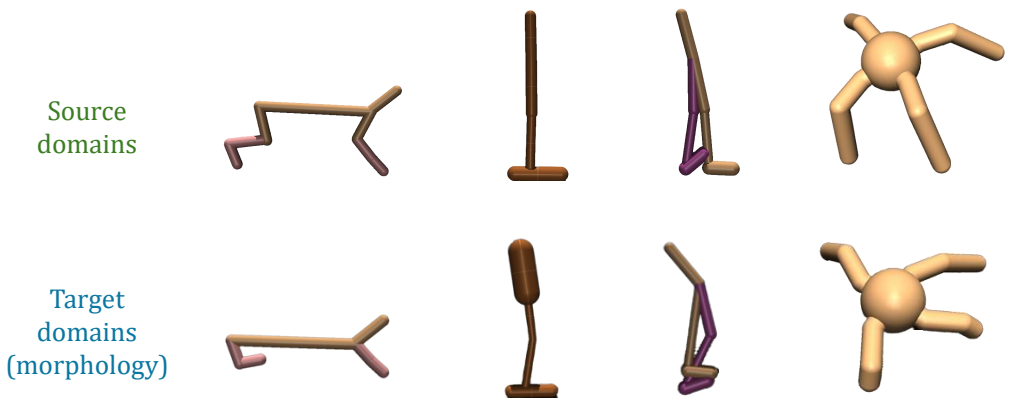


Figure 5: Visual examples of the source domain and morphology-shifted target domains in MuJoCo.

For MuJoCo tasks, we apply three types of domain shifts:

- *Morphology shift* involves structural modifications to the agent's body by altering body-segment dimensions. This includes shortening the thighs in HalfCheetah, shortening the torso in Hopper, shortening the thigh and elongating the lower leg in Walker2d, and reducing the front ankle capsules in Ant. These modifications create clear physical domain gaps that affect locomotion efficiency and balance.
- *Kinematic shift* emulates impairments or restrictions in joint mobility by narrowing the allowed range of joint rotations. This corresponds to restricting the back thigh in HalfCheetah, the thigh and foot joints in Hopper, the foot joint in Walker2d, and the front hip joints in Ant, thereby reducing maneuverability in the target domain.

- *Gravity shift* reduces the gravitational force magnitude acting on the agents by changing the target-domain gravity from the standard Earth value of -9.81 m/s^2 in the source to -4.905 m/s^2. This creates a lighter environment in which the reduced downward force alters the agents' balance, contact dynamics, and overall locomotion efficiency during movement.

For Adroit tasks, following ODRL, we consider two target-domain shift types:

- *Morphology shift* follows the `shrink-finger` setting, which reduces the capsule half-lengths of the finger phalanges, yielding shorter fingers with less contact area for stable grasping.
- *Kinematic shift* follows the `broken-joint` setting, which contracts the rotational ranges of the index-finger and thumb joints, substantially limiting finger dexterity and precise grasp formation.

These domain shifts are implemented by modifying the MuJoCo XML configuration files following Lyu et al. (2025) for MuJoCo and the ODRL XML and environment configurations for Adroit. Together, these settings enable a comprehensive evaluation of cross-domain offline reinforcement learning under varied and challenging environment changes.

HalfCheetah morphology shift : thigh length reduced by approximately 99%

```
<geom fromto="0 0 0 -0.0001 0 -0.0001" name="bthigh" size="0.046" type="
    capsule"/>
<body name="bshin" pos="-0.0001 0 -0.0001">
<geom fromto="0 0 0 0.0001 0 0.0001" name="fthigh" size="0.046" type="
    capsule"/>
<body name="fshin" pos="0.0001 0 0.0001">
```

Hopper morphology shift : torso length reduced by approximately 56%

```
<geom friction="0.9" fromto="0 0 1.45 0 0 1.05" name="torso_geom" size
    ="0.125" type="capsule"/>
```

Walker2d morphology shift : thigh shortened and lower leg elongated by approximately 98%

```
<body name="thigh" pos="0 0 1.05">
  <joint axis="0 -1 0" name="thigh_joint" pos="0 0 1.05" range="-150 0"
    type="hinge"/>
  <geom friction="0.9" fromto="0 0 1.05 0 0 1.045" name="thigh_geom" size
    ="0.05" type="capsule"/>
  <body name="leg" pos="0 0 0.35">
    <joint axis="0 -1 0" name="leg_joint" pos="0 0 1.045" range="-150 0"
    type="hinge"/>
    <geom friction="0.9" fromto="0 0 1.045 0 0 0.3" name="leg_geom" size
    ="0.04" type="capsule"/>
    <body name="foot" pos="0.2 0 0">
      <joint axis="0 -1 0" name="foot_joint" pos="0 0 0.3" range="-45 45"
     type="hinge"/>
      <geom friction="0.9" fromto="-0.0 0 0.3 0.2 0 0.3" name="foot_geom"
     size="0.06" type="capsule"/>
    </body>
  </body>
</body>
```

Ant morphology shift : front ankle capsule size reduced by 20%

```
<geom fromto="0.0 0.0 0.0 0.1 0.1 0.0" name="left_ankle_geom" size="0.08"
     type="capsule"/>
<geom fromto="0.0 0.0 0.0 -0.1 0.1 0.0" name="right_ankle_geom" size
    ="0.08" type="capsule"/>
```

HalfCheetah kinematic shift : back thigh joint range reduced by approximately 99%

```
<joint axis="0 1 0" damping="6" name="bthigh" pos="0 0 0" range="-.0052
    .0105" stiffness="240" type="hinge"/>
```

Hopper kinematic shift : thigh joint range reduced by 90%, foot joint range reduced by 60%

```
<joint axis="0 -1 0" name="thigh_joint" pos="0 0 1.05" range="-0.15 0"
    type="hinge"/>
<joint axis="0 -1 0" name="foot_joint" pos="0 0 0.1" range="-18 18" type
    ="hinge"/>
```

Walker2d kinematic shift : foot joint range reduced by 99%

```
<joint axis="0 -1 0" name="foot_joint" pos="0 0 0.1" range="-0.45 0.45"
    type="hinge"/>
```

Ant kinematic shift : front hip joint ranges reduced by approximately 89%

```
<joint axis="0 0 1" name="hip_1" pos="0.0 0.0 0.0" range="-0.3 0.3" type
    ="hinge"/>
<joint axis="0 0 1" name="hip_2" pos="0.0 0.0 0.0" range="-0.3 0.3" type
    ="hinge"/>
```

Gravity shift : gravity magnitude reduced by 50%

```
<option gravity="0 0 -4.905" timestep="0.01"/>
```

Adroit kinematic shift (medium) : restricted index finger and thumb joints by 75%

```
<joint name="FFJ3" pos="0 0 0" axis="0 1 0" range="-0.109 0.109" user
    ="1103"/>
<joint name="FFJ2" pos="0 0 0" axis="1 0 0" range="0 0.39275" user
    ="1102"/>
<joint name="THJ4" pos="0 0 0" axis="0 0 -1" range="-0.26175 0.26175"
    user="1121"/>
<joint name="THJ0" pos="0 0 0" axis="0 1 0" range="-0.3925 0" user
    ="1117"/>
```

Adroit kinematic shift (hard) : restricted index finger and thumb joints by 87.5%

```
<joint name="FFJ3" pos="0 0 0" axis="0 1 0" range="-0.0545 0.0545" user
    ="1103"/>
<joint name="FFJ2" pos="0 0 0" axis="1 0 0" range="0 0.196375" user
    ="1102"/>
<joint name="THJ4" pos="0 0 0" axis="0 0 -1" range="-0.130875 0.130875"
    user="1121"/>
<joint name="THJ0" pos="0 0 0" axis="0 1 0" range="-0.19625 0" user
    ="1117"/>
```

Adroit morphology shift (medium) : finger phalanx half-lengths reduced to 50%

```
<geom name="C_ffproximal" class="DC_Hand" size="0.01 0.005625" pos="0 0
    0.005625" type="capsule"/>
<geom name="C_ffmiddle"   class="DC_Hand" size="0.00805 0.003125" pos="0
    0 0.003125" type="capsule"/>
<geom name="C_ffdistal"   class="DC_Hand" size="0.00705 0.003" pos="0 0
    0.003" type="capsule" condim="4"/>
```

Adroit morphology shift (hard) : finger phalanx half-lengths reduced to 25%

```
<geom name="C_ffproximal" class="DC_Hand" size="0.01 0.0028125" pos="0 0
    0.0028125" type="capsule"/>
<geom name="C_ffmiddle"   class="DC_Hand" size="0.00805 0.0015625" pos="0
     0 0.0015625" type="capsule"/>
<geom name="C_ffdistal"   class="DC_Hand" size="0.00705 0.0015" pos="0 0
    0.0015" type="capsule" condim="4"/>
```

D.3 HYPERPARAMETER SETUP

Table 9 summarizes the shared settings for model training, data generation, and policy learning. We use $\eta = 0.2$ and $\beta = 0.001$ as the default coverage and policy regularization coefficients, respectively, including for Adroit tasks. For HalfCheetah and Walker2d under gravity shifts, we reduce η to 0.1 to limit expansion-induced generation error, as their target datasets often have lower quality or narrower distributions. For Walker2d under morphology shifts, we increase β to 0.5 to encourage stronger adherence to the target behavior policy under large structural gaps. Unless otherwise specified, TCE(Gen) and TCE(Mix) share the same environment-specific settings for η and β. The mixture coefficient λ is fixed at 0 for TCE(Gen). For TCE(Mix), we evaluate $\lambda \in \{0.1, 0.5, 0.9\}$ and report the best result for each experimental setting. Table 10 summarizes the resulting settings for η and β and the selected λ values for TCE(Mix).

Table 9: Shared hyperparameters for model training, sampling, and RL.

Category	Parameter Name	Value
Model Training	Learning rate	1e-4
	Optimizer	Adam
	Batch size	128
	Training epochs for q_θ^{cov}	10K
	Training epochs for q_θ^{tran}	5K
	Training epochs for Inv_ψ, $\hat{R}_\phi$	1K
	Noise schedule $\alpha_{\min}$	0.1
	Noise schedule $\alpha_{\max}$	20
Sampling	Denoising steps K	500
RL Training	Learning rate for actor, critic	3e-4
	Optimizer for actor, critic	Adam
	Batch size for target sample	128
	Batch size for generated sample	128
	Training steps	1M

Table 10: Hyperparameter settings used for the reported results.

Parameter	Environment	Value
η	HalfCheetah, Walker2d (gravity shift)	0.1
	Other environments	0.2
λ	HalfCheetah (gravity shift)	0.9
	Other environments	0.1
β	Walker2d (morphology shift)	0.5
	Other environments	0.001

E Additional Performance Comparison

This section complements the main-text evaluation with full performance comparisons under kinematic and gravity shifts in Appendices E.1 and E.2. In Appendix E.3, we further examine how the benefit of source reuse changes across a wider range of gravity shifts. We broaden the evaluation through comparisons with additional baselines in Appendix E.4.

E.1 Performance Comparison under Kinematic Shifts

Table 11 shows that both TCE variants outperform all baselines in aggregate, with TCE(Gen) achieving the highest total score. By using source states to expand the support of target-aligned generation, TCE can benefit from broader state coverage even when joint-range restrictions make source transitions poorly aligned with target dynamics. The overall advantage of TCE(Gen) is consistent with this distinction, while the gains from TCE(Mix) on some tasks suggest that additionally incorporating target-near source transitions can still complement the generated data. TCE(Mix) reports $\lambda = 0.1$, the best evaluated positive mixture coefficient for every task.

Table 11: Performance comparison on kinematic-shift tasks. Abbrev.: half=HalfCheetah, hopp=Hopper, walk=Walker2d, ant=Ant; m=`medium`, m-r=`medium-replay`, e=`expert`, m-e=`medium-expert`. Src./Tgt. denote source/target domains. Results are reported as mean $\pm$ standard deviation over 5 seeds; best results are in **bold** and second-best results are underlined.

Src.	Tgt.	IQL*	DARA	BOSA	IGDF	MOBODY	OTDF	TCE(Gen)	TCE(Mix)
half-m	m	12.3±1.2	10.6±1.2	8.3±1.2	23.6±5.7	35.4±4.8	40.2±0.0	**42.2±2.4**	41.6±1.8
half-m	m-e	10.8±1.9	12.9±2.8	8.7±1.3	9.8±2.4	19.5±3.2	10.1±4.0	**40.7±2.7**	40.1±2.0
half-m	e	12.6±1.7	12.1±1.0	10.8±1.7	12.8±0.7	3.9±1.6	8.7±2.0	**21.3±0.8**	6.7±2.1
half-m-r	m	10.0±5.4	11.5±4.9	7.5±3.1	11.6±4.6	14.5±4.9	37.8±2.1	**42.0±0.2**	41.7±1.8
half-m-r	m-e	6.5±3.1	9.2±4.7	6.6±1.7	8.6±2.3	4.1±0.7	9.7±2.0	**41.6±0.5**	37.8±2.8
half-m-r	e	13.6±1.4	14.8±2.0	10.4±4.9	13.9±2.2	4.1±0.9	7.2±1.4	**23.4±1.5**	4.3±0.3
half-m-e	m	21.8±6.5	25.9±7.4	30.0±4.3	21.9±6.5	30.2±5.3	30.7±9.6	**41.9±0.3**	41.7±0.1
half-m-e	m-e	7.6±1.4	9.5±4.2	6.8±2.9	8.9±3.3	7.4±3.3	10.9±4.2	**40.0±2.5**	39.8±3.3
half-m-e	e	9.1±2.4	10.4±1.3	4.9±3.2	10.7±1.4	3.3±1.2	3.2±0.6	**11.2±1.2**	4.3±0.3
hopp-m	m	58.7±8.4	43.9±15.2	12.3±6.6	65.3±1.4	63.3±3.7	65.6±1.9	**66.0±0.4**	65.7±0.5
hopp-m	m-e	68.5±12.4	55.4±16.9	15.6±10.8	51.1±18.5	25.4±12.5	55.4±25.1	**75.7±2.2**	70.3±10.0
hopp-m	e	79.9±35.5	83.7±19.6	14.8±5.5	87.4±25.4	84.2±11.2	35.0±19.4	**91.5±5.7**	76.0±15.5
hopp-m-r	m	36.0±0.1	39.4±7.2	3.2±2.6	35.9±2.4	44.1±5.3	35.5±12.2	65.9±1.4	**66.2±0.2**
hopp-m-r	m-e	36.1±0.1	34.1±3.6	4.4±2.8	36.1±0.1	36.1±4.8	47.5±14.6	**82.8±4.3**	65.1±7.9
hopp-m-r	e	36.0±0.1	36.1±0.2	3.7±2.5	36.1±0.3	**86.7±6.5**	49.9±30.5	84.8±7.0	72.3±16.1
hopp-m-e	m	66.0±0.5	61.1±4.0	35.0±20.1	65.2±1.5	66.1±0.8	65.3±2.4	**67.3±0.8**	66.0±0.1
hopp-m-e	m-e	45.1±15.7	61.9±16.9	13.9±4.9	62.9±15.6	41.8±14.6	38.6±15.9	**74.7±8.9**	72.1±6.8
hopp-m-e	e	44.9±19.8	84.2±21.1	12.0±4.3	52.8±39.7	**96.3±0.6**	29.9±11.3	65.0±7.5	62.9±31.9
walk-m	m	34.3±9.8	35.2±22.5	14.3±11.2	41.9±11.2	52.5±9.8	49.6±18.0	**55.3±2.0**	53.5±7.7
walk-m	m-e	30.2±12.5	**51.9±11.5**	13.6±7.7	42.3±19.3	26.4±11.3	43.5±16.4	35.9±10.2	18.1±10.2
walk-m	e	56.4±18.2	40.7±14.4	15.3±2.5	60.4±17.5	53.8±12.5	46.7±13.6	**62.8±1.7**	38.7±15.3
walk-m-r	m	11.5±7.1	12.5±4.3	1.9±2.1	22.2±5.2	34.5±7.5	49.7±9.7	**55.6±9.5**	43.1±4.8
walk-m-r	m-e	9.7±3.8	11.2±5.0	4.6±3.0	7.6±4.9	23.9±4.2	**55.9±17.1**	40.9±10.0	47.2±10.3
walk-m-r	e	7.7±4.8	7.4±2.4	3.6±1.5	7.5±2.1	27.2±4.5	**51.9±7.9**	49.9±7.4	38.5±9.3
walk-m-e	m	41.8±8.8	38.1±14.4	21.4±8.3	41.2±13.0	46.2±8.7	44.6±6.0	46.3±5.4	**47.1±10.9**
walk-m-e	m-e	22.2±8.7	23.6±8.1	15.9±4.1	28.1±4.0	17.2±3.7	16.5±7.2	33.6±4.3	**38.3±10.5**
walk-m-e	e	26.3±10.4	36.0±9.2	18.5±3.6	46.2±19.4	42.1±9.4	42.4±9.1	**54.6±4.4**	42.5±5.7
ant-m	m	50.0±5.6	42.3±7.6	20.9±2.6	54.5±1.3	51.5±8.7	**55.4±0.0**	55.0±3.7	54.9±5.7
ant-m	m-e	57.8±7.2	54.1±3.8	31.7±7.0	54.5±4.6	65.0±7.6	60.7±3.6	**65.1±6.4**	61.8±1.0
ant-m	e	59.6±18.5	54.2±11.3	45.4±8.6	49.4±14.6	80.9±6.8	90.4±4.8	93.9±4.0	**94.3±1.8**
ant-m-r	m	43.7±4.6	42.0±5.4	19.0±1.8	41.4±5.0	49.2±6.4	52.8±4.4	**55.6±9.1**	54.1±4.2
ant-m-r	m-e	36.5±5.9	36.0±6.7	19.1±1.6	37.2±4.7	53.7±5.9	54.2±5.2	**65.6±5.4**	61.1±1.0
ant-m-r	e	24.4±4.8	22.1±0.4	19.5±0.8	24.3±2.8	39.8±7.4	74.7±10.5	92.3±3.7	**93.4±4.5**
ant-m-e	m	49.5±4.1	44.7±4.3	19.0±8.0	41.8±8.8	46.5±4.5	50.2±4.3	**57.2±2.7**	53.1±4.6
ant-m-e	m-e	37.2±2.0	33.3±7.0	6.4±2.5	41.5±4.9	49.0±8.1	48.8±2.7	**69.0±1.8**	59.4±10.5
ant-m-e	e	18.7±8.1	17.8±23.6	14.5±9.0	14.4±22.9	63.6±5.5	78.4±12.2	**97.3±1.3**	93.7±3.0
Total Score		1193.0	1219.8	513.5	1271.0	1489.4	1547.6	**2063.9**	1867.4

E.2 Performance Comparison under Gravity Shifts

In Table 12, TCE(Mix) achieves the highest total score when target gravity is reduced to $0.5\times$ the standard value, although TCE(Gen) remains stronger on many individual tasks. TCE(Mix) reports $\lambda = 0.9$ for HalfCheetah and $\lambda = 0.1$ for the other agents, the best evaluated mixture coefficients in these settings. The substantial gains from source-heavy mixing in HalfCheetah suggest that selected source transitions can remain sufficiently aligned with target dynamics to benefit policy learning despite the change in gravity. This interpretation aligns with Theorem 4.1: a mild source–target dynamics gap limits the source-induced contribution to the performance-gap bound, supporting the use of target-near source transitions to complement target-aligned generation.

Table 12: Performance comparison on gravity-shift tasks. Abbrev.: half=HalfCheetah, hopp=Hopper, walk=Walker2d, ant=Ant; m=`medium`, m-r=`medium-replay`, e=`expert`, m-e=`medium-expert`. Src./Tgt. denote source/target domains. Results are reported as mean ± standard deviation over 5 seeds; best results are in **bold** and second-best results are underlined.

Src.	Tgt.	IQL*	DARA	BOSA	IGDF	MOBODY	OTDF	TCE(Gen)	TCE(Mix)
half-m	m	39.6±3.3	**41.2±3.9**	38.9±4.0	36.6±5.5	38.4±5.4	40.7±7.7	4.1±0.4	40.7±1.2
half-m	m-e	39.6±3.7	**40.7±2.8**	40.4±3.0	38.7±6.2	35.4±4.6	28.6±3.2	3.1±0.3	39.2±0.5
half-m	e	**42.4±3.8**	39.8±4.4	40.5±3.9	39.6±4.6	37.1±5.5	36.1±5.3	14.8±3.9	41.1±6.1
half-m-r	m	20.1±5.0	17.6±6.2	20.0±4.9	14.4±2.2	17.9±6.8	**21.5±6.5**	4.0±1.8	18.1±7.2
half-m-r	m-e	17.2±1.6	20.2±5.2	16.7±4.2	10.0±2.5	16.8±5.3	14.7±4.1	2.9±0.1	**20.3±5.1**
half-m-r	e	20.7±5.5	**22.4±1.7**	15.4±4.2	15.3±3.7	19.5±4.5	11.4±1.9	7.9±1.8	15.5±1.3
half-m-e	m	38.6±6.0	37.8±3.3	**41.8±5.1**	37.7±7.3	38.2±7.6	39.5±3.5	2.5±0.9	39.4±2.7
half-m-e	m-e	39.6±3.0	39.4±4.4	38.7±2.7	40.7±3.2	34.4±7.8	32.4±5.5	2.8±0.1	**41.3±4.5**
half-m-e	e	43.4±0.9	**45.3±1.3**	39.9±2.7	41.1±4.1	40.2±3.4	26.5±9.1	8.5±2.6	41.7±1.4
hopp-m	m	11.2±1.1	17.3±3.8	15.2±3.3	15.3±3.5	20.1±4.5	32.4±8.0	**61.9±2.7**	52.5±4.7
hopp-m	m-e	14.7±3.6	15.4±2.5	21.1±9.3	15.1±3.6	15.6±3.3	24.2±3.6	**46.2±2.6**	35.0±7.9
hopp-m	e	12.5±1.6	19.3±10.5	12.7±1.7	14.4±0.8	14.3±3.2	33.7±7.8	**53.4±2.2**	18.9±10.4
hopp-m-r	m	13.9±2.9	10.7±4.3	3.3±1.9	15.3±4.4	20.3±9.9	31.1±13.4	**60.7±2.7**	58.3±0.8
hopp-m-r	m-e	13.3±6.3	12.5±5.6	4.6±1.7	15.4±5.5	19.5±4.2	24.2±6.1	24.2±3.1	**32.7±10.4**
hopp-m-r	e	11.0±2.6	14.3±6.0	3.2±0.8	16.1±4.0	17.5±2.8	31.0±9.8	**73.4±7.8**	55.7±15.0
hopp-m-e	m	19.1±6.6	18.5±12.3	15.9±5.9	22.3±5.4	19.2±4.8	26.4±10.1	**61.3±5.6**	54.0±5.8
hopp-m-e	m-e	16.8±2.7	16.0±6.1	17.3±2.5	16.6±7.7	17.9±3.5	28.3±6.7	**45.5±9.8**	33.4±7.2
hopp-m-e	e	20.9±4.1	23.9±14.8	23.2±7.9	26.0±9.2	24.6±4.7	44.9±10.6	**66.3±5.8**	55.8±8.3
walk-m	m	28.1±12.9	28.4±13.7	**38.0±11.2**	22.1±8.4	36.6±3.8	36.6±2.3	36.3±0.9	36.6±4.1
walk-m	m-e	35.7±4.7	30.7±9.7	40.9±7.2	35.4±9.1	42.2±6.8	**44.8±7.5**	20.6±6.4	40.9±6.0
walk-m	e	37.3±8.0	36.0±7.0	41.3±8.6	36.2±13.6	**49.9±7.9**	44.0±4.0	20.3±1.6	46.9±1.6
walk-m-r	m	14.6±2.5	14.1±6.1	7.6±5.8	11.6±4.6	20.8±6.5	32.7±7.0	**35.8±0.8**	20.3±2.8
walk-m-r	m-e	15.3±1.9	15.9±5.8	4.8±5.8	13.9±6.5	20.1±9.6	**31.6±6.1**	27.4±4.7	18.2±1.6
walk-m-r	e	15.8±7.2	15.7±4.5	7.1±4.6	15.2±5.3	29.2±7.4	**31.3±5.3**	21.5±2.9	15.6±3.0
walk-m-e	m	39.9±13.1	41.6±13.0	32.3±7.2	33.8±3.1	39.9±3.1	30.2±9.8	33.0±2.1	**43.2±2.6**
walk-m-e	m-e	49.1±6.9	45.8±9.4	40.1±4.5	44.7±2.9	52.8±7.5	**53.3±7.1**	28.6±2.2	45.6±1.7
walk-m-e	e	40.4±11.9	56.4±3.5	43.7±4.4	45.3±10.4	52.9±10.1	61.1±3.4	31.6±4.9	**65.2±10.7**
ant-m	m	10.2±1.8	9.4±0.9	12.4±2.0	11.3±1.3	11.7±2.6	45.1±12.4	**57.0±6.1**	56.4±5.1
ant-m	m-e	9.4±1.2	10.0±0.9	11.6±1.3	9.4±1.4	10.7±2.3	33.9±5.4	**52.9±5.5**	50.7±9.1
ant-m	e	10.2±0.3	9.8±0.6	11.8±0.4	9.7±1.6	11.2±3.8	33.2±9.0	54.1±1.9	**55.7±6.4**
ant-m-r	m	18.9±2.6	21.7±2.1	13.9±1.5	19.6±1.0	19.1±3.5	29.6±10.7	53.3±12.0	**57.2±8.2**
ant-m-r	m-e	19.1±3.0	18.3±2.1	15.9±2.7	20.3±1.6	18.3±4.1	25.4±2.1	42.4±10.9	**44.0±11.6**
ant-m-r	e	18.5±0.9	20.0±1.3	14.5±1.7	18.8±2.1	18.5±2.6	24.5±2.8	**50.1±7.9**	40.5±13.9
ant-m-e	m	9.8±2.4	8.1±1.8	8.1±3.0	8.9±1.5	10.1±3.5	18.6±11.9	**58.0±11.7**	53.8±10.4
ant-m-e	m-e	9.0±0.8	6.4±1.4	6.2±1.5	7.2±2.9	9.3±2.1	34.0±9.4	**46.0±10.3**	43.8±6.1
ant-m-e	e	9.1±2.6	10.4±2.9	4.2±3.9	9.2±1.5	9.8±1.8	23.2±2.9	**58.0±3.7**	45.7±18.9
Total Score		825.0	851.0	763.2	803.2	910.0	1160.7	1270.4	**1473.9**

E.3 Performance Comparison across Gravity Shifts

To examine how this benefit of source reuse varies with the magnitude of the gravity shift, we extend the HalfCheetah evaluation across eight target gravity levels ranging from 0.7 to 4.0 times the standard value (Table 13). Source gravity remains unchanged, and the coverage coefficient is fixed at $\eta = 0.2$. We use `halfcheetah-medium-expert-v2` as source data and construct expert target datasets with approximately 5K transitions collected in each target environment. All gravity variants use the standard D4RL HalfCheetah normalization references. As in the main-text comparisons, TCE(Mix) reports the best mean score over $\lambda \in \{0.1, 0.5, 0.9\}$ for each environment.

TCE(Mix) generally outperforms TCE(Gen) under mild gravity changes of 0.7–1.3$\times$ the standard value, whereas TCE(Gen) achieves higher mean scores at all evaluated gravity levels of 1.5$\times$ the standard value or above. Among the evaluated mixing configurations, stronger gravity changes favor smaller source fractions. Together, these observations suggest that the benefit of additionally incorporating target-near source transitions diminishes as the gravity shift grows, favoring the use of source states to expand generation support rather than the inclusion of their transitions. This change in the relative benefit of the two forms of source utilization is consistent with the source–generation trade-off in Theorem 4.1.

Table 13: Performance comparison across HalfCheetah gravity shifts with approximately 5K target transitions. The best model is in **bold** and the second-best is underlined.

Gravity ($\times$)	IQL*	DARA	BOSA	IGDF	MOBODY	OTDF	TCE(Gen)	TCE(Mix)
0.7	39.5±1.1	36.9±0.7	36.0±0.7	34.8±1.4	38.4±0.6	40.4±0.7	44.3±4.1	**46.3±0.2**
0.9	42.5±0.5	43.3±0.9	43.3±0.3	44.7±1.4	45.5±1.9	42.0±0.2	41.9±1.9	**49.4±1.8**
1.1	50.5±1.1	67.0±2.1	41.1±0.3	54.7±7.3	53.2±1.5	51.0±0.2	90.8±1.0	**95.9±1.5**
1.3	38.1±1.1	51.0±2.6	39.2±1.1	48.2±1.8	39.9±0.1	38.6±0.6	56.8±0.5	**58.0±0.0**
1.5	32.7±1.6	37.3±1.0	37.5±0.2	36.2±1.4	35.5±0.7	33.9±0.7	**45.9±0.1**	45.6±0.1
2.0	20.3±0.1	20.8±0.2	22.4±0.7	21.2±1.5	22.6±0.2	20.6±1.7	**36.6±0.2**	30.2±0.8
3.0	2.8±1.4	3.4±1.8	2.8±1.8	8.4±1.2	5.1±1.3	3.7±0.8	**42.1±0.2**	37.8±0.4
4.0	1.0±0.5	0.5±0.0	2.1±0.6	0.6±0.4	0.9±0.7	1.4±0.5	**38.2±0.1**	22.0±1.0

E.4 Comparison with Additional Baselines

We additionally compare TCE(Gen) against Meta-DT* (Wang et al., 2024) and DmC (Le Pham Van et al., 2025). Meta-DT* is a representative meta-offline RL approach adapted for our setting by training on the union of source and target datasets, while DmC is a recent diffusion-based method that generates additional source-domain data. Table 14 shows that Meta-DT* scores below TCE(Gen) on all evaluated tasks and below IQL* on many tasks. This comparison indicates that the pooled-data Meta-DT* adaptation is less effective than TCE(Gen) in this cross-domain setting. Meanwhile, TCE(Gen) outperforms DmC on a majority of tasks and achieves a higher total score, although DmC remains stronger on several tasks. This pattern is consistent with the potential benefit of generating target-aligned transitions rather than additional source-domain data. We note that since the official code for DmC is unavailable, we report the performance directly from the original paper.

Table 14: Performance comparison of TCE(Gen) with Meta-DT* and DmC under kinematic shifts.

Src.	Tgt.	IQL*	Meta-DT*	DmC	OTDF	TCE(Gen)
half-m	m	12.3±1.2	13.4±3.4	38.5±1.4	40.2±0.0	**42.2±2.4**
half-m	m-e	10.8±1.9	8.5±0.6	19.1±1.0	10.1±4.0	**40.7±2.7**
half-m	e	12.6±1.7	5.0±0.1	13.1±0.8	8.7±2.0	**21.3±0.8**
half-m-r	m	10.0±5.4	5.5±0.7	19.5±1.8	37.8±2.1	**42.0±0.2**
half-m-r	m-e	6.5±3.1	7.5±1.1	11.4±2.1	9.7±2.0	**41.6±0.5**
half-m-r	e	13.6±1.4	6.4±2.8	15.6±2.9	7.2±1.4	**23.4±1.5**
half-m-e	m	21.8±6.5	4.7±0.2	38.4±1.4	30.7±9.6	**41.9±0.3**
half-m-e	m-e	7.6±1.4	7.7±3.2	24.1±4.6	10.9±4.2	**40.0±2.5**
half-m-e	e	9.1±2.4	2.8±0.2	**13.4±2.0**	3.2±0.6	11.2±1.2
hopp-m	m	58.7±8.4	5.0±0.2	**69.8±2.3**	65.6±1.9	66.0±0.4
hopp-m	m-e	68.5±12.4	4.9±0.2	**78.2±5.1**	55.4±25.1	75.7±2.2
hopp-m	e	79.9±35.5	5.2±0.2	59.8±21.8	35.0±19.4	**91.5±5.7**
hopp-m-r	m	36.0±0.1	36.4±0.2	64.8±2.4	35.5±12.2	**65.9±1.4**
hopp-m-r	m-e	36.1±0.1	36.3±0.1	69.7±7.5	47.5±14.6	**82.8±4.3**
hopp-m-r	e	36.0±0.1	37.6±0.1	69.9±18.0	49.9±30.5	**84.8±7.0**
hopp-m-e	m	66.0±0.5	3.5±0.4	**69.6±1.3**	65.3±2.4	67.3±0.8
hopp-m-e	m-e	45.1±15.7	8.5±2.3	**75.5±9.6**	38.6±15.9	74.7±8.9
hopp-m-e	e	44.9±19.8	6.4±0.2	64.5±24.2	29.9±11.3	**65.0±7.5**
walk-m	m	34.3±9.8	5.0±0.3	**63.2±4.2**	49.6±18.0	55.3±2.0
walk-m	m-e	30.2±12.5	15.7±2.0	**53.5±7.0**	43.5±16.4	35.9±10.2
walk-m	e	56.4±18.2	10.0±0.9	**70.5±12.0**	46.7±13.6	62.8±1.7
walk-m-r	m	11.5±7.1	3.4±1.0	52.9±8.4	49.7±9.7	**55.6±9.5**
walk-m-r	m-e	9.7±3.8	14.6±0.1	36.4±5.4	**55.9±17.1**	40.9±10.0
walk-m-r	e	7.7±4.8	8.9±1.0	44.4±8.5	**51.9±7.9**	49.9±7.4
walk-m-e	m	41.8±8.8	8.5±0.8	**59.4±6.8**	44.6±6.0	46.3±5.4
walk-m-e	m-e	22.2±8.7	10.2±8.7	**53.2±7.3**	16.5±7.2	33.6±4.3
walk-m-e	e	26.3±10.4	5.7±2.6	**69.2±7.0**	42.4±9.1	54.6±4.4
ant-m	m	50.0±5.6	13.9±0.7	**62.1±0.6**	55.4±0.0	55.0±3.7
ant-m	m-e	57.8±7.2	14.8±0.3	**68.9±1.0**	60.7±3.6	65.1±6.4
ant-m	e	59.6±18.5	14.9±0.1	92.1±3.5	90.4±4.8	**93.9±4.0**
ant-m-r	m	43.7±4.6	22.3±0.2	**61.9±0.5**	52.8±4.4	55.6±9.1
ant-m-r	m-e	36.5±5.9	21.0±0.1	58.8±3.6	54.2±5.2	**65.6±5.4**
ant-m-r	e	24.4±4.8	26.8±2.2	43.8±2.6	74.7±10.5	**92.3±3.7**
ant-m-e	m	49.5±4.1	13.5±0.1	**60.6±1.3**	50.2±4.3	57.2±2.7
ant-m-e	m-e	37.2±2.0	13.6±0.1	60.4±3.7	48.8±2.7	**69.0±1.8**
ant-m-e	e	18.7±8.1	18.1±1.2	76.0±4.1	78.4±12.2	**97.3±1.3**
Total Score		1193.0	446.2	1902.2	1547.6	**2063.9**

F Sample Fidelity Analysis

This section complements the main coverage–error analysis in Section 5.3 with quantitative fidelity trends on the HalfCheetah and Hopper morphology-shift (`medium-to-expert`) tasks and a qualitative target-simulator visualization.

F.1 Quantitative Error Analysis

Coverage expansion increases generation error to different degrees across environments. We analyze the fidelity of generated samples on the HalfCheetah and Hopper morphology-shift tasks (`medium-to-expert`) with respect to the coverage coefficient η. Following the definitions in Section 5.3, we report three metrics: action reconstruction error measured using a reference target inverse-dynamics model, and reward prediction and transition dynamics errors measured using the true target simulator.

HalfCheetah. HalfCheetah exhibits increasing generation error as coverage expands. Fig. 6(a)–(c) show that action, reward, and transition errors increase as η expands state coverage. Together with the main performance analysis, this trend supports balancing broader coverage against increased generation error rather than minimizing error alone.

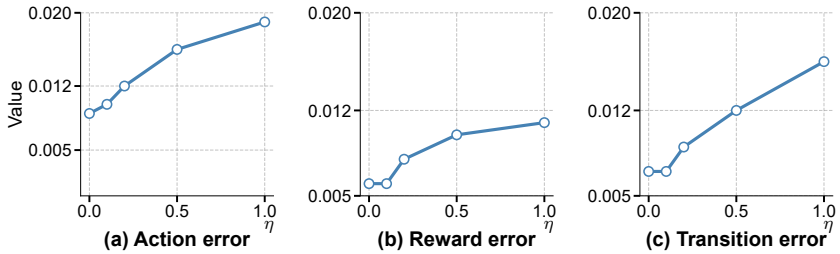


Figure 6: Sample-fidelity errors with respect to η in HalfCheetah morphology shifts.

Hopper. Hopper exhibits a sharp increase in action error under aggressive coverage expansion. In particular, Fig. 7(a) shows a sharp rise in action reconstruction error at larger η values, accompanied by increases in reward and transition errors in Fig. 7(b)–(c). This sensitivity highlights the need to limit expansion when generation error grows rapidly, rather than applying the same expansion level across environments.

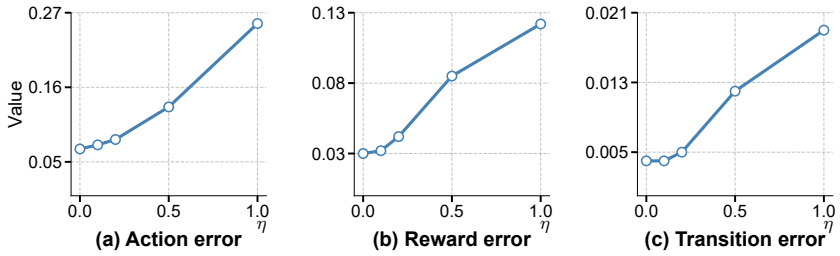


Figure 7: Sample-fidelity errors with respect to η in Hopper morphology shifts.

F.2 Qualitative Simulator Visualization

In the two HalfCheetah morphology-shift examples, TCE's generated next states closely match the target-simulator outcomes. For each target-near source state s_t, we project the state into the target simulator by initializing it with the same state variables. TCE generates $\hat{s}_{t+1}$ and infers the

corresponding action $\hat{a}_t$; we then execute this action for one target-environment step and visually compare the simulator rollout with the generated next state. The agreement in Fig. 8 provides qualitative evidence of one-step target-dynamics consistency for these examples, complementing the simulator-based transition error in Table 3. The target simulator is used only for this evaluation.

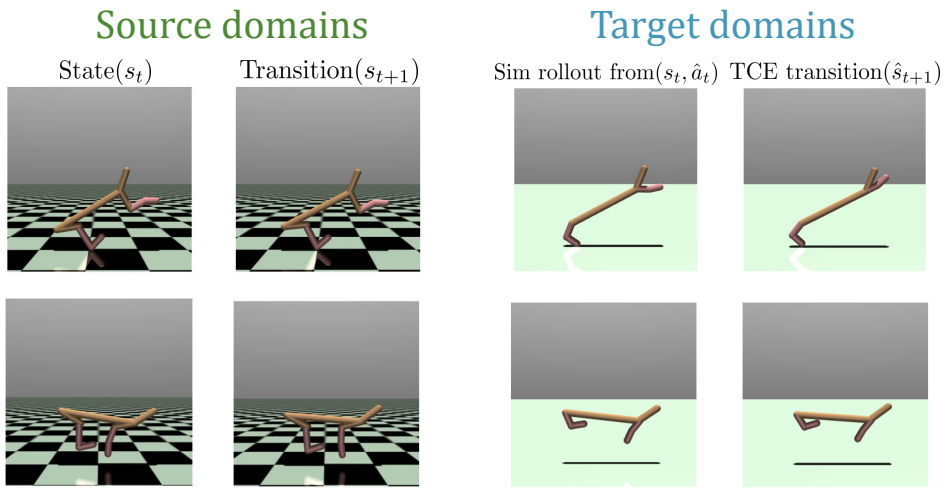


Figure 8: Qualitative comparison on the HalfCheetah morphology shift. Each row shows a target-near source state s_t and its source-domain next state s_{t+1}, followed by the target-simulator rollout obtained after projecting s_t into the target domain and executing $\hat{a}_t$, and TCE's generated next state $\hat{s}_{t+1}$.

G Computational Complexity

TCE(Gen) adds generation-related computation while maintaining peak GPU memory usage comparable to the baselines. This section compares the computational resources required by IQL* (Kostrikov et al., 2022), OTDF (Lyu et al., 2025), and TCE(Gen) over the 36 morphology-shift tasks. Experiments were conducted on a server with AMD EPYC 7513 CPUs and NVIDIA RTX 3090 GPUs. While all methods require approximately three hours for the offline RL phase, OTDF adds roughly 18 minutes for data selection. TCE(Gen) incurs approximately 48 minutes for score network training and 16 minutes for sampling, averaged over the evaluated tasks. These phases and offline RL run sequentially with intermediate memory deallocation, keeping the reported peak GPU memory usage at roughly 2 GB. Table 15 therefore shows additional preprocessing time with comparable peak memory usage, in exchange for the performance improvements reported in the main text.

Table 15: Comprehensive runtime and memory usage comparison averaged over 36 morphology-shift tasks. Note that GPU memory usage represents the peak allocation during the entire pipeline.

Method	Model Training	Data Sampling	Offline RL	Peak GPU Memory
IQL*	–	–	3 h	2 GB
OTDF	–	18 min	3 h	2 GB
TCE(Gen)	48 min	16 min	3 h	2 GB

H Additional Analysis

This section presents detailed analyses not covered in the main text. Appendix H.1 examines coverage expansion and source–generation mixing. Appendix H.2 analyzes the effect of target-data size on inverse-dynamics error and policy performance. Appendix H.3 analyzes sensitivity to the denoising step K.

H.1 Ablation Studies on Coverage Expansion and Source–Generation Mixing

Coverage Coefficient η. Moderate coverage expansion generally improves performance over Simple Aug., while larger η can reduce the gain. Fig. 9 illustrates the performance trends across different values of η for TCE(Gen), with $\lambda = 0$ fixed, under morphology, kinematic, and gravity shifts on the Ant task (`medium-replay-to-medium-expert`). As shown, the performance remains relatively stable across a wide range, generally outperforming the simple augmentation baseline. However, we observe a slight performance degradation as η increases to 0.5. This is consistent with the coverage–error trade-off in Section 5.3: expanding support can improve learning, but excessive expansion increases generation error. Accordingly, the choice of η should account for both coverage gains and the generation-error sensitivity of each setting.

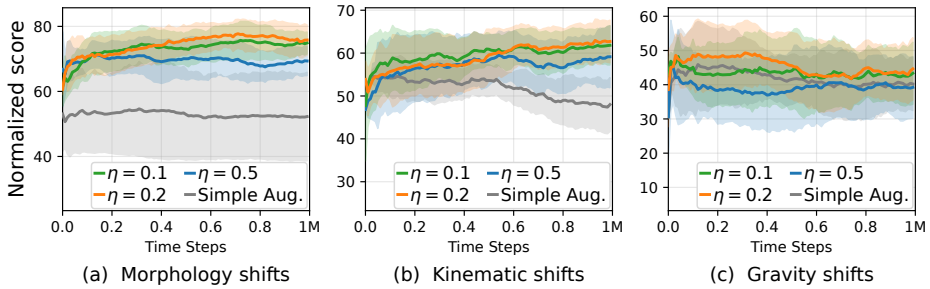


Figure 9: Performance sensitivity to η across different domain shifts on the Ant `medium-replay-to-medium-expert` task.

Mixture Coefficient λ. The benefit of source reuse depends on the dynamics gap of selected source transitions relative to generation error. Fig. 10 analyzes sensitivity to the source–generation mixture by varying λ with $\eta = 0.2$ fixed, with $\lambda = 0$ corresponding to TCE(Gen) and $\lambda > 0$ to

TCE(Mix). Increasing λ replaces generated transitions with NN-selected real source transitions at a fixed augmentation budget. As suggested by Theorem 4.1, source reuse can complement generation when the selected transitions are sufficiently target-aligned, whereas a larger dynamics gap can favor generation.

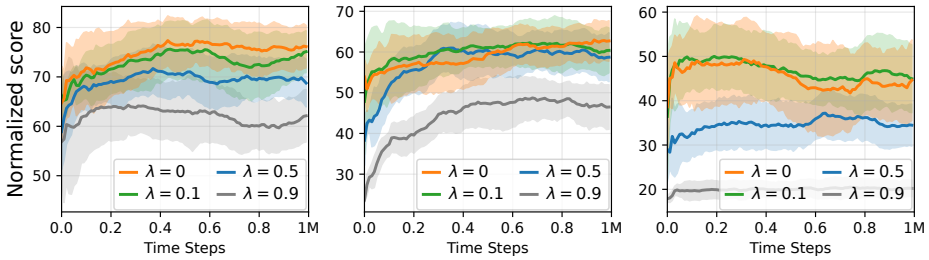


Figure 10: Sensitivity to the mixture coefficient λ across different domain shifts on the Ant `medium-replay-to-medium-expert` task.

H.2 Effect of Target Data Size on Inverse Dynamics Error

Increasing target-data size reduces inverse-dynamics error and improves policy performance, with smaller score gains beyond 5K transitions. We vary the target-data size from 2K to 20K transitions and report the resulting action prediction error and normalized score in Table 16. Reducing the data from 5K to 2K increases action error from 0.065 to 0.142 and lowers the score from 94.0 to 90.3. Increasing the data to 20K further reduces the error to 0.015, while the score rises to 95.0, indicating diminishing performance gains despite continued error reduction.

Table 16: Action prediction error and normalized score under different target-data sizes.

Metric	2K	5K (ours)	10K	20K
Action error	0.142±0.018	0.065±0.012	0.042±0.009	**0.015±0.006**
Score	90.3±4.2	94.0±4.5	94.4±3.8	**95.0±3.1**

H.3 Ablation Study on Denoising Step K

TCE(Gen) shows modest changes in average performance across the tested denoising-step counts.The denoising-step hyperparameter K governs the trade-off between sampling cost and generation quality through the discretization granularity of the reverse process. Table 17 presents results on the Ant morphology-shift environment for $K \in \{100, 200, 500\}$, with $K = 500$ as the default. While $K = 500$ achieves the highest total score, $K = 100$ and $K = 200$ remain competitive on average, although individual tasks differ in sensitivity.

Table 17: Performance of TCE(Gen) on Ant morphology-shift tasks at different K values.

Src.	Tgt.	$K = 100$	$K = 200$	$K = 500$
ant-m	m	**42.2±1.8**	42.0±0.3	41.5±0.8
ant-m	m-e	72.5±2.3	**74.1±0.5**	72.7±2.5
ant-m	e	91.9±0.1	92.8±2.8	**94.0±4.5**
ant-m-r	m	40.7±1.2	40.8±0.7	**41.5±1.5**
ant-m-r	m-e	70.0±2.8	69.2±3.1	**71.0±8.4**
ant-m-r	e	90.8±1.6	91.3±3.8	**96.1±0.3**
ant-m-e	m	41.4±0.9	41.2±0.6	**41.9±0.6**
ant-m-e	m-e	**66.5±3.1**	66.1±1.6	66.4±1.2
ant-m-e	e	93.7±0.5	94.0±1.1	**96.6±0.5**
Total Score		609.7	611.5	**621.7**