IMITATION FROM OBSERVATIONS WITH TRAJECTORY-LEVEL GENERATIVE EMBEDDINGS

Yongtao Qu, Shangzhe Li, Weitong Zhang
University of North Carolina, Chapel Hill
Chapel Hill, NC 27514, USA
{yongtao,shangzhe,weitongz}@unc.edu

ABSTRACT

We consider the offline imitation learning from observations (LfO), where expert demonstrations are scarce and contain only state observations, and the suboptimal policy is far from expert behavior. In this regime, many existing imitation learning approaches struggle to extract useful information from imperfect data since they impose strict support constraints and rely on brittle one-step models. To tackle this challenge, we propose **T**rajectory-level **G**enerative **E**mbedding (TGE) for offline LfO. TGE constructs a dense, smooth surrogate reward by using particle based entropy estimation to maximize the log-likelihood of expert trajectories in the latent space of a temporal diffusion model trained on offline suboptimal data. By leveraging the structured geometry of the learned diffusion embedding, TGE captures long-horizon temporal dynamics and effectively bridges the gap under severe support mismatch, ensuring a robust learning signal even when offline data is distributionally distinct from the expert. Empirically, the proposed approach consistently matches or outperforms prior offline LfO methods across a range of D4RL locomotion and manipulation benchmarks.

1 INTRODUCTION

Imitation learning (IL) has demonstrated remarkable success in enabling agents to acquire complex behaviors across various domains, from robotic manipulation (Florence et al., 2021; Chi et al., 2023; Brohan et al., 2023) to autonomous navigation (Codevilla et al., 2018; Bansal et al., 2019). While standard IL methods can effectively recover high-performance policies by supervising explicit action labels, applying these techniques to real-world scenarios poses substantial challenges. A primary bottleneck is the absence of explicit action information in many natural data sources, such as video demonstrations or third-person recordings, which precludes direct supervision of state-action mappings (Torabi et al., 2018; Stadie et al., 2017; Liu et al., 2018). This has led to the emergence of Learning from Observations (LfO), where the agent must infer desirable behaviors from state-only expert trajectories.

To recover the underlying actions, many of the existing LfO methods attempt to estimate the inverse dynamics or auxiliary inference mechanisms (Hanna & Stone, 2017; Kidambi et al., 2020). However, an equally critical challenge lies in the quality of the offline data itself. As large-scale offline datasets inevitably contain suboptimal or inconsistent behaviors arising from human error or task ambiguity, directly applying behavioral cloning to such datasets often leads to poor performance. Effectively leveraging these imperfect demonstrations therefore requires algorithms that can robustly filter noise and distill informative behavioral priors, without being misled by non-expert trajectories.

In addition to the challenge in the offline data quality, current LfO approaches confront structural weaknesses that limit their robustness despite their progress. To mention a few, surrogate reward approaches, such as ORIL (Zolna et al., 2020), rely on discriminator or inverse dynamics models to annotate offline data. However, their performance can deteriorate over long horizons when the learned signal is imperfect (Ross et al., 2011). On the other hand, state-of-the-art distribution-matching approaches rely on estimating density ratios between expert and offline distributions (Ho & Ermon, 2016; Ma et al., 2022). However, these methods typically assume the expert policy

is well-covered by the offline data. When the offline behavioral data have limited support to the expert, strictly aligning them yields sparse or undefined signals and therefore fails to guide the agent towards the expert behavior. We review these paradigms in more detail in Sec. 2.1. These limitations raise a central question for learning from observations:

How can we extract a reliable imitation signal from imperfect offline data under support mismatch?

To address these limitations, we propose **Trajectory-level Generative Embeddings (TGE)**, a novel framework for offline LfO that bypasses the structural reliance on support coverage inherent in density ratio estimation. Instead of relying on unstable density ratios, TGE leverages the smooth representation learned by a trajectory-based diffusion model such as diffuser (Janner et al., 2022). By encoding trajectories into this generative latent space, we construct a dense surrogate reward signal that bridges regions of sparse support. Specifically, we employ a logarithmic distance kernel with heavy-tailed decay for reward estimation inspired by the particle-based entropy estimation (Liu & Abbeel, 2021; Mutti et al., 2020; Singh et al., 2003), which ensures that maximizing the surrogate reward using standard offline RL algorithms, such as IQL (Kostrikov et al., 2021) or ReBRAC (Tarasov et al., 2023) is essentially maximizing the log-likelihood of the expert policy. Notably, this design of the surrogate reward also ensures an informative signal throughout the suboptimal data, preventing the reward collapse issues common in prior distance-based methods.

Our main contributions are summarized as follows:

- We introduce a trajectory-level representation learning approach based on a generative planner. We demonstrate that the resulting latent embeddings are naturally separable and can distinguish between suboptimal and expert-like data in offline datasets, without requiring explicit discriminative training.
- We propose a kernel-based reward formulation defined over the trajectory-level latent embedding, which enables the use of standard offline RL methods for imitation learning with observation-only expert data.
- We empirically evaluate our method across a diverse set of benchmarks. The results show that our approach consistently matches or outperforms state-of-the-art offline learning-from-observations baselines, with particularly strong robustness in regimes where expert coverage in the offline dataset is limited.

2 Related Work

2.1 Offline Imitation Learning

Offline imitation learning (IL) aims to recover an expert policy from a fixed dataset of demonstrations without access to online environment interactions. While the foundational approach, Behavioral Cloning (BC) (Pomerleau, 1989), treats this as a supervised regression problem, it notoriously suffers from distribution shift and compounding errors when the agent accesses out-of-distribution states (Ross et al., 2011). To mitigate these compounding errors, distribution-matching approaches shift the objective from local action supervision to global distribution alignment, minimizing the divergence between the learner and expert occupancy measures (Kostrikov et al., 2020; Garg et al., 2021). Beyond improving the training objective, recent advances have emphasized policy expressivity, shifting toward conditional generative modeling to better capture the multi-modal nature of the expert's behavior. Prominent methods now leverage Energy-Based Models (Florence et al., 2021), Transformers (Shafiullah et al., 2022), and Diffusion Models (Chi et al., 2023) to represent complex, non-Gaussian action distributions, thereby enabling robust policy synthesis even in high-dimensional and multi-modal settings.

Learning from Observations (LfO) arises as a critical and more challenging subset of this paradigm, where the expert demonstrations lack action labels. Offline LfO approaches generally fall into two distinct paradigms based on how they utilize expert data which we provide an overview below:

Occupancy Matching. This class of methods formulates imitation as a optimization problem, solving for the state-occupancy ratio $w(s) = d^{\pi_E}(s)/d^{\pi}(s)$ between the expert and offline suboptimal data distributions. Methods like SMODICE (Ma et al., 2022) and LobsDICE (Kim et al., 2022), which minimize the KL-divergence between these distributions, are often sensitive to the distribution of the offline buffer. Recent variants like PW-DICE (Yan et al., 2024) introduce Wasserstein

geometry to relax the strict support constraints, while DILO (Sikchi et al., 2025) utilizes a dual formulation of the χ^2-divergence to improve stability. Despite these advances, these methods fundamentally rely on estimating density ratios, which can become ill-conditioned when the offline behavioral data has limited support to the expert policy.

Surrogate Reward Learning. Another line of work is to introduce the *surrogate rewards* guiding the agent to align with the expert policy. Usually it contains a two-stage approach in which the agent first learns a surrogate reward function $r(s)$ from data and then optimizes a policy using offline RL algorithms. Broadly speaking, this paradigm extends beyond imitation learning and is central to unsupervised reinforcement learning (URLB) (Laskin et al., 2021), where methods typically derive intrinsic *pseudo-rewards* to guide policy acquisition or environment exploration in the absence of external supervision. In the context of offline LfO, standard approaches like ORIL (Zolna et al., 2020) apply adversarial learning to train a discriminator-based reward. However, adversarial objectives are notoriously unstable and often produce signals that are difficult to optimize. Remarkably, our method **TGE** falls into this surrogate reward paradigm but circumvents these stability issues by replacing adversarial discriminators with a dense, geometric distance metric derived from the latent space of a trajectory-level generative planner trained on the offline dataset.

2.2 Diffusion Models in Reinforcement Learning

Diffusion models have rapidly been adopted in RL, primarily for generative control and data synthesis. In the control domain, diffusion models function as policies (Chi et al., 2025; Wang et al.; Ding et al., 2024) or trajectory planners (Janner et al., 2022; Ajay et al., 2022), leveraging iterative denoising to capture multi-modal distributions and solve long-horizon tasks. Recent works have extended this to value-based methods, using diffusion to regularize offline Q-learning (Hansen-Estruch et al., 2023). Beyond control, diffusion models are increasingly used as world models for hallucinating environments (Alonso et al., 2024). In the context of diffusion models in RL, our work focuses on a distinct application in using diffusion model for metric learning. While methods like Stable Rep (Tian et al., 2023) or Diffusion Reward (Huang et al., 2024) similarly derive learning signals from generative models, they typically rely on static visual semantics from video or image models. In contrast, TGE exploits the trajectory-level latent embedding of a temporal diffusion encoder to capture system dynamics, augmenting the reward-free offline dataset that enables robust imitation without the computational cost of generative sampling during downstream offline RL training.

2.3 Representation Learning in Reinforcement Learning

Representation learning is central to scaling RL to high-dimensional observation spaces by extracting compact embeddings that filter task-irrelevant noise. Early approaches largely relied on reconstruction-based auxiliary tasks with world models or autoencoders to compress sensory inputs into latent states that capture the factors of the environment (Ha & Schmidhuber, 2018). Subsequent research adopted contrastive learning and bi-simulation metrics to group states with similar transition dynamics (Laskin et al., 2020; Zhang et al., 2021) to improve the model robustness to task-irrelevant details. However, these approaches typically learn the representation for single-stage observation. To tackle this, we instead leverage the representation learnt from a trajectory diffusion model (i.e., Diffuser (Janner et al., 2022)). These trajectory-level representations provide the temporal context necessary to distinguish behaviors that are visually similar but possess divergent underlying dynamics.

In supporting this line, recent works have shown that diffusion models can be viewed as denoising autoencoders where intermediate features serve as robust representations for downstream tasks (Fuest et al., 2024). For instance, RepFusion (Yang & Wang, 2023) shows that distilling diffusion model yields high-quality representations on image recognition. This suggests that the diffusion training induces semantically meaningful features for complex dynamics.

3 Preliminaries

Markov Decision Process. We consider the discounted Markov Decision Process (MDP) setting, defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. The transition kernel $P(s'|s, a)$ specifies the probability of transitioning to a next state s'

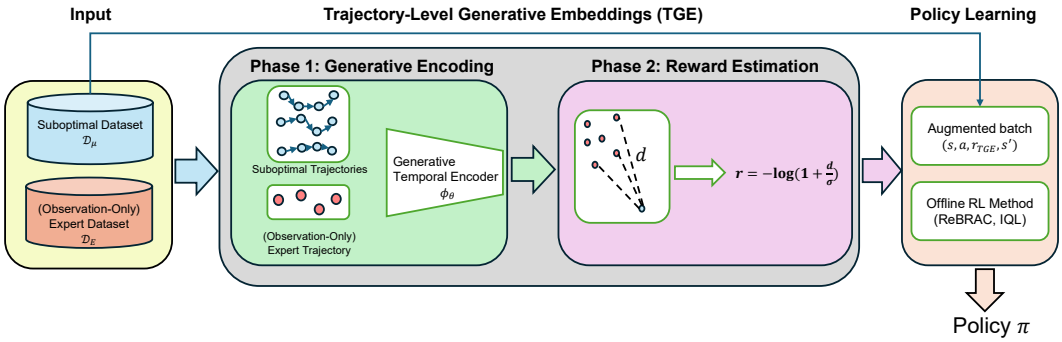


Figure 1: **Overview of Trajectory-level Generative Embeddings (TGE).** The framework employs a trajectory-level diffusion encoder to map trajectory segments into a latent embedding space. A surrogate reward is then computed using a logarithmic kernel over these embeddings to augment the reward-free suboptimal dataset, enabling offline RL training for policy learning.

given the current state s and action a with the reward function defined by $r(s, a)$ and $\gamma \in [0, 1)$ be a discount factor. A policy $\pi(a|s)$ defines a distribution over actions conditioned on the state, and the objective of reinforcement learning is to identify a policy that maximizes the expected discounted return $J(\pi) = \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

In the imitation learning from observations setting, both the true reward function r and the expert actions are typically unavailable. Consequently, the learner must infer desirable behaviors from state-only expert trajectories and suboptimal data providing limited state–action coverage.

Offline Imitation Learning from Observations. In imitation learning from observation (LfO), the agent is given a small expert dataset containing only state observations, without any expert actions or reward annotations. In addition, our work focuses on the offline learning, where the agent cannot directly interact with the environment. Instead, we assume the agent can access to an offline dataset composed of suboptimal state–action trajectories, which provides the necessary coverage for learning.

Let the expert dataset containing one complete state-only expert episode be denoted by $\mathcal{D}_E$, and let $\mathcal{D}_\mu$ denote the suboptimal offline dataset consisting of state–action trajectories generated by a mixed behavioral policy μ, where $\mathcal{D}_\mu$ contains a combination of expert and suboptimal trajectories. The agent does not have access to the trajectory labels; that is, it cannot directly distinguish which trajectories are expert and which are suboptimal. This setting is standard practice in offline imitation learning from observations (Sikchi et al., 2025; Yan et al., 2024). We define the normalized discounted state visitation distribution of a policy π as $\rho_\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$. Accordingly, let ρ_E, ρ_π, and ρ_μ represent the distributions induced by the expert policy π_E, the learned policy π, and the data-collecting policy μ, respectively. The goal of the learning algorithm is to recover a policy π whose behavior closely matches that of the expert policy π_E.

Denoising Diffusion Probabilistic Models. Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020) are a class of generative models that learn to approximate complex data distributions through a sequence of iterative denoising operations. In particular, the forward diffusion process gradually corrupts a clean sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ by adding Gaussian noise over K steps according to $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = \mathcal{N}\big(\sqrt{1 - \beta_k}\, \mathbf{x}_{k-1},\, \beta_k I\big)$, where $\{\beta_k\}_{k=1}^{K}$ is a variance schedule controlling the noise magnitude at each step. After sufficient diffusion steps, this process yields a nearly isotropic Gaussian distribution $q(\mathbf{x}_K) \approx \mathcal{N}(0, I)$.

The generative (reverse) process is parameterized by a neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_k, k)$ that predicts the injected noise and the posterior $p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_k, k), \boldsymbol{\Sigma}_\theta(\mathbf{x}_k, k))$, where the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_k, k)$ is computed from the denoising model $\boldsymbol{\epsilon}_\theta(\mathbf{x}_k, k)$. Training is performed by minimizing a reweighted variational lower bound, which simplifies to the following mean-squared error objective:

$$\mathcal{L} = \mathbb{E}_{k,\mathbf{x}_0,\boldsymbol{\epsilon}} \left[\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_k}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_k}\boldsymbol{\epsilon}, k) \|_2^2 \right],$$

where $\bar{\alpha}_k = \prod_{i=1}^{k}(1 - \beta_i)$. Once trained, the model generates new samples by iteratively denoising through the learned reverse process starting from Gaussian noise $\mathbf{x}_K \sim \mathcal{N}(0, I)$. Recent advances in

offline reinforcement learning have leveraged diffusion models as trajectory-level planners, enabling long-horizon reasoning and trajectory generation (Janner et al., 2022; Ajay et al., 2022).

4 METHODOLOGY

We aim to develop a novel offline imitation learning from observations framework that leverages trajectory-level generative planners as encoders. Such planners enable the algorithm to reason over extended horizons and incorporate rich temporal structure that is difficult to capture using stepwise imitation approaches. To achieve this, we propose **T**rajectory-level **G**enerative **E**mbeddings (TGE), a method that constructs reward signals from the latent embeddings of a generative planner trained to perform density estimation over the suboptimal dataset $\mathcal{D}_\mu$. These trajectory-level latent embeddings provide a discriminative representation capable of distinguishing expert-like and suboptimal trajectories, thereby serving as an informative reward proxy for offline policy optimization on $\mathcal{D}_\mu$. Finally, we train an offline RL algorithm on the entire offline suboptimal dataset $\mathcal{D}_\mu$ using the TGE-derived reward function. The overall procedure is summarized in Algorithm 1 in the appendix, and an overview of the pipeline is illustrated in Figure 1.

4.1 AN ENTROPY PERSPECTIVE ON DISTRIBUTION MATCHING

The fundamental goal of imitation learning is to recover a policy π such that its induced latent state distribution ρ_π matches the expert distribution ρ_E. We formalize this by seeking to minimize the particle-based cross-entropy $H(\rho_\pi, \rho_E)$. Given n data points $\{z_i\}_{i=1}^n$ sampled from a latent state distribution induced by a policy π, we leverage non-parametric entropy estimation techniques (Singh et al., 2003; Liu & Abbeel, 2021) to express the particle-based cross-entropy with respect to the expert distribution as:

$$H(\rho_\pi, \rho_E) = -\frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{m}{n v_i^m}\right) + b(m) \propto \sum_{i=1}^{n} \frac{\log v_i^m}{n},$$

where $b(m)$ is a bias correction term depending solely on the nearest neighbor parameter m. Here, v_i^m denotes the volume of a hypersphere with radius $\|z_i - z_E^m\|$, representing the Euclidean distance between a policy datapoint z_i and its m-th nearest neighbor in the expert distribution:

$$v_i^m = \frac{\pi^{d_z/2}}{\Gamma(d_z/2+1)} \|z_i - z_E^m\|^{d_z},$$

where $d_z = |\mathcal{Z}|$ is the dimension of the latent space and Γ is the gamma function. Consequently, by substituting the volume term, the cross-entropy can be reformulated as:

$$H(\rho_\pi, \rho_E) \approx \frac{1}{n}\sum_{i=1}^{n} \log\Big(c + \frac{1}{m}\sum_{z_E^m \in \mathcal{N}_m(\mathcal{Z}_E)} \|z_i - z_E^m\|^{d_z}\Big). \tag{1}$$

This derivation reveals that minimizing the cross-entropy between distributions is equivalent to minimizing the log-distance between behavioral samples and their expert neighbors. To provide a smooth and robost learning signal, we generalize this estimator by averaging over the entire m-nearest neighborhood.

Following (Liu & Abbeel, 2021), we approximate the estimator $\log(c + \frac{1}{m}\sum \|z_i - z_E\|^{d_z})$ as $\log(1 + \frac{1}{m}\sum \|z_i - z_E\|)$ for numerical stability and define the surrogate reward by

$$r(z_i) = \log\left(1 + \tfrac{1}{m}\textstyle\sum_{z_E^m \in \mathcal{N}_m(\mathcal{Z}_E)} \|z_i - z_E^m\|\right), \tag{2}$$

plugging this into equation 1 yields that the cross-entropy between the occupancy measure ρ_π and the expert behavior ρ_E can be approximated by:

$$\begin{aligned} H(\rho_\pi, \rho_E) &\approx \frac{1}{n}\sum_{i=1}^{n} \log\Big(1 + \frac{1}{m}\sum_{z_E^m \in \mathcal{N}_m(\mathcal{Z}_E)} \|z_i - z_E^m\|^{d_z}\Big) \\ &\approx \tfrac{1}{n}\sum_{i=1}^{n} r(z_i) \approx \mathbb{E}_{z_i \sim \rho_\pi}[r(z_i)] \\ &= \mathbb{E}_\pi\left[\sum_{h=0}^{\infty} \gamma^h r(z_h)\right], \end{aligned} \tag{3}$$

where the second line is because z_i is sampled from ρ_π and the third line holds according to the definition of occupancy measure $\rho_\pi(\cdot) = \sum_{h=0}^{\infty} \gamma^h P_h(\cdot|\pi)$. The entropy estimation equation 3 suggests that optimizing the policy to maximize the cumulative surrogate reward defined in equation 2 is equivalent to minimizing the cross-entropy between the latent distributions of the expert and behavioral policies. Having established this theoretical framework, we describe our method in the following sections.

4.2 Generative Planner Learns Separable Embeddings

We treat the encoder of the trajectory diffusion planner Diffuser (Janner et al., 2022) as an unsupervised representation learner and use its trajectory-level latent embedding for imitation. As discussed in Section 2.3, the denoising objective encourages the model to learn temporally coherent features that are robust to noise, making them ideal for imitation. Specifically, as outlined in Line 1 of Algorithm 1, we train the diffusion noise model on the suboptimal dataset $\mathcal{D}_\mu$ using the objective

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau_0 \sim \mathcal{D}_\mu,\, k,\, \boldsymbol{\epsilon}}\big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\tau_k, k)\|_2^2\big], \tag{4}$$

where $k \sim \mathrm{Unif}[K]$ denotes the diffusion timestep, $\epsilon \sim \mathcal{N}(0, I)$ is the Gaussian noise target, and τ_k is the noisy trajectory obtained by adding noise to the clean trajectory τ_0 with length H at diffusion timestep k. Let $\tau \in \mathcal{T}$ denote a trajectory segment of length H, where the trajectory space for training is defined as $\mathcal{T} = \mathbb{R}^{H \times (|\mathcal{S}| + |\mathcal{A}|)}$. To construct these segments from $\mathcal{D}_\mu$, we extract overlapping sequences of length H using a sliding window. While the diffusion model is trained on these state–action trajectories, the encoder ϕ_θ is designed to produce a state-only latent embedding. Specifically, during the embedding process, we mask the action dimensions of τ, ensuring the representation space $\mathcal{Z}$ captures temporal state dynamics consistent with the observation-only expert data in $\mathcal{D}_E$. We achieve this by conceptually decomposing the U-Net noise model ϵ_θ (Ronneberger et al., 2015) into a temporal encoder $\phi : \mathcal{T} \times \mathbb{R} \to \mathcal{Z}$ and a decoder $\psi : \mathcal{Z} \times \mathbb{R} \to \mathcal{T}$, such that $\epsilon_\theta(\tau_k, k) = (\psi_\theta \circ \phi_\theta)(\tau_k, k)$. Minimizing the denoising score-matching objective in Equation equation 4 encourages the latent representation $\mathcal{Z}$ to preserve information about the underlying clean trajectory distribution, as it must support reconstruction across a range of Gaussian noise levels. Consequently, $\mathcal{Z}$ captures temporally coherent features that are stable under perturbations, yielding an embedding space that is more semantically discriminative for downstream similarity-based rewards.

Given a clean input trajectory $\tau_0 \sim \mathcal{D}_\mu$, we extract a latent embedding using the encoder. Figure 2 illustrates that, even without any explicit discriminative supervision between expert-like and suboptimal trajectories in $\mathcal{D}_\mu$, the encoder-induced embedding $z = \phi_\theta(\tau_0, 0)$ naturally separates expert-like samples from suboptimal ones in the mixed dataset. Given this desirable separation property of the embeddings, it is natural to leverage them for reward construction by directly measuring an appropriate distance between embeddings as we discussed in Sec. 4.1.

4.3 Reward Estimation on Trajectory-level Embeddings

We now describe the procedure for generating synthetic rewards based on the trajectory-level embeddings induced by the encoder of the generative planner. Given the expert dataset $\mathcal{D}_E$, we extract expert embeddings $\mathcal{Z}_E$ by applying a sliding window with a stride of 1 to the state-only expert episode. Specifically, $\mathcal{Z}_E = \{\phi_\theta(\tau_{i:i+H}) \mid \tau \in \mathcal{D}_E, 0 \le i \le |\tau| - H\}$, where the action dimensions of the input to ϕ_θ are treated as null for these observation-only segments (Line 2 in Algorithm 1).

For each state s_t in the suboptimal dataset $\mathcal{D}_\mu$, we consider the trajectory segment followed by, $\tau_\mu = s_{t:t+H}$, and encode it into a latent vector $z_t = \phi_\theta(\tau_\mu)$. In the expert embedding space $\mathcal{Z}_E$, we compute the ℓ_2-norms between z_t and all expert embeddings $z_E \in \mathcal{Z}_E$, and then identify the top-m nearest neighbors $\mathcal{N}_m(\mathcal{Z}_E)$.

To construct an entropy-based surrogate reward signal from the suboptimal trajectory segment τ_μ, we compute the average logarithmic distance between the latent embedding z_t and its m nearest expert neighbors:

$$r_{\mathrm{TGE}}(s_t) = \frac{1}{m} \sum_{z_E \in \mathcal{N}_m(\mathcal{Z}_E)} f(\|z_t - z_E\|_2/\sigma)\,, \tag{5}$$

where σ is a kernel temperature. Note that we normalize all latent embeddings by projecting them onto the unit sphere (dividing each vector by its L_2-norm), ensuring that all embeddings have equal length and allowing for consistent distance computations under the kernel. We adopt a logarithmic kernel $f(d) = -\log(1 + d)$. This serves as a proxy for density estimation, consistent with the theoretical derivation linking this metric to particle-based entropy maximization (Liu & Abbeel, 2021) provided in Section 4.1. To justify our theoretical insights, we provide the ablation study in Appendix E which shows that our theoretically grounded selection provides the best performance.

4.4 Downstream Policy Learning

Since the original suboptimal dataset $\mathcal{D}_\mu$ consists of state–action transitions without reward labels, the reward estimation procedure described in Section 4.3 enables us to annotate each transition with a synthetic reward $r_{\mathrm{TGE}}(s)$, which measures the similarity of the current state to the expert state distribution. Higher values indicate stronger expert-likeness, while lower values correspond to more suboptimal behavior.

With these synthetic rewards, the augmented dataset $\mathcal{D}'_\mu = \{(s_t, a_t, r_{\mathrm{TGE}}(s_t))\}$ can be used to train any offline RL method for policy learning. In our experiments, we adopt two state-of-the-art offline RL algorithms, Implicit Q-Learning (IQL) (Kostrikov et al., 2021) and Revisited Behavior Regularized Actor-Critic (ReBRAC) (Tarasov et al., 2023), and demonstrate that both methods effectively learn high-quality policies when equipped with our synthetic reward signal. Notably, for ReBRAC, we introduce an adaptive behavior cloning mechanism that weights the imitation penalty by the TGE reward to prioritize expert-like transitions during training. Full implementation details are provided in Appendix B.2.

4.5 Learning under Support Mismatch

We analyze the structural differences between our proposed TGE approach and the prevailing occupancy matching methods. Standard dual optimization fundamentally rely on estimating density ratios, $w(s) \approx d^{\pi_E}(s)/d^{\mathcal{D}_\mu}(s)$, to reweight offline transitions. This formulation typically assumes a nontrivial degree of support overlap (Ma et al., 2022), such that expert transitions are reasonably represented within the offline dataset. In more realistic settings where the offline data poorly covers the expert distribution, this ratio can become ill-conditioned. For suboptimal states outside the expert support, $d^{\pi_E}(s) \to 0$, causing importance weights to vanish. As a result, supervision concentrates on a small subset of in-support states, while much of the offline data receives nearly indistinguishable near-zero weights, limiting the ability to provide graded feedback on how different suboptimal behaviors compare or how they should be improved.

In order to reduce the reliance on strict support overlap, we seek to construct a distance-shaped reward in the learned latent space of the generative planner under the entropy formulation mentioned above. Rather than assigning near-uniformly negligible supervision to out-of-distribution states, the logarithmic entropy-based reward $r(s) = -\log(1+\|z_s - z_E\|)$ induces a smoothly decaying, graded signal that persists even for states far from the expert support. This allows suboptimal trajectories to remain comparable through their relative proximity to expert behavior, providing a dense preference signal that prioritizes trajectories closer to the expert instead of restricting learning to a small overlapping subset.

The effectiveness of this extrapolation depends on the semantic quality of the underlying metric. Standard distance metrics, such as L_2 distance on raw observations, often fail due to state aliasing, where states that are close under a raw metric may be functionally distinct (e.g., spatially proximate

states separated by an obstacle in maze environments). By leveraging the encoder of a trajectory-level generative planner, we exploit the model's generative inductive bias, which is trained to capture joint temporal structure over trajectories. As a result, the induced latent space is organized according to long-horizon dynamics rather than superficial single-step features, making the resulting distances more reflective of functional similarity and suited for robust imitation learning.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Environments and Datasets. We evaluate our method on offline learning-from-observations (LfO) benchmarks built on D4RL, covering both locomotion and manipulation domains. The locomotion tasks are based on MuJoCo, including `hopper`, `halfcheetah`, `walker2d`, and `ant`, while the manipulation tasks are based on Adroit, including `pen`, `door`, and `hammer`. For MuJoCo, the offline suboptimal dataset $\mathcal{D}_\mu$ contains approximately 1M transitions drawn from the D4RL `random` or `medium` datasets, following established LfO protocols used in SMODICE (Ma et al., 2022) and DILO (Sikchi et al., 2025). For Adroit, $\mathcal{D}_\mu$ consists of `human` and `cloned` trajectories, again adhering to the standard benchmark protocol.

Following prior work, we consider two mixture settings that differ in the proportion of expert transitions included in $\mathcal{D}_\mu$: (i) *expert*, in which $\mathcal{D}_\mu$ contains 200 unlabeled expert trajectories, and (ii) *few-expert*, in which $\mathcal{D}_\mu$ contains 30 unlabeled expert trajectories; and (iii) *medium*, in which no expert trajectories are added to $\mathcal{D}_\mu$. For the detailed mixture strategy, we refer to Appendix C.

Expert data. All methods are additionally provided with an **observation-only** demonstration $\mathcal{D}_E$ consisting of one single expert episode, which supplies the expert state sequences required by LfO algorithms.

Baselines and Evaluation. We compare against recent offline learning-from-observations (LfO) methods, including SMODICE (Ma et al., 2022), PW-DICE (Yan et al., 2024), and DILO (Sikchi et al., 2025), which have been shown to outperform earlier approaches such as LobsDICE (Kim et al., 2022) and DemoDICE (Kim et al., 2021). We additionally include trajectory-level behavior cloning (BC) as a baseline, trained on the suboptimal data using diffuser (Janner et al., 2022) without reward guidance. We report the mean and standard deviation of normalized D4RL scores across five seeds for each tasks and methods. The pseudo code and hyperparameter are presented in Appendix B and the detailed training dynamics is visualized in Appendix F.

5.2 EXPERIMENTAL RESULTS

As shown in Table 1, TGE combined with existing offline RL backbones consistently matches or outperforms all baselines across MuJoCo locomotion environments.

In `random+expert` and `medium+expert`, where the offline buffer contains relatively sufficient expert-like coverage, TGE remains highly competitive and achieves strong performance. More importantly, the performance gains are most pronounced in `random+few-expert` and `medium+few-expert`, where expert trajectories are scarce and the offline dataset is dominated by suboptimal behaviors, inducing a substantially more challenging distributional mismatch with limited support overlap between $\mathcal{D}_\mu$ and $\mathcal{D}_E$. In these settings, occupancy-matching methods such as SMODICE (Ma et al., 2022), DILO (Sikchi et al., 2025) and PW-DICE (Yan et al., 2024) suffer substantial performance degradation and often fail to recover a functional policy. We observe similar robustness in the `medium` regime, where no unlabeled expert trajectories are mixed into $\mathcal{D}_\mu$, suggesting that TGE can still identify and exploit expert-like structure through trajectory-level generative geometry even under severe support mismatch.

Notably, in several settings TGE even surpasses the performance of the provided expert demonstrations. We hypothesize that this is because the single observation-only expert trajectory used for imitation may not represent the best expert behavior in the dataset. By measuring trajectory-level proximity in the diffusion embedding space, TGE can assign higher rewards to all expert-like trajectories and effectively recover high-quality behaviors present in the offline buffer.

Table 1: **Main Results.** TGE (combined with IQL or ReBRAC, highlighted in cyan) is compared against state-of-the-art offline LfO baselines. The results demonstrate that TGE achieves superior performance. We report the mean and standard deviation for each result with five random seeds. The top-performing results and those with overlapping standard deviations are highlighted in **bold**.

Dataset	Env	Diffuser (BC)	SMODICE	PW-DICE	DILO	TGE+IQL	TGE+ReBRAC	Expert
random+expert	hopper	1.45±0.17	103.79±4.76	**106.22±8.21**	**104.96±6.52**	103.69±3.68	**109.72±0.68**	111.34
	halfcheetah	0.15±0.49	77.10±6.72	90.63±1.08	89.40±2.65	87.88±0.99	**93.20±0.70**	88.83
	walker2d	-0.01±0.04	**108.74±0.72**	**107.97±0.94**	**108.88±0.39**	**108.47±0.53**	**108.68±0.20**	106.93
	ant	9.58±4.77	**124.10±2.19**	**123.15±8.48**	**122.87±4.18**	**122.58±4.04**	**122.72±4.19**	130.75
random+few-expert	hopper	1.38±0.06	62.04±17.02	**89.32±17.29**	**92.66±7.42**	50.57±16.55	**88.86±12.28**	111.34
	halfcheetah	0.03±0.54	2.60±0.64	31.16±32.38	48.26±14.18	5.76±1.98	**89.70±1.63**	88.83
	walker2d	0.01±0.03	16.97±35.72	97.68±10.98	**107.32±1.96**	96.56±6.64	**105.48±2.52**	106.93
	ant	7.25±0.07	32.11±8.38	**108.08±15.99**	**112.50±7.77**	42.76±10.78	**111.46±4.14**	130.75
medium+expert	hopper	45.52±4.81	57.77±8.32	69.38±28.15	102.52±5.73	99.89±4.16	**109.64±0.25**	111.34
	halfcheetah	41.33±0.49	57.27±2.20	60.76±3.15	89.98±0.61	66.66±5.34	**93.10±0.28**	88.93
	walker2d	59.74±5.69	70.03±18.38	85.17±6.85	**108.52±0.71**	**108.88±0.57**	**108.95±0.61**	106.93
	ant	87.42±6.38	104.68±5.21	117.51±6.84	92.41±2.62	104.56±5.40	**137.89±0.45**	130.75
medium+few-expert	hopper	50.74±2.92	52.02±2.64	56.79±11.55	38.07±11.56	61.97±1.65	**97.24±2.22**	111.34
	halfcheetah	41.49±1.28	41.60±2.99	46.33±7.44	66.07±8.52	42.74±0.11	**91.83±0.80**	88.83
	walker2d	61.57±6.92	73.72±4.57	80.77±3.14	70.19±2.51	77.20±2.25	**108.02±0.76**	106.93
	ant	86.13±7.24	89.11±1.53	101.26±8.73	92.55±5.60	95.95±3.13	**130.23±1.36**	130.75
medium	hopper	46.33±1.86	56.18±1.16	**65.82±11.39**	48.68±17.85	64.38±1.68	**67.90±1.59**	111.34
	halfcheetah	41.64±0.30	42.44±0.50	43.27±0.58	42.07±0.47	42.87±0.29	**48.00±0.30**	88.83
	walker2d	65.62±4.89	71.28±7.21	75.64±3.58	69.74±1.13	75.15±0.85	**84.56±0.68**	106.93
	ant	85.67±11.66	88.94±2.89	94.90±4.55	90.13±4.68	93.18±0.70	**118.52±0.53**	130.75
cloned+expert	pen	30.98±46.56	22.63±6.35	1.49±3.61	**61.05±10.86**	**59.58±4.49**	**63.15±5.11**	167.18
	door	0.27±1.84	-0.08±0.08	1.17±2.16	**100.90±2.48**	0.02±0.00	11.20±7.82	103.95
	hammer	0.59±0.19	0.48±0.46	6.00±5.44	**54.29±23.25**	1.34±1.06	37.84±8.05	125.72
human+expert	pen	73.84±18.47	47.64±14.83	27.71±6.32	**99.33±10.13**	**99.64±3.63**	90.30±5.36	167.18
	door	47.21±12.72	1.50±1.06	0.06±0.03	**96.45±4.55**	**98.02±2.63**	**101.43±3.73**	104.73
	hammer	42.33±20.70	0.34±0.23	29.26±34.09	91.24±13.50	**106.27±2.49**	**109.63±9.16**	125.72

These empirical findings support our analysis in Section 4.5, which suggests that occupancy-matching objectives become less informative under severe support mismatch between $\mathcal{D}_\mu$ and $\mathcal{D}_E$. By constructing rewards based on geometric proximity in the latent space of the generative planner, rather than direct support overlap, TGE is able to recover meaningful reward signals, thereby bridging the gap where occupancy ratio–based methods tend to fail.

In Adroit tasks, TGE exhibits a trade-off between human-collected and generated data. On `human+expert` datasets, TGE outperforms DILO (Sikchi et al., 2025) and PW-DICE (Yan et al., 2024) to set a new SOTA, demonstrating robustness to non-Markovian noise in human demonstrations. Conversely, on `cloned+expert` data, TGE is less consistent—leading on `pen` but trailing DILO on `door` and `hammer`. This suggests density-ratio methods like DILO may better exploit the structural consistency of cloned data. Nonetheless, TGE's superior performance on human data highlights its practical utility for noisy, real-world settings.

6 Conclusion

In this work, we introduced Trajectory-level Generative Embeddings (TGE), a novel framework for offline imitation learning from observations that addresses the structural limitations of distribution-matching approaches. By shifting the paradigm from density ratio estimation to generative representation learning, TGE overcomes the support coverage assumption that hampers prior methods. Motivated by the theoretically justified particle-based entropy estimation, we demonstrated that encoding trajectories into the latent space of a temporal diffusion model allows for the construction of a dense, informative reward signal even when the support mismatch between offline data and the expert is severe.

These findings suggest that generative models can serve as powerful tools for structuring the state space in RL, going beyond their traditional roles in RL-based planning and synthesis. Future work could explore extending this trajectory-level latent reward framework to online fine-tuning settings or leveraging larger-scale video diffusion models to enable imitation from cross-embodiment observations.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems (RSS)*, 2019.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. *Advances in Neural Information Processing Systems*, 37:53945–53968, 2024.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning (CoRL)*, 2021.

Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Bjorn Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Josiah Hanna and Peter Stone. Grounded action transformation for robot learning in simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

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

Tao Huang, Guangqi Jiang, Yanjie Ze, and Huazhe Xu. Diffusion reward: Learning rewards via conditional video diffusion. In *European Conference on Computer Vision*, pp. 478–495. Springer, 2024.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Geon-Hyeong Kim, Jongmin Lee, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. Lobsdice: Offline learning from observation via stationary distribution correction estimation. *Advances in Neural Information Processing Systems*, 35:8252–8264, 2022.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pp. 5639–5650. PMLR, 2020.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191*, 2021.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1118–1125. IEEE, 2018.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. A policy gradient method for task-agnostic exploration. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya A Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Harshit Sikchi, Caleb Chuck, Amy Zhang, and Scott Niekum. A dual approach to imitation learning from observations with offline datasets. In *Conference on Robot Learning*, pp. 1125–1147. PMLR, 2025.

Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest-neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23(3-4):301–321, 2003.

Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third person imitation learning. In *International Conference on Learning Representations*, 2017.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36:48382–48402, 2023.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 4950–4957. International Joint Conferences on Artificial Intelligence Organization, 2018.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Kai Yan, Alex Schwing, and Yu-Xiong Wang. Offline imitation from observation via primal wasserstein state occupancy matching. In *International Conference on Machine Learning*, pp. 55814–55838. PMLR, 2024.

Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18938–18949, 2023.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=-2FCwDKRREu`.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando De Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

A Algorithm Pseudocode for TGE

We present the step-by-step implementation of the Trajectory-level Generative Embedding (TGE) framework. The following pseudocode details the reward calculation process and the downstream offline RL training loop.

Algorithm 1 Trajectory-Level Generative Embeddings

Require: Offline suboptimal dataset $\mathcal{D}_\mu$, expert dataset $\mathcal{D}_E$, horizon H, number of candidates m, temperature σ, and kernel function $f(\cdot)$.

Train generative planner on $\mathcal{D}_\mu$ and encoder ϕ_θ.
Compute embeddings $\mathcal{Z}_\mu = \phi_\theta(\mathcal{D}_\mu)$ and $\mathcal{Z}_E = \phi_\theta(\mathcal{D}_E)$ on trajectory segments of length H.
Project all latent vectors to a unit sphere: $z \leftarrow z/\|z\|_2$.
for each $z_t \in \mathcal{Z}_\mu$ **do**
    Compute $d(z_t, z_E) = \|z_t - z_E\|_2$ for all $z_E \in \mathcal{Z}_E$.
    Identify the top-m nearest neighbors $\mathcal{N}_m(\mathcal{Z}_E)$.
    $r_{\text{TGE}}(s_t) \leftarrow \frac{1}{m} \sum_{z_{nb} \in \mathcal{N}_m(\mathcal{Z}_E)} f\left(d(z_t, z_{nb})/\sigma\right)$.
end for
Run any offline RL algorithm on the suboptimal dataset augmented with synthetic rewards: $\mathcal{D}'_\mu = \{(s_t, a_t, r_{\text{TGE}}(s_t), s'_t)\}$.

B Hyperparameters and Implementation Details

B.1 Hyperparameters

We list the hyperparameters used for the diffuser, including details of the U-Net architecture, as well as those for TGE reward estimation in Table 2. We also report the hyperparameters for the downstream offline RL backbones, IQL and ReBRAC, in Tables 3 and 4, respectively. For all baseline methods, we strictly follow the original implementations and hyperparameter settings reported in their respective papers.

Table 2: **Hyperparameter settings for TGE.** The hyperparameters below are used to train the trajectory-level generative planner and estimate the rewards.

Hyperparameter	**Value**
Reward Estimation	
Kernel type	Logarithmic
Kernel temperature (τ)	1.0
Number of nearest neighbors (m)	10
Diffuser Architecture	
Diffuser horizon (H)	32
Channel multipliers	(1, 2, 4, 8)
Diffusion steps	20
Embedding dimension	64
Learning rate	2e-4
Batch size	32
EMA decay	0.995
Scheduler	DDPMScheduler

Table 3: **Hyperparameter Settings for IQL.**

Hyperparameter	**Value**
Optimizer	
Optimizer	AdamW
Actor Learning rate	$3e-4$
Critic Learning rate	$3e-4$
Value Function Learning rate	$3e-4$
Batch size	1024
Algorithm Specifics	
Discount factor (γ)	0.99
Target smoothing (τ_{target})	0.005
Expectile	0.8
Inverse Temperature (β)	0.5
Dropout rate	0 (Mujoco) / 0.1(Adroit)

Table 4: **Hyperparameter Settings for ReBRAC.** For the actor BC coefficient, values $\{0.1, 0.4, 1.0\}$ are used for {Adroit-Cloned, Mujoco, Adroit-Human}, respectively.

Hyperparameter	**Value**
Optimizer & Architecture	
Optimizer	AdamW
Actor Learning rate	$5e-4$
Critic Learning rate	$5e-4$
Weight decay	$1e-4$
Batch size	4096
Hidden layers	3
Hidden dimension	256
Layer normalization	True
Actor Dropout	0 (Mujoco) / 0.1(Adroit)
Algorithm Specifics	
Discount factor (γ)	0.99
Target smoothing (τ_{target})	0.005
Policy update frequency	2
Policy noise (σ)	0.2
Noise clip	0.5
Actor BC coefficient (β_{actor})	0.1, 0.4, 1.0
Critic BC coefficient (β_{critic})	1.0

B.2 Implementation Details

We implement the TGE reward as the logarithmic distance to the nearest expert embeddings, consistent with the entropy minimization objective derived in Section 4.5. All embeddings are first normalized to the unit hypersphere ($z \leftarrow z/\|z\|_2$). Distances are computed using Euclidean distance on the sphere, which is monotonic with cosine distance ($\|z_i - z_j\|_2 = \sqrt{2 - 2\cos\theta}$).

For each offline transition, we retrieve its m nearest expert neighbors in the embedding space (with m=10 by default) and compute the reward using a logarithmic kernel: $r = -\sum_{k=1}^{m} \log(1 + d_k/\sigma)$, where d_k is the distance to the k-th neighbor and σ is a temperature scaling parameter. This reward signal is then normalized and used to label transitions in $\mathcal{D}_\mu$.

Finally, we train offline policies using IQL and ReBRAC as drop-in backbones, relying solely on $\mathcal{D}_\mu$ and the learned TGE reward, without any online interaction. Additionally, to ensure stable learning and balanced sampling across the diverse trajectories in the suboptimal dataset, we utilize the normalized TGE reward as a weighting factor for the behavioral cloning term in our ReBRAC

backbone. We treat rewards as fixed confidence weights that scale the MSE penalty between the actor's actions and the dataset actions. This ensures that the policy regularizes more strongly toward expert-labeled regions (high reward) while maintaining flexibility in suboptimal regions, effectively preventing the policy from collapsing toward poor-quality data points.

B.3 Computational Overhead

As noted in the experimental evaluation, TGE involves a multi-stage pre-processing pipeline before policy learning begins. We report the typical wall-clock times for these stages below:

Generative Training: Training the trajectory diffusion model on the suboptimal dataset $\mathcal{D}_\mu$ requires approximately 4-6 hours on a single L40S GPU. While this is higher than the zero-cost initialization of occupancy-matching baselines, it is a one-time offline cost that enables the extraction of temporally coherent features.

Reward Annotation: Computing embeddings and performing the top-m nearest neighbor search for 1M transitions takes roughly 1 hour.

Total Training Efficiency: Although the pre-processing is more intensive than baselines, it enables the use of standard offline RL backbones without any generative sampling overhead during the RL loop. This makes the downstream policy training phase as efficient as standard RL on labeled data.

C Details of the Mixture Strategy for the Suboptimal Dataset

We provide a detailed breakdown of the data mixture strategy used to construct the suboptimal dataset $\mathcal{D}_\mu$. Following established protocols in offline imitation learning from observations introduced in (Ma et al., 2022; Sikchi et al., 2025), we construct $\mathcal{D}_\mu$ by combining a large corpus of suboptimal transitions with a limited number of expert trajectories. Table 5 summarizes the detailed composition for all suboptimal datasets.

Table 5: **Dataset Mixture Details.** We summarize the data mixture strategies for the suboptimal datasets $\mathcal{D}_\mu$ used across different experimental settings.

Dataset Name	Suboptimal Data	Expert Trajectories	Episode Length
Locomotion Tasks			
Random + Expert	1e6 transitions	200	1000
Medium + Expert	1e6 transitions	200	1000
Random + Few-Expert	1e6 transitions	30	1000
Medium + Few-Expert	1e6 transitions	30	1000
Medium	1e6 transitions	0	1000
Adroit Tasks			
Pen (Cloned + Expert)	5e5 transitions	30	100
Pen (Human + Expert)	5e3 transitions	30	100
Door (Cloned + Expert)	1e6 transitions	30	200
Door (Human + Expert)	6.7e3 transitions	30	200
Hammer (Cloned + Expert)	1e6 transitions	30	200
Hammer (Human + Expert)	1.1e4 transitions	30	200

D T-SNE Embedding Visualization

To further investigate the properties of the learned latent space, we visualize the embeddings using T-SNE. Figure 2 illustrates how TGE distinguishes between different data distributions.

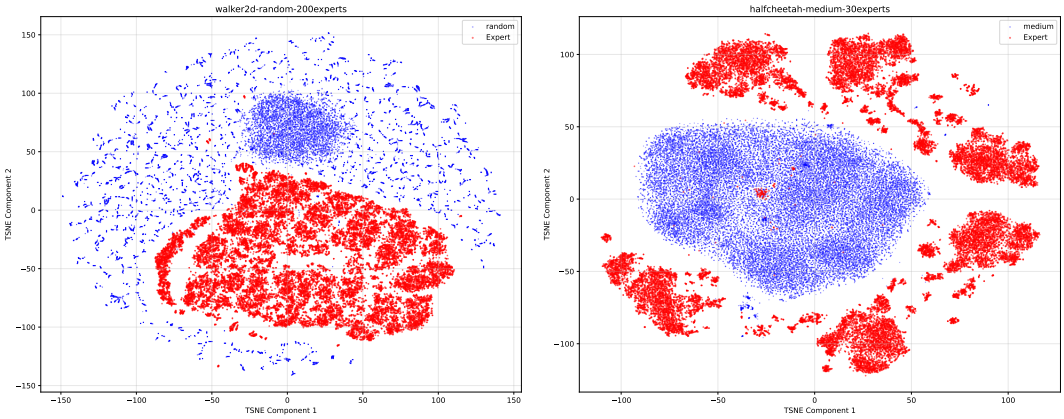


Figure 2: **T-SNE Embedding Visualization.** The latent embeddings produced by the generative planner naturally separate expert transitions from suboptimal ones. As shown, expert samples (colored in red) and suboptimal samples (colored in blue) form distinct clusters with a clear boundary, indicating that the trajectory-level embedding space is highly discriminative.

E Ablation Studies

E.1 Impact of Horizons in the Generative Planner

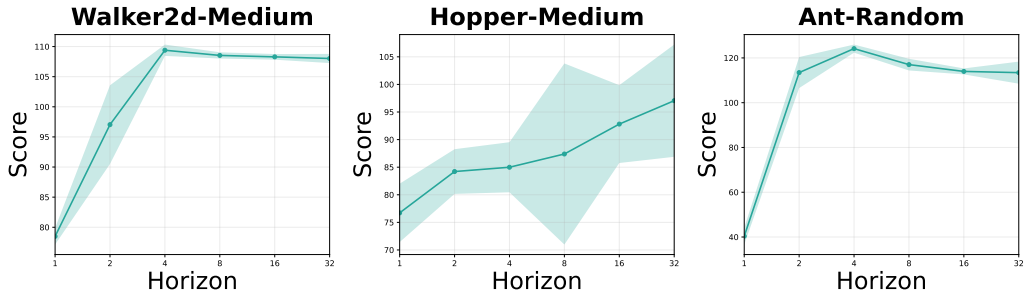
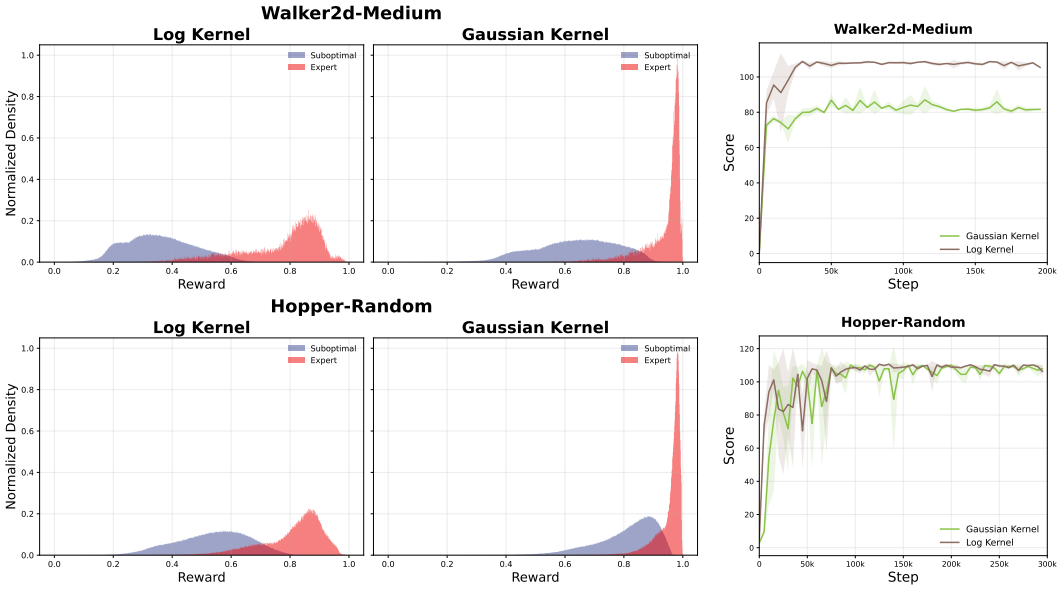


Figure 3: **Ablation of the Temporal Horizon.** The results demonstrate how performance varies with the context horizon H. We observe that larger H generally leads to improved performance, highlighting the importance of temporal context.

We examine how the length of temporal context used by the generative planner influences reward estimation quality and the performance of the downstream offline RL policy. Specifically, we vary the horizon length H employed during planner training and embedding computation. As shown in Figure 3, using a single-step encoding (H=1) leads to a substantial degradation in the final policy performance, highlighting the importance of incorporating temporal context along the trajectory. Increasing H yields nearly consistent performance improvements across tasks when H is small, and saturating after certain horizon lengths. This trend suggests that short-to-mid temporal windows are sufficient to capture essential dynamic information. While longer horizons maintain consistent performance without significant degradation, they may introduce unnecessary computation or over-smoothing effects. Based on these observations, we adopt a conservative default of H=32 for a unified and stable hyperparameter choice across all environments and configurations. Additional ablation studies on other aspects of our proposed method is deferred to Appendix E.

E.2 Choice of Kernels for Reward Estimation

We investigate the choice of kernel function $f(d)$ by analyzing the resulting reward distributions and their associated training dynamics. Figure 4(a) visualizes the density of synthetic rewards assigned to expert and suboptimal trajectories in the offline dataset under Logarithmic and Gaussian kernels, respectively.



(a) Reward Distribution Comparison

(b) Performance

Figure 4: **Ablation of the Reward Signal Density.** (Top) Walker2d-Medium. (Bottom) Hopper-Random. Distributions of synthetic rewards assigned to expert (red) and suboptimal (blue) data. Compared to Gaussian-kernel-based reward estimation, the Logarithmic kernel preserves a heavier-tailed reward signal and is associated with improved performance and training stability.

As shown in Figure 4(a), the Gaussian kernel produces a reward distribution that is highly concentrated in narrow regions: expert rewards collapse near 1.0, forming a sharp peak, while suboptimal trajectories cluster nearby. As a result, meaningful behavioral differences correspond to only minor reward variations, which limits the clarity of the learning signal. In contrast, the Logarithmic kernel (Figure 4(a, left)) yields a broader, heavy-tailed reward distribution. The slow decay better preserves the relative ordering of trajectories in the embedding space, improving reward separability. This enhanced distinguishability facilitates more accurate value estimation by the critic and leads to more stable and optimal learning curves, as shown in Figure 4(b).

E.3 Kernel Temperature

We investigate the sensitivity of our framework to the kernel temperature σ, which regulates the decay rate of the kernel and determines the sharpness of the reward signal in the latent space. Table 6 reports the normalized scores on `Walker2d-medium` and `Hopper-random` across a wide range of values $\sigma \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$.

As shown in the results, TGE demonstrates remarkable robustness to this hyperparameter. Despite varying σ by an order of magnitude, the performance fluctuations remain minimal. These findings confirm that the proposed trajectory-level embedding provides a well-structured metric space where the relative proximity of trajectories is preserved. Consequently, precise task-specific tuning of σ is unnecessary, and we adopt $\sigma = 1.0$ as the default to balance signal discriminativeness and smoothness.

Table 6: **Ablation on Kernel Temperature (σ).** We compare normalized scores across varying temperatures. The method shows robust performance for $\sigma \in [0.1, 5.0]$.

Environment	**Kernel Temperature** (σ)				
	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1.0$ (Default)	$\sigma = 2.0$	$\sigma = 5.0$
Walker2d-medium	108.89±0.39	107.84±0.27	108.68±0.20	108.19±0.31	108.02±0.50
Halfcheetah-random	89.41±1.57	88.26±1.67	89.70±1.63	89.54±0.52	89.56±0.10

E.4 Number of Nearest Neighbors

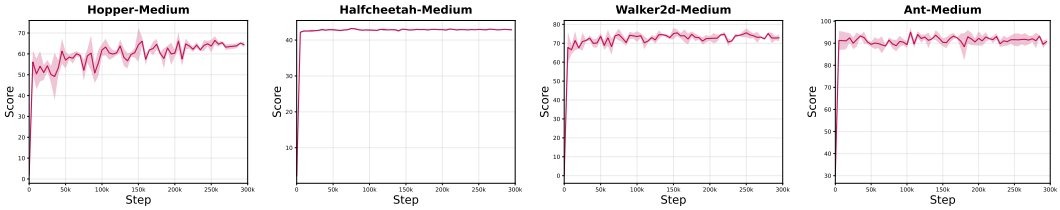
We evaluate the influence of the number of nearest neighbors m on the final performance by changing the value across $m \in \{1, 3, 5, 10, 20\}$. As shown in Table 7, the empirical results demonstrate that our method is remarkably insensitive to the choice of m. Across all tested values, the normalized scores remain highly consistent, with fluctuations largely falling within the standard deviation. This stability indicates that the learned trajectory embeddings are sufficiently discriminative to provide high-quality reward signals even with a small m, while remaining robust to over-smoothing at larger m. Given this robustness, we adopt $m = 10$ as a fixed default for all experiments.

Table 7: **Ablation on Number of Nearest Neighbors (m).** We evaluate the impact of hyperparameter m on the performance over Walker2d-medium and Halfcheetah-random settings.

Environment	**Number of Nearest Neighbors** (m)				
	$m = 1$	$m = 3$	$m = 5$	$m = 10$ (Default)	$m = 20$
Walker2d-medium-v2	108.92±0.83	109.04±0.49	108.47±0.42	108.68±0.20	108.19±0.36
Halfcheetah-random	88.38±1.79	88.24±1.35	88.10±3.06	89.70±1.63	89.19±1.25

F Training Dynamics Visualizations

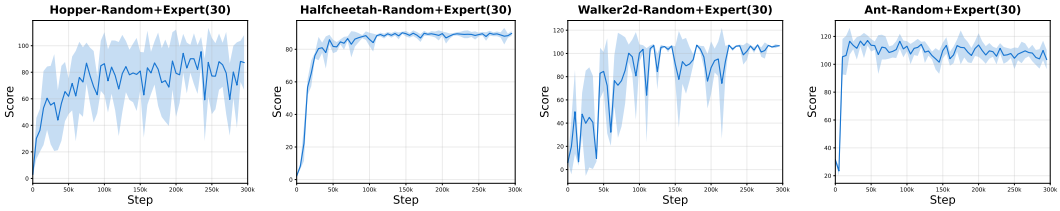
We provide the training curves of TGE combined with IQL and ReBRAC as downstream offline RL methods across different environments to demonstrate the training stability of our proposed method. Figure 5 and Figure 6 visualize the normalized D4RL score (y-axis) over training steps (x-axis) for the IQL and ReBRAC backbones, respectively.



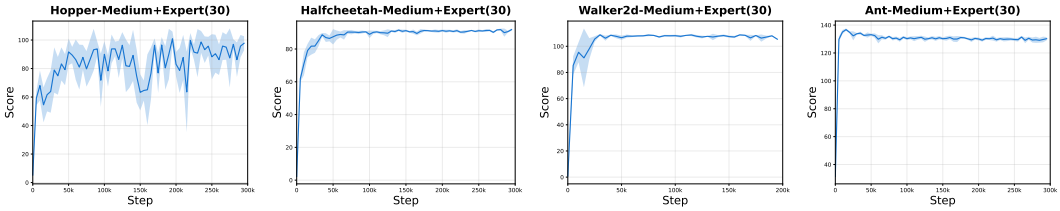
(a) Hopper-Medium (b) HalfCheetah-Medium (c) Walker2d-Medium (d) Ant-Medium

(e) Hopper-Random (f) HalfCheetah-Random (g) Walker2d-Random (h) Ant-Random

Figure 5: **Training Dynamics of TGE + IQL.** The curves display the mean normalized score and standard deviation (shaded region). TGE combined with IQL shows stable policy improvement, confirming that our geometric reward signal enables robust learning across different offline RL methods.



(a) Hopper-Random (b) HalfCheetah-Random (c) Walker2d-Random (d) Ant-Random



(e) Hopper-Medium (f) HalfCheetah-Medium (g) Walker2d-Medium (h) Ant-Medium

Figure 6: **Training Dynamics of TGE + ReBRAC.** The curves display the mean normalized score and standard deviation (shaded region). Similarly, the results demonstrate that the agent achieves rapid convergence to expert-level performance and maintains asymptotic stability, effectively robust to the noise in suboptimal datasets.