TORA: Train Once, Realign Anytime for Offline Multi-Objective Reinforcement Learning

Weichen Li*, Waleed Mustafa*, Marcio Monteiro, Puyu Wang, Marius Kloft, Sophie Fellenz

RPTU University Kaiserslautern-Landau
Gottlieb-Daimler-Strasse, 67663 Kaiserslautern, Germany

Abstract

Intelligent agents in real-world applications must adapt their behavior to changing contexts and user preferences. For example, planning a road trip requires considering both travel time and cost. Multi-objective reinforcement learning (MORL) provides a principled approach to navigate such trade-offs. However, most existing approaches require predefined preference weights during training and jointly optimize the model for all objectives. In this paper, we introduce TORA (Train Once, Realign Anytime), a novel framework that defers preference integration to inference time, enabling flexible adaptation to user preferences without retraining. TORA independently trains diffusion planning models for each objective and combines them at inference time using user-specified preferences to generate behavior aligned with desired trade-offs. Furthermore, new objectives can be added seamlessly by training additional models without modifying existing ones. Empirical evaluations on standard offline MORL benchmarks demonstrate that TORA achieves competitive and consistent performance compared to methods that require fixed preference weights.

1 Introduction

Many real-world decision-making problems involve balancing multiple, often conflicting objectives. For example, a self-driving car may prioritize speed when the user is in a hurry but switch to energy-saving driving during relaxed trips. Multi-objective reinforcement learning (MORL) provides a principled framework for navigating such trade-offs by optimizing policies over multiple, competing objectives.

In high-stakes or safety-critical domains such as healthcare, finance, or robotics, learning by trial-and-error is expensive or dangerous (Levine et al. 2020). Therefore, we focus on the particularly challenging setting of *offline MORL*, where policies must be learned entirely from pre-collected datasets without any additional environment interaction.

A common approach in offline MORL is to assume access to a preference vector during training (Zhu, Dang, and Grover 2023; Yuan et al. 2024; Lin et al. 2024), which provides a weight for each objective. The policies are often conditioned on these weights and trained to respect the specified balance between objectives. However, this paradigm suffers from several key limitations: (i) policies can overfit to the training preferences and generalize poorly to novel ones; (ii) retraining is required to incorporate new objectives; and (iii) collecting preference-weighted data is expensive and unintuitive. In practice, eliciting precise numerical trade-offs from users is often unrealistic. Asking users to specify exact preference weights to label their past actions, assumes a level of clarity and consistency that rarely exists—especially in domains where preferences are vague, context-dependent, or evolve over time. This creates a significant barrier to deploying MORL systems in the real world.

This motivates our approach: an inference-time preference alignment method for MORL that removes the need to annotate training data with preference weights for multiple objectives. Instead, preferences can be flexibly specified at inference time. This leaves the task of finding a trajectory with the specified balance to the model, rather than letting a human specify trajectories that correspond to a given weight vector for training. However, inference-time alignment must generalize to preference vectors it has never encountered during training, making the problem significantly more challenging.

Inspired by recent advances in *diffusion planning* (Lu et al. 2025; Ajay et al. 2022), our method trains a separate diffusion model for each reward objective, where each model learns to generate high-return trajectories for a single objective. At inference time, we synthesize individual policies dynamically by combining the denoising functions of the individual diffusion models into a composite denoiser. This weighted combination allows us to control the trajectory sampling process using a user-specified preference vector, without having to retrain or modify the individual models. This approach follows the recommendation of Amodei et al. (2016) to enhance agent safety by decomposing and combining multiple reward signals, rather than optimizing a single aggregated one.

Summing up, our contributions are as follows:

- We propose TORA, an offline, multi-objective RL method that requires no training-time preference weights. Preferences can be specified or adjusted at test time without retraining and new objectives can be incorporated without impacting existing components.
- Our experiments on both MORL and safety RL bench-

*These authors contributed equally.

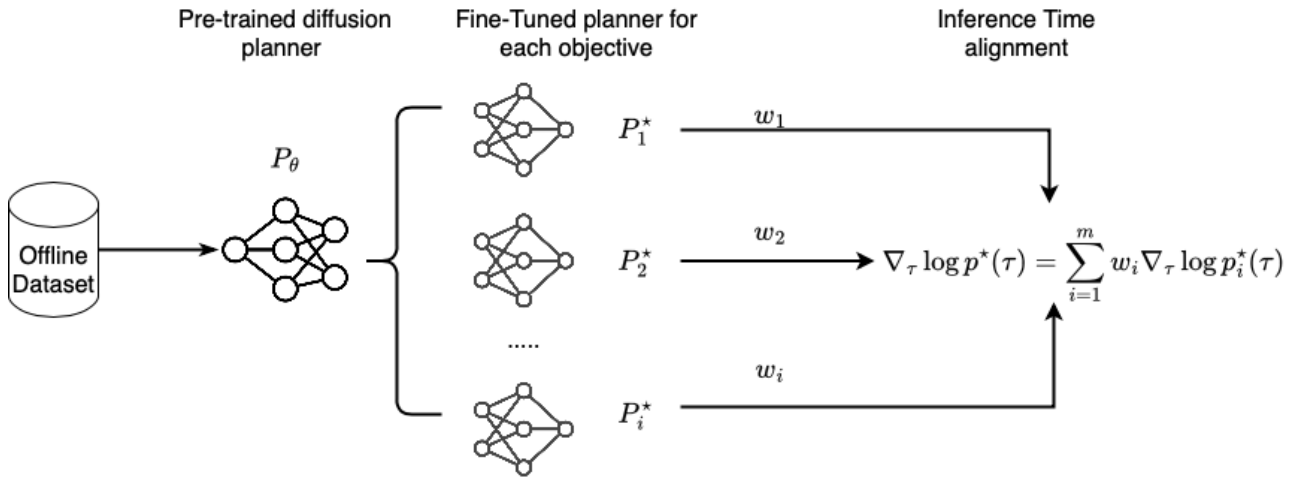


Figure 1: Overview of our framework: We first pre-train a diffusion planner model, and then fine-tune each objective P_i independently using Direct Preference Optimization (DPO),without requiring preference weights during training. At inference time, preference weights are provided for each objective to sample the planned trajectory τ. Since each objective is trained separately, new objectives can be added at any time.

marks show that training MORL without preference weights can achieve performance comparable to methods that require preferences, demonstrating the effectiveness and generality of our approach.

2 Related Work

This section briefly outlines key areas relevant to our work: offline multi-objective and safe RL, and inference-time preference alignment, which serves as the primary motivation for our proposed framework.

2.1 Offline Multi-Objective Reinforcement Learning

Offline MORL aims to learn policies that balance multiple, often conflicting, objectives from a fixed dataset collected. Compared to online MORL (Felten et al. 2023; Hayes et al. 2022; Abels et al. 2019; Yang, Sun, and Narasimhan 2019; Liu et al. 2025), offline MORL has received relatively limited attention.

Previous works typically consider preference weights as inputs during training to guide the policy toward desired trade-offs among objectives. Zhu, Dang, and Grover (2023) introduced the first large-scale benchmark for offline MORL, *D4MORL*, which provides offline trajectories from MuJoCo environments (Towers et al. 2024). In our experiments, we use the same dataset for training. Zhu, Dang, and Grover (2023) integrate preferences into the training process. To condition the policy on preferences, they combine the weights with information such as state and action as inputs instead of only the states. This reformulation enables classical offline RL methods such as CQL (Kumar et al. 2020) to be extended for MORL through preference-aware conditioning. Other works address offline MORL by introducing policy and weights regularization (Lin et al. 2024).

More recently, Yuan et al. (2024) were the first to adopt diffusion planning models in the offline MORL setting. Their key contribution is conditioning the denoising process on the preference weight vector. To address the challenge of out-of-distribution preferences, they further propose a sliding guidance approach, which trains an additional model to capture the preference changes.

Despite the progress made by previous work, a key assumption of existing approaches is that preference vectors are known and fixed during training. As noted in Hayes et al. (2022), multiobjective decision making in the real world often involves scenarios in which user preferences are initially unknown or may change over time. This underscores the need for agents that can flexibly adapt to new or changing preferences at inference time, without requiring retraining. Furthermore, in settings where multiple objectives are jointly optimized by a single agent, adding or removing objectives at typically necessitates retraining the entire agent.

2.2 Offline Safe Reinforcement Learning

In safe RL, the objectives typically include maximizing rewards while minimizing costs (Zhang et al. 2023; Li et al. 2024). Recent extensions of diffusion models have addressed both offline MORL and offline safe RL (Lin et al. 2023; Yao et al. 2024). Similar to MORL approaches, which condition diffusion models on user preference vectors to capture trade-offs among competing objectives, OASIS (Yao et al. 2024) employs a diffusion model conditioned on both reward and cost to generate safe, high-return trajectories. These trajectories are then used to train an offline RL agent. Since safe RL can be viewed as a special case of MORL (Honari, Tamizi, and Najjaran 2024), we adopt diffusion planning in our work to enable flexible preference alignment in both offline MORL and safe RL settings. Most prior work formulates safe RL as a constrained optimization problem

(Hong, Li, and Tewari 2024; Stooke, Achiam, and Abbeel 2020), often relying on techniques such as the Lagrangian method, which involve complex optimization procedures. In contrast, we simplify this process by learning a diverse set of trajectories and selecting the desired trade-off at inference time.

2.3 Inference-Time Preference Alignment

Inference-time alignment has received significant attention in both large language models (LLMs) (Shi et al. 2024; Rame et al. 2023) and diffusion models for image generation (Tang et al. 2024; Yeh, Lee, and Chen 2024). Since user preferences can change at any time, retraining large language or diffusion models is often impractical due to computational cost. Therefore, aligning outputs with user preferences at inference time becomes essential.

While several inference-time alignment methods exist in the LLM domain, applying this idea to offline MORL brings new challenges and is not straightforward. Unlike LLMs, reinforcement learning agents are typically framed as long-horizon sequential decision-making systems rather than single-step generators. This work is the first to explore inference-time alignment for offline MORL, providing both theoretical insights and empirical validation.

3 Preliminaries

This section presents the formal definition of MORL for our following main approach and provides a brief introduction to diffusion models and pairwise preference learning, which will be used to fine-tune the diffusion model in our framework.

3.1 Multi-Objective Reinforcement Learning

We consider Multi-Objective Reinforcement Learning (MORL), which can be formalized as a *Multi-Objective Markov Decision Process (MOMDP)* (Hayes et al. 2022), defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathbf{r}, \mu, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ defines the transition dynamics; $\mathbf{r} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^m$ is a vector-valued reward function with m rewards; μ is the initial state distribution; and $\gamma \in (0, 1]$ is the discount factor. Given a weight vector $w \in \mathbb{R}^m$ with $\sum_{i=1}^m w_i = 1$, the reward vector is *linearly scalarized* as

$$r_w(s, a, s') = \sum_{i=1}^{m} w_i \mathbf{r}_i(s, a, s').$$

Let $T \in \mathbb{N}$ denote the time horizon. A stochastic policy π defines action probabilities $\pi(a \mid s)$ over $\mathcal{A}$. A trajectory of length T is a sequence $\tau = (s_0, a_0, s_1, \ldots, s_{T-1}, a_{T-1}, s_T)$. Given a policy π, transition dynamics $\mathcal{P}$, and an initial state distribution μ, the induced trajectory distribution P^π has density

$$p^\pi(\tau) = \mu(s_0) \prod_{t=0}^{T-1} \pi(a_t \mid s_t)\, \mathcal{P}(s_{t+1} \mid s_t, a_t).$$

The (discounted) return along τ is $R_w(\tau) = \sum_{t=0}^{T-1} \gamma^t r_w(s_t, a_t, s_{t+1})$. We define the *realizable set of trajectories* as

$$\mathcal{T}_{\text{real}} = \bigcup_{\pi} \text{supp}(P^\pi) \subseteq \mathcal{T},$$

i.e., the set of physically valid trajectories in the environment. In MORL, an agent follows a stochastic policy $\pi(a|s; w)$, conditioned on a user-specified weight vector w that may vary at inference time. The goal is to maximize expected scalarized return while remaining close to a reference policy π_{ref}. The optimal policy corresponding to weight w is obtained by solving:

$$\pi^*(\cdot|\cdot; w) = \arg\max_{\pi} \mathbb{E}_{\tau \sim P^\pi} \Big[R_w(\tau) - \beta\, \text{KL}(\pi \| \pi_{\text{ref}}) \Big].$$

Equivalently, one can optimize at the trajectory level. Let P_{ref} be the distribution induced by π_{ref}. Then the optimal trajectory distribution $P^\star_\tau(\cdot; w)$ is given by:

$$P^\star_\tau(\cdot; w) = \arg\max_{P \in \mathcal{P}_{\text{real}}} \mathbb{E}_{\tau \sim P} \Big[R_w(\tau) - \beta\, \text{KL}(P \| P_{\text{ref}}) \Big], \tag{1}$$

where $\mathcal{P}_{\text{real}}$ denotes distributions supported on $\mathcal{T}_{\text{real}}$. This objective seeks a trajectory distribution that maximizes the expected scalarized reward while remaining close to a reference distribution. The constraint $P \in \mathcal{P}_{\text{real}}$ ensures that only trajectories realizable under the Markov Decision Process (MDP) dynamics receive non-zero probability. Notably, while these distributions respect the environment's support, they need not factorize through a policy and transition model, allowing optimization over a richer class than policy-induced distributions.

3.2 Score-Based Diffusion Modeling

Score-based generative models (Song and Ermon 2019; Song et al. 2020) learn to model the data distribution by estimating its *score function*, defined as the gradient of the log-density: $\nabla_\tau \log p_\tau(\tau)$. *Diffusion models* (Ho, Jain, and Abbeel 2020) are a prominent example, where a data point $x_0 \sim p_{\text{data}}$ is corrupted over T steps via a Gaussian forward process:

$$q(\tau_t|\tau_0) = \mathcal{N}(\tau_t; \sqrt{\bar{\alpha}_t}\tau_0, (1 - \bar{\alpha}_t)I).$$

The model learns a neural network $\epsilon_\theta(\tau_t, t)$ to predict the noise added to τ_0, minimizing the denoising objective:

$$\mathbb{E}_{\tau_0, t, \epsilon} \left[\|\epsilon_\theta(\tau_t, t) - \epsilon\|^2 \right], \tag{2}$$

where $\tau_t = \sqrt{\bar{\alpha}_t}\tau_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.

While trained to predict noise, ϵ_θ approximates the score function of the perturbed data distribution (Song, Meng, and Ermon 2021), i.e.,

$$\epsilon_\theta(\tau_t, t) \propto \nabla_{\tau_t} \log p(\tau_t),$$

where τ_t denotes a noisy trajectory at diffusion step t. This approximation enables sample generation via a learned reverse process $p_\theta(\tau_{t-1}|\tau_t)$, making diffusion models a tractable and expressive instance of score-based generative modeling.

3.3 Direct Preference Optimization

Direct Preference Optimization (DPO) (Rafailov et al. 2024) was originally proposed for aligning large language models with human preferences. Unlike traditional reinforcement learning methods that rely on explicit reward signals, DPO optimizes model behavior directly through pairwise preference comparisons.

Wallace et al. (2024) extend DPO to the diffusion model setting. In this context, DPO encourages the model to assign higher likelihood to preferred samples while regularizing it to stay close to a reference distribution using a KL-divergence penalty. The optimization objective is:

$$\max_{p_\theta} \mathbb{E}_{\tau\sim p_\theta}\left[r(\tau)\right] - \beta\, \mathrm{D}_{\mathrm{KL}}\left(p_\theta(\tau) \,\|\, p_{\mathrm{ref}}(\tau)\right)],$$

where p_θ is the optimized generated model, p_{ref} is the reference model, $r(\tau)$ represents the latent reward signal implied by preferences, and β controls the strength of regularization.

This objective can be optimized using a pairwise loss function:

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E}_{(\tau^+,\tau^-)} \log \sigma\big(\beta \log \frac{p_\theta(\tau^+)}{p_{\mathrm{ref}}(\tau^+)} - \log \frac{p_\theta(\tau^-)}{p_{\mathrm{ref}}(\tau^-)}\big) \tag{3}$$

where τ^+ is the preferred sample, τ^- the less preferred one, and $\sigma(\cdot)$ is the sigmoid function. The use of likelihood ratios enforces preference consistency while implicitly minimizing the divergence between p_θ and p_{ref}.

4 Inference-Time Multi-Objective Alignment

In this section, we first present the theoretical motivation behind our framework, followed by a detailed explanation of how we integrate diffusion planning into the offline MORL setting.

4.1 Main Approach

In this subsection, we lay the foundation for our approach. Recall that our goal is to learn a trajectory distribution that solves the optimization problem in Equation (1). Since the linearization preference w is provided at inference time, our objective is to develop a flexible algorithm that can accommodate changes in w without retraining.

The following proposition characterizes the score function of the optimal trajectory distribution $P_\tau^\star$ in the multi-objective setting.

Proposition 1 (Informal) *The score function of the optimal distribution in Equation* (1) *satisfies*

$$\nabla_\tau \log p^\star(\tau) = \sum_{i=1}^{m} w_i \nabla_\tau \log p_i^\star(\tau),$$

where $p_i^\star$ is the density of the optimal distribution P_i^* for the i-th objective:

$$P_i^\star = \arg\max_{P\in\mathcal{P}_{\mathrm{real}}} \mathbb{E}_{\tau\sim P}\Big[\sum_{t=0}^{T-1} \gamma^t \mathbf{r}_i(s_t, a_t, s_{t+1})\Big] - \beta\, \mathrm{KL}(P\|P_{\mathrm{ref}})$$

Proposition 1 shows that the score of the optimal trajectory distribution under linearly scalarized rewards decomposes as a weighted combination of the individual scores corresponding to the reward-specific optimal distributions. Each $P_i^\star$ denotes the optimal trajectory distribution for maximizing reward $\mathbf{r}_i$ while remaining close to the reference distribution. The proof of Proposition 1 is provided in the Appendix.

Motivated by this structure, we model a separate score function $\epsilon_i(\tau)$ maximizing each reward $\mathbf{r}_i$, independently approximating $\nabla_\tau \log P_i^\star(\tau)$. At inference time, these learned scores are combined linearly using the weight vector w to construct a composite score for sampling that is the final scoreing function is given by

$$\epsilon(\tau) = \sum_{i=1}^{m} w_i \epsilon_i(\tau) \tag{4}$$

This modular design enables flexible control over preferences without retraining. Moreover, new objectives can be incorporated by training an additional score model independently, without affecting existing components.

This modularity marks a key distinction between our framework and prior approaches using classifier-free conditional sampling (Ho and Salimans 2022), which rely on a fixed interpolation between unconditional and conditional score predictions:

$$\hat{\epsilon} := \epsilon_\theta(x_k(\tau), \varnothing) + \gamma\left(\epsilon_\theta(x_k(\tau), w_i) - \epsilon_\theta(x_k(\tau), \varnothing)\right), \tag{5}$$

where w_i denotes conditioning information, such as preference weights, and $\varnothing$ denotes the unconditional setting. The scalar $\gamma \in [0, 1]$ controls the strength of conditioning during generation.

4.2 Practical Instantiation: Diffusion-Based Offline MORL Planning

Diffusion models have recently been applied in offline RL to model trajectory distributions from an offline dataset (Ajay et al. 2022; Lu et al. 2025). For planning, they generate sequences of future states, $x(\tau) := (s_t, s_{t+1}, \ldots, s_{t+N})$, by denoising corrupted trajectories. To learn the optimal model P_i for each objective, we propose a two-state training procedure, as illustrated in Figure 1. The complete algorithm is shown in Algorithm 1.

Pre-training We first train a diffusion model P_{ref} using the entire offline dataset to learn a general-purpose planner with the surrogate loss in Equation 2. We refer this model P_{ref} as the *reference model*, which captures the trajectory distribution of behavior policies without any preference bias.

DPO-based Fine-Tuning In the second stage, we fine-tune separate diffusion models for each objective P_i using pairwise trajectory preference alignment from the pairwise dataset via DPO.

We adopt a diffusion DPO from image generation Wallace et al. (2024) to the trajectory planning task with Equation 3. The pairwise preference dataset D_p is constructed from the offline dataset D by randomly sampling N pairs of sub-trajectories. Each entry in D_p is denoted as $(\tau_i, \tau_j, \boldsymbol{\ell})$, where τ_i and τ_j are sub-trajectories extracted from different trajectories in D, and $\boldsymbol{\ell}$ is a vector of preference labels, one for

Algorithm 1: Diffusion-Based Offline Multi-Objective RL Planning

Input: Offline dataset $\mathcal{D}$, pairwise preference dataset $\mathcal{D}_p$, planning stride m, planning horizon $\mathcal{H}$
Initialize: Diffusion planner P_θ, inverse dynamics model f_ϕ
function PRETRAINING
    Sample state and action sequences $\left(s_{t:t+m:(t+(\mathcal{H}-1)m)}, a_{t:t+m:(t+(\mathcal{H}-1)m)}\right)$ from $\mathcal{D}$
    Train diffusion model ϵ_θ using Eq. (2)
    Train inverse dynamics model f_ϕ to minimize $||a_t - f_\phi(s_t, s_{t+m})||$
end function
function FINETUNING
    Sample preference pairs $(\tau_0^{(+)}, \tau_0^{(-)})$ from $\mathcal{D}_p$
    Fine-tune each objective-specific planner ϵ_{θ_i} using DPO (Eq. (3))
end function
function INFERENCE
    Generate plan using each fine-tuned ϵ_{θ_i} for corresponding objective
    Combine plans with preference weights using Eq. (4)
    Use inverse dynamics f_ϕ to obtain actions and simulate forward
end function

each objective. These labels are determined by comparing the cumulative rewards of the two sub-trajectories with respect to each objective. For instance, for the objective *speed*, the sub-trajectory with the higher average speed is considered preferable and assigned the corresponding label in $\boldsymbol{\ell}$. A preferred trajectory is labeled as one and the less preferred one as zero.

Inference-Time Composition To realize the optimized trajectory distribution $P^\star(\tau)$ in practice, we employ a modular diffusion model, where each denoising component ϵ_{θ_i} is trained independently to model the distribution $P_i^\star$ associated with reward component $\mathcal{R}_i$. At inference time, we compose these modules based on the user-specified preference vector $w \in \mathbb{R}^m$.

Specifically, at each timestep, the scoring function used for generation (as in Equation 4) is expressed as:

$$x_{t-1} = x_t - \eta_t \cdot \left(\sum_{i=1}^{m} w_i \cdot \epsilon_{\theta_i}(x_t, t) \right),$$

where η_t is a step-size term determined by the noise schedule. w_i represents the preference weight for objective i, and ϵ_{θ_i} is the noise prediction model trained for the i-th objective.

To demonstrate the flexibility of our framework, we evaluate it across two distinct domains: offline MORL and offline safe RL. While these benchmarks differ in their specific goals, they share a common structure that aligning policy behavior with multiple, potentially conflicting objectives. In offline MORL, the objective is to align policies with user-defined preference weights (w_i, w_j), whereas offline safe RL aims to satisfy both reward and cost constraints ($R(\tau)$, $C(\tau)$) of the trajectories.

As shown in Table 1, existing diffusion-based planning frameworks such as MODULI (Yuan et al. 2024) and OASIS (Yao et al. 2024) require explicit conditioning inputs, $[w_i, w_j]$ regarding preferences or $[C(\tau), R(\tau)]$ for cost and reward during both training and inference sampling. In contrast, our approach treats preference weights as scalar coefficients and applies them by linearly combining the outputs of each objective model. The advantage is that if an objective is unknown or irrelevant to the user, its corresponding weight w_i can simply be set to zero. Conversely, new objective-specific models can be directly integrated by assigning them a non-zero weight.

Inverse Dynamics Model Since diffusion planning generates future states, interacting with the environment at inference time requires a separately trained inverse dynamics model f_ϕ. This model predicts the corresponding action needed to transition from a current state s_t to a future state s_{t+m} using a step stride m (Lu et al. 2025):

$$a_t \leftarrow f_\phi(s_t, s_{t+m}).$$

The predicting actions are used to execute behavior in the environment. The inverse dynamics model f_ϕ is also trained using the same offline data.

5 Experiments

In this section, we present empirical results on two benchmarks: the *Offline MORL* benchmark and the *Offline Safe RL* benchmark.

5.1 Offline MORL Experiments

Offline MuJoCo Dataset The offline MORL dataset (D4MORL) was created by Zhu, Dang, and Grover (2023) for the MuJoCo environment. It includes several continuous control tasks based on the MuJoCo simulator, each with multiple objectives. For example, the tasks *MO-Swimmer*, *MO-HalfCheetah*, and *MO-Walker2D* have two main objectives: speed and energy efficiency. These objectives are inherently conflicting: achieving higher speed typically increases energy consumption, while minimizing energy use often leads to slower movement. In the following experiments, we use the Expert dataset, which contains actions generated by an optimal reference policy, representing high-quality demonstrations.

Method	Diffusion-based Training	Diffusion-based Generation
MODULI (Offline MORL)	$\epsilon_\theta\big(x(\tau), [w_i, w_j], k\big)$	$\epsilon_\theta\big(x(\tau), [w_i, w_j], k\big)$
OASIS (Offline safe RL)	$\epsilon_\theta\big(x(\tau), [C(\tau), R(\tau)], k\big)$	$\epsilon_\theta\big(x(\tau), [C(\tau), R(\tau)], k\big)$
Ours	$\epsilon_{\theta_i}\big(x(\tau), k\big)$	$\sum w_i\, \epsilon_i\big(x(\tau)\big)$

Table 1: Comparison of conditioning inputs during training and inference. MODULI and OASIS condition on preference weights ($[w_i, w_j]$) or cost-reward signals ($[C(\tau), R(\tau)]$) in both phases. In contrast, our method requires no conditioning information. Instead, it performs a linear combination over samples from each objective at inference time.

Method	Ant ($\times 10^6$)	HalfCheetah ($\times 10^6$)	Swimmer ($\times 10^4$)	Walker2d ($\times 10^6$)
MODT	5.52±.16	5.59±.03	2.49±.19	0.65±.46
MORvS	5.52±.02	4.19±.74	3.19±.01	5.10±.02
BC	0.84±.60	1.53±.09	1.68±.38	0.07±.02
CQL	3.52±.45	3.78±.46	2.08±.08	0.82±.62
Ours	6.21±.02	5.71±.19	3.17±.00	5.16±.00

Table 2: Comparison of Hypervolume (HV) scores for offline RL methods trained without preference information. Higher HV scores indicate better performance.

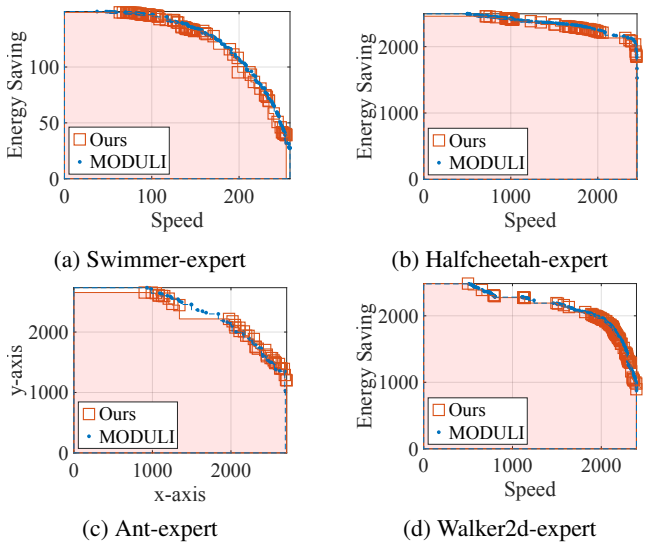
Metrics In MORL, the goal is not to learn a single optimal policy but a set of trade-off solutions known as the *Pareto set*. Each point on this front represents a solution where improving one objective would worsen at least one other.

To evaluate the quality of the solutions obtained, we use the *hypervolume (HV)* indicator (Emmerich, Beume, and Naujoks 2005), which quantifies the volume in objective space that is dominated by a set of solutions and bounded by a reference point V_{ref}. A higher HV indicates better coverage, diversity, and quality of trade-offs across objectives.

For evaluation, all experiments are evaluated across 501 preference weights of the form $(w, 1-w)$, where $w \in [0, 1]$. The reference point for the HV calculation is set to coordinate zero for each objective.

Results In Table 2, we compare our method with prior approaches that also do not use preference information during training (Zhu, Dang, and Grover 2023). These baselines include CQL (Kumar et al. 2020), a state-of-the-art offline RL algorithm; behavioral cloning (BC), a standard imitation learning method; and decision-transformer-based methods (Chen et al. 2021), including a modified multi-objective version of Decision Transformer (MODT) and MORvs (Emmons et al. 2021). All these methods are evaluated by scalarizing the objective return vectors using weighted sums. Unlike these approaches, which scale reward vectors by the weights during inference, our method incorporates the weights directly into the sampling process via linear multiplication, enabling more effective preference alignment. As a result, our method achieves higher HV scores, indicating improved coverage and diversity along the objective space.

Figure 2 compares HV metrics between our method and MODULI, both diffusion-based planning frameworks. Despite not using preference information during training, our method achieves performance comparable to MODULI. In *HalfCheetah* and *Walker2D*, HV scores are similar with largely overlapping Pareto frontiers. In *Ant*, we observe a slight drop in HV, indicating reduced solution diversity. This gap is likely due to MODULI seeing preference weights during evaluation that are similar or even identical to those seen during training, effectively benefiting from supervised signals. It is also important to note that while HV is a widely used indicator of coverage in the objective space, it does not fully capture the core advantage of our framework, which is its adaptability to real-world applications where user preferences may be dynamic, context-dependent, or unavailable during training. This practical usage highlights our approach beyond what HV alone can measure.



(a) Swimmer-expert (b) Halfcheetah-expert

(c) Ant-expert (d) Walker2d-expert

Figure 2: Comparison of HV between MODULI and our method. We have largely similar pareto front.The overlapping HV regions indicate that our method achieves competitive performance, even without access to preference weights during training.

5.2 Offline Safe RL Experiments

In safe RL domains, RL agents must optimize not only for task performance but also for safety constraints. This naturally lends itself to a MORL formulation, where one objective corresponds to maximizing reward and the other to minimizing safety-related cost.

Algorithm	Stats	BallRun	CarRun	DroneRun	BallCircle	CarCircle	DroneCircle
BC	*reward* ↑	0.55 ± 0.23	0.94 ± 0.02	0.62 ± 0.11	0.73 ± 0.05	0.59 ± 0.11	0.82 ± 0.01
	cost ↓	2.04 ± 1.32	1.50 ± 1.11	3.48 ± 0.68	2.53 ± 0.15	3.39 ± 0.85	3.29 ± 0.18
CDT	reward ↑	0.35 ± 0.01	*0.96 ± 0.01* †	0.84 ± 0.12	0.73 ± 0.01	0.71 ± 0.01	0.17 ± 0.08
	cost ↓	1.56 ± 1.10	**0.67 ± 0.03**	7.56 ± 0.33	1.36 ± 0.03	2.39 ± 0.15	1.08 ± 0.62
FISOR	reward ↑	*0.17 ± 0.03*	0.85 ± 0.02	0.44 ± 0.14	*0.28 ± 0.03*	*0.24 ± 0.05*	*0.49 ± 0.05*
	cost ↓	**0.04 ± 0.06**	0.15 ± 0.20	2.52 ± 0.61	**0.00 ± 0.00**	**0.15 ± 0.27**	**0.02 ± 0.03**
OASIS	*reward* ↑	*0.28 ± 0.01*	*0.85 ± 0.04*	*0.13 ± 0.08*	*0.70 ± 0.01*	*0.76 ± 0.03*†	*0.60 ± 0.01*†
	cost ↓	**0.79 ± 0.37**	**0.02 ± 0.03**	**0.79 ± 0.54**	**0.45 ± 0.14**	**0.89 ± 0.59**	**0.25 ± 0.10**
Our Results	*reward* ↑	*0.30 ± 0.03*†	*0.90 ± 0.003*	*0.27 ± 0.01*†	*0.70 ± 0.00*†	*0.42 ± 0.00*	*0.16 ± 0.00*
	cost ↓	**0.92 ± 0.22**	**0.01 ± 0.01**	**0.00 ± 0.00**	**0.94 ± 0.03**	**0.75 ± 0.10**	**0.76 ± 0.01**

Table 3: Normalized reward and cost across tasks. Reward ↑: higher is better; Cost ↓: lower than one is preferred. **Bold** highlights safe agents (cost < 1). † denotes the highest reward that satisfies the safety constraint. Results are averaged over three seeds

Offline Safe RL Environments We evaluate six tasks from the Bullet-Safety-Gym benchmark, organized into two categories: *Circle Tasks* and *Run Tasks*. These tasks challenge agents to complete navigation objectives while minimizing safety violations, such as leaving a safe region or colliding with obstacles. For offline RL training, we use the DSRL dataset introduced by Liu et al. (Liu et al. 2023a).

Metrics Agent performance is evaluated using two key metrics: reward and cost. Higher rewards indicate better task performance, while the cost indicates the degree to which the agent violates safety constraints (Yao et al. 2024; Zheng et al. 2024). To evaluate safety consistently, we use a normalized cost, where values below one indicate that the agent remains within acceptable safety parameters. During evaluation, we compare rewards only among agents that satisfy the safety criterion, where cost value below the threshold of one.

Results During inference, we evaluated various combinations of reward-cost weights to assess how well different trade-offs balance performance and safety, without requiring any retraining. We randomly generate weight pairs such as $(0.1, 0.9)$, $(0.5, 0.5)$, and $(0.6, 0.4)$, using a uniform linear spacing over the interval $[0, 1]$. We evaluate each weight and select the weight combination that satisfies the cost constraint and yields the highest reward. This approach is straightforward to use and avoids the complexity of constrained RL algorithms that require Lagrange multiplier updates, such as Lagrangian-based methods.

In Table 3, we compare with the sequential modeling framework: CDT (Liu et al. 2023b) and FISOR (Zheng et al. 2024). We can notice that our model ensures safety across all evaluated environments. In the *OfflineDroneRun-v0* task, with reward and cost weights set to 0.1 and 0.9 respectively, our method achieves a cost of exactly 0.00 while maintaining a competitive reward of 0.27, outperforming many safety-focused baselines. Similarly, in the *OfflineCarRun-v0* task, our model attains a near-optimal reward of 0.90 while keeping the cost extremely low at 0.01.

More importantly, OASIS generates trajectories using diffusion models and then applies standard RL algorithms with reward signals to learn the policy. In contrast, our method eliminates the need for imitated reward and cost signals at each time step by directly optimizing the model using DPO, providing a significant advantage.

5.3 Limitations

Our proposed framework builds on prior work in diffusion-based decision making, and it inherits several of its limitations (Ajay et al. 2022). First, the model faces challenges in environments with high stochasticity or discrete action spaces. Hybrid continuous–discrete diffusion can help handle structured sparsity in trajectories or features (Ostheimer et al. 2025). Furthermore, the core methodology relies on learning the trajectory distribution from a dataset, which is assumed to be clean and reliable. It struggles when the dataset is imperfect, for example, when demonstrations contain noise or adversarial perturbations (Mustafa et al. 2024; Mustafa, Vandermeulen, and Kloft 2020)—causing actions to deviate from the intended policy.

From a societal perspective, while we want the agent to be flexible and align with user preferences based on context, there are potential safety risks if such capabilities are misused. Ensuring the safe and responsible deployment of such systems remains an essential direction for future work.

6 Conclusion and Future Work

A central challenge in AI alignment is ensuring that agents not only optimize specified objectives but also remain adaptable to evolving human preferences (Ji et al. 2023). In this paper, we propose TORA, a novel framework that enables preference adaptation at inference time rather than during training. This provides a significant practical advantage: since each objective is optimized independently, objectives can be easily added or removed in response to dynamic changes in user preferences.

Future work includes extending TORA's modular alignment framework to online RL and real-world applications, such as robotic manipulation and molecular design (Xue et al. 2024), as well as to domains with physical or structural constraints, inspired by approaches like SetPINNs and PIANO (Nagda et al. 2024, 2025).

Acknowledgements

MK and SF acknowledge support by the DFG through SPP 2331 (441958259, 553345933, 466468799), the BMFTR award 01IS24071A, by the DFG through FOR 5359 (ID 459419731), TRR 375 (ID 511263698), SPP 2298 (Id 441826958), and by the Carl-Zeiss Foundation through the initiatives AI-Care and Process Engineering 4.0. The work of PW is partially supported by the Alexander von Humboldt Foundation.

References

Abels, A.; Roijers, D.; Lenaerts, T.; Nowé, A.; and Steckelmacher, D. 2019. Dynamic weights in multi-objective deep reinforcement learning. In *International conference on machine learning*, 11–20. PMLR.

Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J.; Jaakkola, T.; and Agrawal, P. 2022. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Emmerich, M.; Beume, N.; and Naujoks, B. 2005. An EMO algorithm using the hypervolume measure as selection criterion. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, EMO'05, 62–76. Berlin, Heidelberg: Springer-Verlag. ISBN 3540249834.

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*.

Felten, F.; Alegre, L. N.; Nowe, A.; Bazzan, A.; Talbi, E. G.; Danoy, G.; and C da Silva, B. 2023. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 23671–23700.

Hayes, C. F.; Rădulescu, R.; Bargiacchi, E.; Källström, J.; Macfarlane, M.; Reymond, M.; Verstraeten, T.; Zintgraf, L. M.; Dazeley, R.; Heintz, F.; et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1): 26.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Honari, H.; Tamizi, M. G.; and Najjaran, H. 2024. Safety optimized reinforcement learning via multi-objective policy optimization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2873–2879. IEEE.

Hong, K.; Li, Y.; and Tewari, A. 2024. A primal-dual-critic algorithm for offline constrained reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 280–288. PMLR.

Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Zhou, J.; Zhang, Z.; et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33: 1179–1191.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Li, W.; Devidze, R.; Mustafa, W.; and Fellenz, S. 2024. Ethics in action: training reinforcement learning agents for moral decision-making in text-based adventure games. In *International Conference on Artificial Intelligence and Statistics*, 1954–1962. PMLR.

Lin, Q.; Tang, B.; Wu, Z.; Yu, C.; Mao, S.; Xie, Q.; Wang, X.; and Wang, D. 2023. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, 21127–21152. PMLR.

Lin, Q.; Yu, C.; Liu, Z.; and Wu, Z. 2024. Policy-regularized offline multi-objective reinforcement learning. *arXiv preprint arXiv:2401.02244*.

Liu, E.; Wu, Y.-C.; Huang, X.; Gao, C.; Wang, R.-J.; Xue, K.; and Qian, C. 2025. Pareto set learning for multi-objective reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18789–18797.

Liu, Z.; Guo, Z.; Lin, H.; Yao, Y.; Zhu, J.; Cen, Z.; Hu, H.; Yu, W.; Zhang, T.; Tan, J.; et al. 2023a. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*.

Liu, Z.; Guo, Z.; Yao, Y.; Cen, Z.; Yu, W.; Zhang, T.; and Zhao, D. 2023b. Constrained decision transformer for offline safe reinforcement learning. In *International conference on machine learning*, 21611–21630. PMLR.

Lu, H.; Han, D.; Shen, Y.; and Li, D. 2025. What Makes a Good Diffusion Planner for Decision Making? In *The Thirteenth International Conference on Learning Representations*.

Mustafa, W.; Liznerski, P.; Ledent, A.; Wagner, D.; Wang, P.; and Kloft, M. 2024. Non-vacuous generalization bounds for adversarial risk in stochastic neural networks. In *International conference on artificial intelligence and statistics*, 4528–4536. PMLR.

Mustafa, W.; Vandermeulen, R. A.; and Kloft, M. 2020. Input hessian regularization of neural networks. *arXiv preprint arXiv:2009.06571*.

Nagda, M.; Abijuru, J.; Ostheimer, P.; Kloft, M.; and Fellenz, S. 2025. PIANO: Physics Informed Autoregressive Network. *arXiv preprint arXiv:2508.16235*.

Nagda, M.; Ostheimer, P.; Specht, T.; Rhein, F.; Jirasek, F.; Mandt, S.; Kloft, M.; and Fellenz, S. 2024. Setpinns:

Set-based physics-informed neural networks. *arXiv preprint arXiv:2409.20206*.

Ostheimer, P.; Nagda, M.; Radig, J.; Herrmann, C.; Mandt, S.; Kloft, M.; and Fellenz, S. 2025. Sparse Data Generation Using Diffusion Models. *arXiv preprint arXiv:2502.02448*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Rame, A.; Couairon, G.; Dancette, C.; Gaya, J.-B.; Shukor, M.; Soulier, L.; and Cord, M. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36: 71095–71134.

Shi, R.; Chen, Y.; Hu, Y.; Liu, A.; Hajishirzi, H.; Smith, N. A.; and Du, S. 2024. Decoding-time language model alignment with multiple objectives. *URL https://arxiv.org/abs/2406.18853*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations (ICLR)*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stooke, A.; Achiam, J.; and Abbeel, P. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, 9133–9143. PMLR.

Tang, Z.; Peng, J.; Tang, J.; Hong, M.; Wang, F.; and Chang, T.-H. 2024. Inference-Time Alignment of Diffusion Models with Direct Noise Optimization. *arXiv preprint arXiv:2405.18881*.

Towers, M.; Kwiatkowski, A.; Terry, J.; Balis, J. U.; De Cola, G.; Deleu, T.; Goulão, M.; Kallinteris, A.; Krimmel, M.; KG, A.; et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032*.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8228–8238.

Xue, K.; Tan, R.-X.; Huang, X.; and Qian, C. 2024. Offline multi-objective optimization. *arXiv preprint arXiv:2406.03722*.

Yang, R.; Sun, X.; and Narasimhan, K. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32.

Yao, Y.; Cen, Z.; Ding, W.; Lin, H.; Liu, S.; Zhang, T.; Yu, W.; and Zhao, D. 2024. Oasis: Conditional distribution shaping for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37: 78451–78478.

Yeh, P.-H.; Lee, K.-H.; and Chen, J.-C. 2024. Training-free diffusion model alignment with sampling demons. *arXiv preprint arXiv:2410.05760*.

Yuan, Y.; Zheng, Z.; Dong, Z.; and Hao, J. 2024. MODULI: Unlocking Preference Generalization via Diffusion Models for Offline Multi-Objective Reinforcement Learning. *arXiv preprint arXiv:2408.15501*.

Zhang, L.; Zhang, Q.; Shen, L.; Yuan, B.; Wang, X.; and Tao, D. 2023. Evaluating model-free reinforcement learning toward safety-critical tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 15313–15321.

Zheng, Y.; Li, J.; Yu, D.; Yang, Y.; Li, S. E.; Zhan, X.; and Liu, J. 2024. Safe offline reinforcement learning with feasibility-guided diffusion model. *arXiv preprint arXiv:2401.10700*.

Zhu, B.; Dang, M.; and Grover, A. 2023. Scaling pareto-efficient decision making via offline multi-objective rl. *arXiv preprint arXiv:2305.00567*.