GEAR: Bridging the Planner-Actor Gap via Gradient-Aligned Policy Extraction

Yao-Hui Li[1] **Xin Li**[1*] **Hasnaa Bennis**[1] **Meijiu Li**[1]
Boya Zhang[2] **Yingfang Yuan**[3,4] **Wei Pang**[3]

[1]Beijing Institute of Technology, Beijing, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[3]Heriot-Watt University, Edinburgh, U.K. [4]Northumbria University, Newcastle upon Tyne, U.K.

Abstract

Hybrid model-based reinforcement learning (MBRL) integrates lookahead planning with actor-critic optimization for exceptional sample efficiency. However, this decoupled architecture suffers from a critical planner-actor mismatch. Existing policy alignment methods face a structural dilemma: forward KL triggers mode-covering objective conflicts, while reverse KL relies on brittle proxies that bottleneck expressivity. To address this challenge, we propose **G**radient-**E**mbedded **A**lignment **R**egularization (**GEAR**), a unified in-sample extraction framework. At its core, we derive a first-order geometric alignment metric from the continuous-time Hamilton-Jacobi-Bellman (HJB) equation to evaluate action evolution against the local value gradient. By embedding this metric as an absolute modulation weight, GEAR eliminates spurious imitation and transforms the inherently diffuse forward KL into a focused, mode-seeking objective. Evaluations on the challenging HumanoidBench suite demonstrate GEAR achieves significantly higher sample efficiency and asymptotic performance than state-of-the-art baselines. The project code is available at https://github.com/Arya87/GEAR.

1 Introduction

Model-based reinforcement learning (MBRL) with online planning has demonstrated remarkable data efficiency across various domains, ranging from discrete games [1, 2, 3], high-dimensional continuous control [4, 5, 6, 7, 8] and large-scale multi-task settings [9, 10, 11, 12]. By learning a predictive world model of latent dynamics and rewards, these methods enable internal simulation and lookahead search. This capability significantly enhances sample efficiency compared to purely model-free methods, which must learn policies strictly through trial and error.

However, relying solely on online planning introduces compounding errors and heavy computational burdens in long-horizon tasks. To overcome these limitations, hybrid frameworks that tightly integrate model predictive control (MPC) with model-free actor-critic learning have emerged as a highly effective paradigm [5, 6, 10]. By utilizing learned functions to guide and truncate the planning horizon, these methods achieve a critical balance, retaining world-model sample efficiency while ensuring robust asymptotic performance of model-free RL. Despite this success, such hybrid architectures often suffer from an underexplored issue internally: ***planner-actor mismatch***. Environment interaction is driven by a model-based planner, whereas the learning process relies on a separate actor-critic to estimate terminal values and construct training targets. This decoupled mechanism, while empirically effective, introduces an inherent discrepancy between the executed planning behavior and the optimized policy.

*Correspondence to: Xin Li <xinli@bit.edu.cn>

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

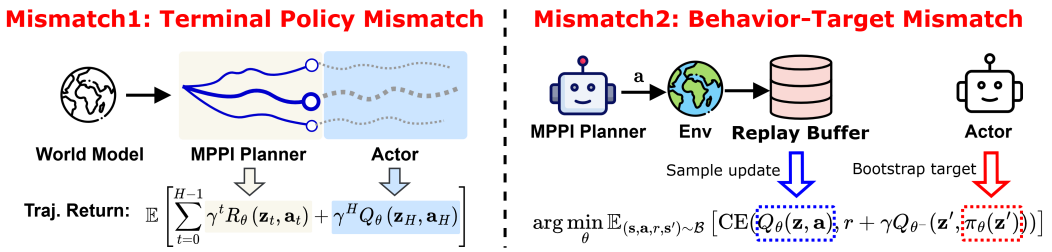

Figure 1: **Illustration of the two-level policy mismatch in MBRL.** *(Left)* Bootstrapping bias: truncating planner trajectories with actor-based value estimates. *(Right)* Behavior-target mismatch: the planner action populates the buffer while the actor constructs TD targets.

More specifically, as illustrated in Figure 1, this mismatch arises at two levels. First, during planning, to avoid the expensive computational cost and the accumulation of model errors associated with infinite-horizon rollouts, the prefix of a trajectory is generated by the model-based planner, whereas the terminal value is typically estimated by the critic through the Q-value of the actor's action at the final state. Such a bootstrap implicitly assumes that the planned trajectory will naturally hand over to the actor policy at the horizon boundary. However, in challenging tasks, the actor is often weaker than the planner [13], leading to biased terminal value estimates. Second, during critic updates, TD targets are commonly constructed using the actor, while the replay data are primarily generated by the planner. As a result, the data distribution is induced by the planner, while the learning signal is partly governed by the actor, creating an inconsistency between policy evaluation and policy improvement.

To mitigate this mismatch, recent works have attempted to explicitly align the actor with the planner. BMPC [13] distills the online Model Predictive Path Integral (MPPI) [14] planner via Kullback-Leibler (KL) divergence. While this reduces the planner-actor gap, it requires frequent re-planning to provide up-to-date teacher signals and largely turns actor learning into behavior distillation. As a result, the actor becomes more of an approximate replica of the planner, rather than an improved policy that integrates both behavioral and value information. Other approaches seek to balance behavior imitation with value guidance by augmenting the standard Q-gradient actor loss with planner-guided regularization. BOOM [15] incorporates a forward KL-style in-sample imitation term on planner-generated data, whereas TD-M(PC)^2 [16] adopts a TD3+BC-style [17] reverse KL regularizer. While these methods successfully avoid frequent online replanning, they present a critical structural dilemma. To evaluate the reverse KL, TD-M(PC)^2 requires an analytical density. Since the sample-based, non-parametric planner cannot provide this, the method is forced to rely on an approximate parametric proxy, inherently biasing the extraction. Conversely, BOOM's forward KL offers a clean, likelihood-free alternative, but naively appending it to the standard mode-seeking Q-loss triggers a fundamental objective conflict, resulting in spurious, mode-covering imitation.

In this paper, we propose **G**radient-**E**mbedded **A**lignment **R**egularization (**GEAR**), a unified, value-modulated extraction framework that retains the likelihood-free elegance of forward KL while entirely eliminating its objective conflict. Instead of combining disjoint penalties or brittle proxy distributions, our approach builds upon a first-order geometric alignment metric derived from the continuous-time Hamilton-Jacobi-Bellman (HJB) equation. This foundation allows us to rigorously evaluate action evolution against the local value gradient. By embedding this metric as a geometric modulation weight and anchoring it on an absolute gating mechanism rather than relative batch assignments, we eliminate the risk of spurious imitation. This design organically drives the diffuse forward KL objective toward focused, mode-seeking behavior, ensuring stable in-sample policy improvement with strictly aligned guidance.

Empirically, we evaluate our proposed method on HumanoidBench [18], a challenging suite of continuous control tasks featuring highly complex, high-dimensional humanoid robots. Experimental results demonstrate that our gradient-aligned extraction achieves both significantly higher sample efficiency and better asymptotic performance compared to existing SOTA MBRL methods.

2 Related Work

Planning-driven MBRL. Planning-driven MBRL scales to high-dimensional control by tightly coupling lookahead search with parametric policy learning [20, 21, 22]. While methods vary in

Table 1: **Comparison of policy extraction methods in Hybrid MBRLs.** To highlight the core differences in alignment mechanisms, standard actor objectives (e.g., Q-maximization) are omitted. Unlike forward KL methods that suffer from conflicting mode-covering penalties, and reverse KL methods that rely on brittle proxy distributions, GEAR leverages first-order value geometry for value-aware, in-sample policy extraction.

Method	Objective Paradigm	Core Alignment Formulation	Sampling Support
BMPC [13]	Online Distillation (Reverse KL)	$\min_\theta \mathbb{E}_{\mathbf{z}\sim\mathcal{B}}\Big[\mathcal{D}_{\text{KL}}\big(\pi_\theta(\cdot\|\mathbf{s}) \parallel \hat{\Pi}_{\text{replan}}(\cdot\|\mathbf{s})\big)\Big]$	Online Planner ($\hat{\Pi}_{\text{replan}}(\cdot\|\mathbf{s})$)
BOOM [15]	Forward KL	$\max_\theta \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{B}}\Big[\exp\big(\frac{Q_\theta(\mathbf{s},\mathbf{a})}{\tau}\big)\log\pi_\theta(\mathbf{a}\|\mathbf{s})\Big]$	Replay Buffer ($\mathbf{a}\sim\mathcal{B}$)
TD-M(PC)^2 [16]	Reverse KL	$\max_\theta \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\theta(\cdot\|\mathbf{s})}\big[Q_\theta(\mathbf{s},\mathbf{a})\big]$ s.t. $\mathcal{D}_{\text{KL}}\big(\pi_\theta(\cdot\|\mathbf{s}) \parallel \hat{\Pi}_{\text{elite}}(\cdot\|\mathbf{s})\big) \le \epsilon$	Actor Policy ($\mathbf{a}\sim\pi_\theta$)
MRS.Q [19]	Massive Q-Ensemble	$\max_\theta \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\theta(\cdot\|\mathbf{s})}\Big[\min_{i=1..10} Q_i(\mathbf{s},\mathbf{a})\Big]$	Actor Policy ($\mathbf{a}\sim\pi_\theta$)
GEAR(Ours)	**Gradient-Aligned In-Sample Extraction**	$\max_\theta \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{B}}\Big[\Big[\underbrace{A^{(0)}}_{\text{0-order}} + \underbrace{\lambda\mathbb{1}_{A^{(0)}>0}A^{(1)}}_{\text{1st-order}}\Big]_+^2 \log\pi_\theta(\mathbf{a}\|\mathbf{s})\Big]$	**Replay Buffer** ($\mathbf{a}\sim\mathcal{B}$)

their coupling mechanisms, ranging from policy-constrained sampling [23] to joint latent dynamics learning [5, 10], they position the actor as an informed prior to accelerate planning. However, this structural integration inevitably induces planner-actor mismatch, a fundamental behavioral mismatch between the non-parametric planner and the parametric actor. To suppress the resulting OOD overestimation, MRS.Q [19] penalizes mismatched actions using a massive pessimistic Q-ensemble. Alternatively, alignment-based methods tackle the mismatch directly: BMPC [13] purely imitates the planner by discarding explicit policy optimization, whereas TD-M(PC)^2 [16] enforces a reverse KL constraint against a Gaussian proxy fitted to elite samples. Most related to our work, BOOM [15] employs a Q-weighted forward KL alignment to preserve distributional consistency without relying on brittle proxies. However, BOOM naively appends this likelihood-free imitation as an additive penalty. This structural disconnect triggers objective conflict with the standard mode-seeking Q-loss, leaving the inherent mode-covering pathology of forward KL unresolved. In contrast, our approach structurally resolves this conflict by embedding the critic's gradient as a modulation weight, enabling focused mode-seeking extraction. Detailed comparison is summarized in Table 1.

Offline Policy Extraction. Although planning-driven MBRL operates online, optimizing the actor from the planner's buffer poses a fundamental offline extraction challenge: evaluating the Q-function on the actor's generated, unseen actions triggers OOD overestimation. To constrain these updates safely, it is natural to adopt explicit policy regularization of offline RL [17, 24], importance sampling [25] and in-sample learning [26, 27]. Implicit Value Regularization (IVR) [28] introduces a behavior penalty into the reward, modifying the Bellman operator to evaluate only dataset actions and strictly avoid extrapolation errors. Although IVR ensures in-sample safety here, it relies entirely on zero-order evaluations, ignoring local gradient geometry. To inject directional guidance, UniRL [29] extends IVR by matching the in-sample gradient with the standard policy gradient. However, this requires a complex auxiliary neural network to minimize gradient distances, destabilizing training. In contrast, our approach directly embeds the critic's first-order gradient as a simple modulation weight, achieving focused extraction and strict in-sample safety without relying on any auxiliary networks.

3 Preliminaries

3.1 Problem Formulation

We consider the problem formulated as an infinite-horizon Markov Decision Process (MDP), characterized by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, \mathcal{P}, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, respectively. The environment dynamics are governed by the transition probability $\mathcal{P}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, while the agent receives a feedback signal defined by the reward function $r(\mathbf{s}_t, \mathbf{a}_t) \in \mathbb{R}$. $\gamma \in [0, 1)$ serves as the discount factor balancing immediate and future returns. Starting from an initial state $\mathbf{s}_0 \in \mathcal{S}$, the fundamental objective of an RL agent is to learn a policy $\pi_\theta(\cdot|\mathbf{s}_t)$, parameterized by θ, to maximize the expected discounted return. The optimal policy π^* is therefore defined as $\pi^* = \arg\max_\pi \mathbb{E}_{\mathbf{a}_t\sim\pi(\cdot|\mathbf{s}_t),\mathbf{s}_t\sim\mathcal{P}}\Big[\sum_{t=1}^{T}\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\Big]$.

3.2 Model-based Planning

We build our method upon **TD-MPC2** [10], a scalable plan-based MBRL framework. While TD-MPC2 is capable of large-scale multi-task learning, we omit the task embedding formulation and focus on its single-task setting for clarity. The algorithm learns a latent-space world model parameterized by θ, comprising five core components: an encoder $\mathbf{z} = h_\theta(\mathbf{s})$ mapping environment states to latent representations; a latent dynamics model $\mathbf{z}' = d_\theta(\mathbf{z}, \mathbf{a})$ predicting future latent states; a reward predictor $\hat{r} = R_\theta(\mathbf{z}, \mathbf{a})$; a Q-value estimator $\hat{q} = Q_\theta(\mathbf{z}, \mathbf{a})$; and a prior policy network $\mathbf{a} \sim \pi_\theta(\mathbf{z})$. Similar to standard actor-critic approaches, TD-MPC2 learns the Q-function through iterative Bellman updates, while the policy network is optimized to maximize the expected Q-value alongside an entropy regularization term. For notational simplicity, θ collectively denotes all trainable network parameters.

During environment interaction, TD-MPC2 derives its behavior via MPPI control. Specifically, the planner iteratively samples action sequences over a horizon H from a parameterized Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. The distribution parameters are updated using the top-k trajectories to maximize the expected cumulative return:

$$\mu^*, \sigma^* = \arg\max_{(\mu,\sigma)} \mathbb{E}_{\mathbf{a}_{t:t+H-1} \sim \mathcal{N}(\mu,\sigma^2)} \left[J(\tau) \right], \tag{1}$$

where the trajectory objective $J(\tau)$ evaluates both the simulated intermediate rewards and the terminal state value:

$$J(\tau) = \sum_{h=0}^{H-1} \gamma^h R_\theta(\mathbf{z}_{t+h}, \mathbf{a}_{t+h}) + \gamma^H Q_\theta(\mathbf{z}_{t+H}, \mathbf{a}_{t+H}). \tag{2}$$

By incorporating the learned Q-value function to bootstrap the terminal state, TD-MPC2 effectively truncates the planning horizon, blending the long-term foresight of model-free RL with the sample efficiency of model-based planning.

4 Method

4.1 Revisiting Policy Extraction: Forward vs. Reverse KL

To address the *planner-actor mismatch* in hybrid MBRL, we can directly imitate the actions stored in the replay buffer $\mathcal{B}$, inherently generated by the planner Π, to align behaviors without impairing training efficiency. Moving beyond pure behavioral cloning (e.g., BMPC [13]), recent works seek to balance imitation learning with value guidance by augmenting the standard Q-gradient actor loss with planner-guided regularization. Typically, this extraction is formulated as minimizing either the forward KL [15] or reverse KL [16] divergence.

(1) Forward KL: $\mathcal{D}_{\text{KL}}(\Pi \parallel \pi_\theta)$

Since the exact action likelihood of the sample-based MPPI planner Π is intractable, prior works like BOOM [15] employ a likelihood-free alignment metric. By discarding the intractable planner entropy, the forward KL simplifies into a weighted behavior cloning objective over the replay buffer data $\mathcal{B}$. BOOM integrates this alignment term into the standard maximum-entropy actor objective:

$$\max_\theta \quad \underbrace{\mathcal{J}_{\text{SAC}}(\theta)}_{\text{mode-seeking Q-loss}} \quad + \quad \underbrace{\lambda \mathcal{J}_{\text{F-KL}}(\theta)}_{\text{mode-covering imitation}} \tag{3}$$

where $\mathcal{J}_{\text{SAC}}(\theta) = \mathbb{E}_{\mathbf{s} \sim \mathcal{B}} \Big[\mathbb{E}_{\mathbf{a}_\pi \sim \pi_\theta(\cdot|\mathbf{z})} \big[Q_\theta(\mathbf{z}, \mathbf{a}_\pi) \big] + \alpha \mathcal{H}\big(\pi_\theta(\cdot|\mathbf{z})\big) \Big], \mathbf{z} = h_\theta(\mathbf{s})$, represents the mode-seeking RL objective with entropy $\mathcal{H}$, and the likelihood-free imitation term is defined as:

$$\mathcal{J}_{\text{F-KL}}(\theta) = \mathbb{E}_{\mathbf{s},\mathbf{a} \sim \mathcal{B}} \left[w(\mathbf{z}, \mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z}) \right], z = h_\theta(s), \quad \text{with} \quad w(\mathbf{z}, \mathbf{a}) \propto \exp\big(Q_\theta(\mathbf{z}, \mathbf{a})/\tau\big). \tag{4}$$

While this likelihood-free formulation is elegant, it suffers from two critical limitations. First, the inherently *mode-covering* forward KL penalty strictly conflicts with the *mode-seeking* Q-gradient loss. This forces the unimodal actor to allocate probability mass across multiple conflicting modes, limiting the precision of the extracted policy. Second, the exponential weighting mechanism evaluates actions strictly relative to the sampled batch. If a batch consists entirely of sub-optimal actions, the relatively highest-valued action still receives a dominant effective weight. This relative assignment causes severe weight explosion and spurious imitation of low-quality targets.

(2) Reverse KL: $\mathcal{D}_{\text{KL}}(\pi_\theta \parallel \Pi)$

To enable the use of the reverse KL divergence, methods like TD-M(PC)^2 [16] must explicitly construct a tractable proxy for the non-parametric planner. Specifically, they approximate the planner using a parametric surrogate distribution $\hat{\Pi}_{\text{elite}}(\cdot|\mathbf{z})$, fitted using the top-$k$ elite samples obtained from the MPPI iterations. The actor is then optimized to maximize the expected Q-value, subject to a behavioral constraint against this proxy planner:

$$\max_\theta \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\theta(\cdot|\mathbf{z})}\big[Q_\theta(\mathbf{z},\mathbf{a})\big], \mathbf{z} = h_\theta(\mathbf{s}) \quad \text{s.t.} \quad \mathcal{D}_{\text{KL}}\big(\pi_\theta(\cdot|\mathbf{z}) \parallel \hat{\Pi}_{\text{elite}}(\cdot|\mathbf{z})\big) \leq \epsilon. \tag{5}$$

Unlike forward KL, this objective is natively *mode-seeking*, encouraging the actor to actively locate the peak using the critic's local gradients. However, compressing complex elite samples into a simple Gaussian discards crucial behavioral information. More critically, the high concentration of elite samples frequently causes severe variance collapse in the proxy distribution. This creates an artificially rigid KL boundary that prematurely penalizes exploration and severely destabilizes training via exploding gradients.

This establishes a critical dilemma: Forward KL offers a clean likelihood-free implementation but suffers from objective conflicts and spurious imitation, whereas Reverse KL provides cohesive mode-seeking guidance but relies on brittle proxy distributions. This naturally raises the question: *Can we retain the likelihood-free elegance of forward KL, while replacing the conflicting multi-objective combination with a* ***unified, value-modulated*** *extraction? Furthermore, by anchoring this modulation on an absolute gating mechanism rather than relative batch assignments, can we eliminate spurious imitation and drive a diffuse objective toward focused, mode-seeking behavior?*

4.2 Theoretical Motivation from Continuous-Time HJB

To achieve unified, value-modulated extraction, we seek a geometric signal that complements zero-order value comparisons. Standard value-weighted forward KL methods use scalar action values without explicitly evaluating whether the displacement toward a candidate agrees with the critic's local ascent direction. Moreover, adding an imitation penalty to a Q-maximization objective can produce competing actor updates. This motivates incorporating directional information directly into the extraction weights. We draw inspiration from continuous-time optimal control through the Hamilton–Jacobi–Bellman (HJB) equation.

For theoretical clarity, we formulate the following analysis in the original Markov state space, denoted by $\mathbf{s}$. For a deterministic continuous-time system governed by $\dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{a}(t))$, applying the dynamic programming principle and a first-order Taylor approximation yields the Hamilton–Jacobi–Bellman (HJB) equation for the continuous Q-function [30]:

$$\gamma Q^{\text{HJB}}(\mathbf{s},\mathbf{a}) = \max_{\mathbf{a}\in\mathcal{A}} \left[r(\mathbf{s},\mathbf{a}) + \nabla_{\mathbf{s}} Q^{\text{HJB}}(\mathbf{s},\mathbf{a}) \cdot f(\mathbf{s},\mathbf{a}) + \nabla_{\mathbf{a}} Q^{\text{HJB}}(\mathbf{s},\mathbf{a}) \cdot \dot{\mathbf{a}} \right]. \tag{6}$$

The right-hand side of this equation explicitly reveals that the instantaneous value improvement is governed by the joint dynamics of two factors: the environmental state transitions $\nabla_{\mathbf{s}} Q \cdot f(\mathbf{s},\mathbf{a})$ and the action evolution $\nabla_{\mathbf{a}} Q \cdot \dot{\mathbf{a}}$. Directional alignment leads to more efficient improvement.

Inspired by the HJB equation insight that optimal value improvement requires aligning action evolution with the local gradient, we frame the policy update evaluation as a fundamental question: *given a fixed state* $\mathbf{s}$*, how does shifting a base behavior* $\mathbf{a}_{base}$ *toward a target action* $\mathbf{a}_{target}$ *alter the Q-value?* Since this evaluation occurs at a stationary state, the virtual state displacement is zero ($\dot{\mathbf{s}} = \mathbf{0}$), eliminating the state-transition term from the HJB expansion. The instantaneous value improvement is thus dictated by the action evolution. By parameterizing this action evolution as the displacement $\dot{\mathbf{a}} \approx \mathbf{a}_{\text{target}} - \mathbf{a}_{\text{base}}$, we derive a general first-order geometric alignment metric:

$$A_{\text{align}}(\mathbf{s}, \mathbf{a}_{\text{target}}) = \left\langle \mathbf{a}_{\text{target}} - \mathbf{a}_{\text{base}}, \nabla_{\mathbf{a}_{\text{target}}} Q(\mathbf{s}, \mathbf{a}_{\text{target}}) \right\rangle. \tag{7}$$

Crucially, this alignment metric provides a mechanism for the proposed structural fusion. Rather than appending a mode-covering forward KL penalty to a mode-seeking Q-loss, which can produce conflicting actor updates, we embed first-order Q-gradient information directly into the likelihood-free extraction process. The resulting geometric weighting favors targets aligned with the critic's local steepest ascent and suppresses conflicting ones. This encourages selective extraction among competing modes, integrating imitation and value guidance into a unified objective with focused, mode-seeking behavior. In our latent MBRL implementation, $\mathbf{s}$ is encoded as $\mathbf{z} = h_\theta(\mathbf{s})$, and the action-space alignment metric in Eq. 7 is instantiated using the latent critic.

4.3 Gradient-Aligned In-Sample Policy Extraction

Motivated by the HJB analysis in Section 4.2, we propose **Gradient-Embedded Alignment Regularization**. By fusing a zero-order global advantage with a first-order gradient alignment, we modulate the imitation weights to achieve mode-seeking behavior within a stable, likelihood-free objective.

Zeroth-order Global Advantage. We first construct a zero-order advantage to measure the relative value of a planner action $\mathbf{a}$ compared to the current actor action $\mathbf{a}_\pi \sim \pi_\theta(\cdot|\mathbf{z})$:

$$A^{(0)}(\mathbf{z}, \mathbf{a}) = \frac{Q_\theta(\mathbf{z}, \mathbf{a}) - Q_\theta(\mathbf{z}, \mathbf{a}_\pi)}{\alpha}, \quad \text{with } \mathbf{a}_\pi \sim \pi_\theta(\cdot|\mathbf{z}), \tag{8}$$

where α is a scaling temperature. This isolates actions that are globally preferable but remains inherently blind to the local gradient of the value landscape.

First-order Gradient Alignment. To incorporate the directional guidance derived in Eq. 7, we instantiate the theoretical alignment metric A_{align} by setting the base behavior to the actor's current mean ($\mathbf{a}_{\text{base}} = \boldsymbol{\mu}_\theta(\mathbf{z})$) and the target to the planner action ($\mathbf{a}_{\text{target}} = \mathbf{a}$). This yields the first-order advantage:

$$A^{(1)}(\mathbf{z}, \mathbf{a}) = \langle \mathbf{a} - \boldsymbol{\mu}_\theta(\mathbf{z}), \nabla_{\mathbf{a}} Q_\theta(\mathbf{z}, \mathbf{a}) \rangle \,. \tag{9}$$

A positive $A^{(1)}$ explicitly confirms that pulling the actor toward $\mathbf{a}$ is locally consistent with the critic's direction of steepest ascent. By enforcing this directional consistency, we guarantee that the imitation update inherently drives value improvement, effectively filtering out geometrically conflicting targets that would otherwise pull the policy in divergent directions.

Advantage Modulation and Weighting. We combine these two scales of evaluation to form a modulated advantage:

$$A(\mathbf{z}, \mathbf{a}) = A^{(0)}(\mathbf{z}, \mathbf{a}) + \lambda \cdot \mathbb{1}_{A^{(0)}(\mathbf{z},\mathbf{a})>0} \cdot A^{(1)}(\mathbf{z}, \mathbf{a}), \tag{10}$$

where λ controls the alignment strength. The indicator function ensures that the geometric filter only refines already-promising samples, preventing sub-optimal actions from being promoted merely due to locally favorable slopes. To ensure training stability, we map this advantage to a bounded imitation weight via a truncated polynomial form:

$$w_{\text{align}}(\mathbf{z}, \mathbf{a}) = \left[\text{clip}\left(1 + \frac{A(\mathbf{z}, \mathbf{a})}{2}, 0, w_{\max}\right) \right]^2, \tag{11}$$

where $w_{\max}$ is the clipping upper bound. This formulation keeps the imitation weight bounded to prevent unstable gradients caused by outlier advantages, while assigning zero weight to samples with sufficiently negative modulated advantages.

Unified Value-Modulated Extraction Objective. Crucially, we utilize the critic's local geometry strictly for value modulation, refraining from perturbing the underlying buffer actions. Instead, the actor is optimized by maximizing the gradient-aligned weighted likelihood. By structurally fusing the value modulation into the extraction process, we replace the conflicting multi-objective combination with a single, unified policy extraction target:

$$\mathcal{J}_{\text{align}}(\theta) = \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{B}}\Big[w_{\text{align}}(\mathbf{z}, \mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z}) \Big] + \alpha \mathcal{H}\big(\pi_\theta(\cdot|\mathbf{z})\big), \quad \mathbf{z} = h_\theta(\mathbf{s}) \tag{12}$$

where $\mathcal{H}$ provides the regularization of entropy. This objective resolves the dilemma outlined in Section 4.1: it retains the likelihood-free simplicity of forward KL extraction while equipping it with the cohesive, mode-seeking properties previously exclusive to reverse KL methods.

4.4 Theoretical Analysis

Unlike BMPC [13] requiring active online replanning, our objective extracts the actor π_θ directly using the *historical planner policy* Π stored in the replay buffer $\mathcal{B}$. While our action sampling is strictly based on Π, the imitation weights w_{align} effectively reshape the fitting target. Mathematically, our weighted objective implicitly defines a refined target distribution $\tilde{\Pi}$:

$$\tilde{\Pi}(\mathbf{a}|\mathbf{z}) \propto w_{\text{align}}(\mathbf{z}, \mathbf{a}) \Pi(\mathbf{a}|\mathbf{z}). \tag{13}$$

Minimizing the forward KL divergence to this target, $\mathcal{D}_{\mathrm{KL}}(\tilde{\Pi} \parallel \pi_\theta)$, requires maximizing the expected log-likelihood $\mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \pi_\theta(\mathbf{a}|\mathbf{z})]$. Since the implicit normalization constant of $\tilde{\Pi}$ is independent of the actor parameters θ, this optimization is mathematically equivalent to maximizing our tractable, unnormalized weighted likelihood objective:

$$\arg\min_\theta \mathcal{D}_{\mathrm{KL}}(\tilde{\Pi} \parallel \pi_\theta) \equiv \arg\max_\theta \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w_{\text{align}}(\mathbf{z},\mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z})\big]. \tag{14}$$

Derivation. See Appendix A.1 for the full derivation. □

Safety of In-Sample Learning. Because the refined target distribution $\tilde{\Pi}$ is constructed strictly within the support of the buffer $\mathcal{B}$, our method structurally satisfies the in-sample safety condition. By ensuring that the learned policy aligns with this safe target, we inherit robust theoretical guarantees against out-of-distribution errors. Adapting the theoretical framework from [15] to our refined target $\tilde{\Pi}$, we can establish strict bounds on both the performance degradation and value overestimation:

Theorem 4.1 (In-Sample Safety Bounds, adapted from [15])**.** *Let $\tilde{\Pi}$ be the refined empirical distribution constrained within the replay buffer $\mathcal{B}$, and π_θ be the learned actor policy. Assume the per-step reward is bounded by $R_{\max}$, the discount factor is $\gamma \in [0,1)$, and $Q(\mathbf{z},\mathbf{a})$ is L_Q-Lipschitz continuous in $\mathbf{a}$. If the extraction objective ensures $\mathcal{D}_{\mathrm{KL}}(\tilde{\Pi} \parallel \pi_\theta) \le \varepsilon$ for any state $\mathbf{z}$, then both the return gap and the Q-value expected error are strictly bounded:*

$$|J(\tilde{\Pi}) - J(\pi_\theta)| \le \frac{R_{\max}}{1-\gamma}\sqrt{2\epsilon}, \quad \textit{and} \quad |Q(\mathbf{z},\mathbf{a}_{\tilde{\Pi}}) - Q(\mathbf{z},\mathbf{a}_\pi)| \le L_Q \cdot D(\varepsilon), \tag{15}$$

where $D(\varepsilon)$ is a deterministic upper bound on the action divergence strictly dependent on ε.

Proof. The formal proof refers to [15], which is applied here to our refined target distribution $\tilde{\Pi}$. □

Breaking Mode-Covering via Support Sparsity. While safe, standard in-sample extraction methods (e.g., BOOM [15]) intrinsically suffer from mode-covering behavior. We formally demonstrate that our truncated weighting method mathematically breaks this limitation by strategically shrinking the support set of the target distribution, defined as $\mathrm{supp}(P(\cdot|\mathbf{z})) = \{\mathbf{a} \in \mathcal{A} \mid P(\mathbf{a}|\mathbf{z}) > 0\}$.

Proposition 4.2 (Support Sparsity Breaks Mode-Covering)**.** *Let $\Pi(\mathbf{a}|\mathbf{z})$ be the empirical planner distribution. Under standard exponential weighting, the refined target $\tilde{\Pi}_{exp}$ strictly preserves the planner's support:* $\mathrm{supp}(\tilde{\Pi}_{exp}) = \mathrm{supp}(\Pi)$. *Conversely, our truncated weighting w_{align} explicitly zeroes out suboptimal regions, inducing a strictly sparser support set for our target $\tilde{\Pi}_{ours}$:*

$$\mathrm{supp}(\tilde{\Pi}_{ours}) = \mathrm{supp}(\Pi) \setminus \mathcal{A}_{subopt}(\mathbf{z}), \tag{16}$$

where $\mathcal{A}_{subopt}(\mathbf{z}) = \{\mathbf{a} \in \mathrm{supp}(\Pi) \mid A(\mathbf{z},\mathbf{a}) \le -2\}$ denotes the eliminated suboptimal actions.

Proof. See Appendix A.2 for detailed proof. □

Monotonic Local Ascent via First-Order Alignment. Beyond breaking mode-covering, our framework structurally guarantees a steeper local ascent. While standard in-sample methods rely solely on zero-order scalars ($A^{(0)}$) and remain blind to value geometry, we formally prove that injecting our first-order alignment, $A^{(1)}(\mathbf{z},\mathbf{a}) = \langle \mathbf{a} - \boldsymbol{\mu}_\theta(\mathbf{z}), \nabla_{\mathbf{a}} Q_\theta(\mathbf{z},\mathbf{a})\rangle$, geometrically coerces the actor's update direction to align with the critic's steepest ascent. Assuming the value landscape is locally smooth in the vicinity of the policy mean, we formally demonstrate that:

Theorem 4.3 (Gradient Alignment Guarantees Superior Local Ascent)**.** *Let $\mu_\theta(\mathbf{z})$ be the mean of a Gaussian actor policy, and $\mathbf{d}_Q = \nabla_{\mathbf{a}} Q_\theta(\mathbf{z},\mathbf{a})|_{\mathbf{a}=\mu_\theta}$ be the local steepest ascent direction of the critic. Let $\Delta\mu_{zero}$ denote the expected policy update driven by a purely zero-order baseline advantage $A^{(0)}$, and $\Delta\mu_{align}$ denote the update driven by our geometrically modulated advantage $A = A^{(0)} + \lambda \cdot \mathbb{1}_{A^{(0)}(\mathbf{z},\mathbf{a})>0} \cdot A^{(1)}$. The projection of our aligned policy update onto the true value gradient is strictly greater than or equal to the unaligned baseline:*

$$\mathbf{d}_Q^\top \Delta\mu_{align} \ge \mathbf{d}_Q^\top \Delta\mu_{zero}. \tag{17}$$

Proof. See Appendix A.3 for detailed proof. □

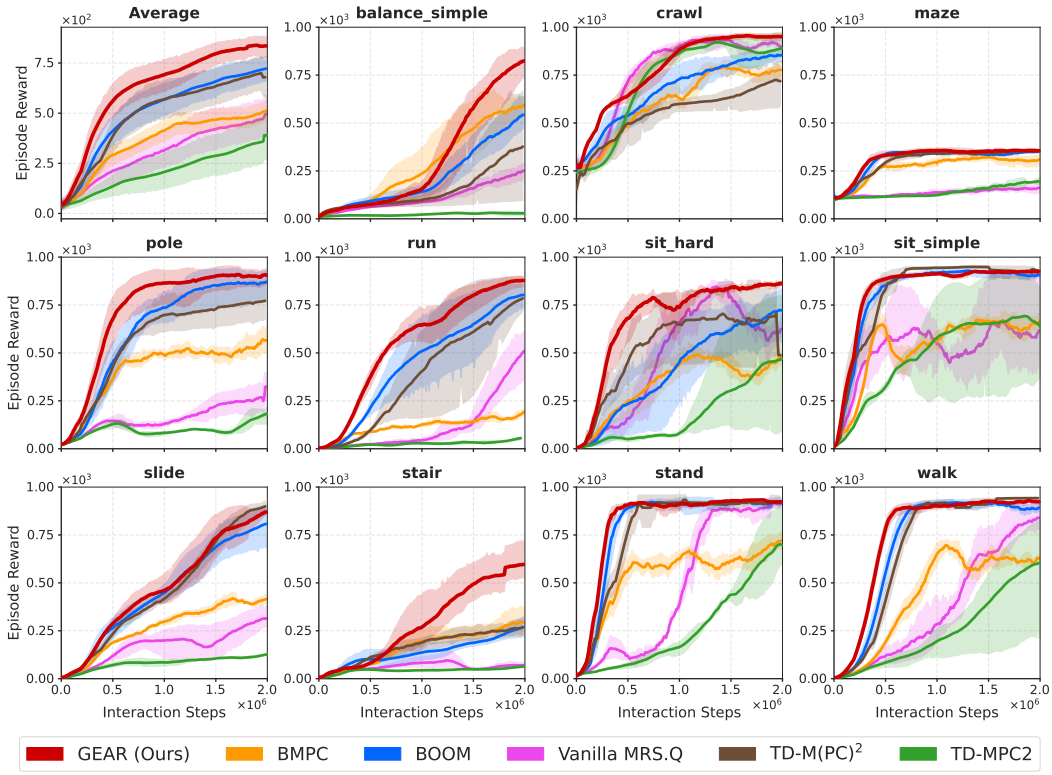

Figure 2: **Overall performance of HumanoidBench.** Top-left: aggregate average return. Curves and shaded regions represent the mean and 95% CIs over 5 seeds (3 seeds for TD-MPC2).

5 Experiments

5.1 Experimental Setup

Baselines. We compare GEAR against representative online RL algorithms, with a specific focus on planning-driven MBRL and recent policy extraction algorithms: (1) **TD-MPC2** [10]: a foundational planning-driven MBRL algorithm that relies on an MPPI planner without explicit actor-planner alignment; (2) **BMPC** [13]: an actor-aligned variant of TD-MPC2 that distills the online MPPI planner via behavioral cloning, which heavily relies on frequent online re-planning; (3) **BOOM** [15]: an extraction method that incorporates a likelihood-free forward KL penalty for in-sample imitation, representing inherently mode-covering extraction; (4) **TD-M(PC)^2** [16]: an approach adopting a reverse KL regularizer, which forces reliance on an approximate parametric proxy since the sample-based, non-parametric planner cannot provide an exact analytical density; (5) **Vanilla MRS.Q** [19]: for a fair comparison, we reimplement its core mechanism within the TD-MPC2 architecture, suppressing overestimation via aggressive minimization across a massive Q-ensemble.

Benchmark. We evaluate our method on a challenging suite of 11 high-dimensional locomotion tasks from `HumanoidBench` [18]. These tasks feature a simulated Unitree H1hand humanoid robot with a 61-dimensional continuous action space. Unlike standard continuous control environments, controlling this high-dimensional humanoid requires the algorithm to carefully manage whole-body balance and complex physical dynamics. Empirical evaluations on this benchmark indicate that standard end-to-end RL algorithms (e.g., SAC [31], DreamerV3 [9], and TD-MPC2 [10]) struggle with controlling the complex humanoid robot dynamics and solving the most challenging tasks.

Implementation Details. See Appendix B for full implementation details. Notably, GEAR inherits BOOM's default hyperparameters and requires no task-specific tuning across any environment.

Table 2: **Final performance of different algorithms.** Scores are averaged over the final 10% training steps over 5 seeds. The best score is highlighted in **colorbox**, and the second best is underlined.

Task	TD-MPC2	BMPC	BOOM	Vanilla MRS.Q	TD-M(PC)^2	GEAR (Ours)
balance_simple	28.45 ± 8.27	594.70 ± 129.84	542.35 ± 121.85	251.38 ± 79.77	381.65 ± 264.74	**824.45** ± 82.34
crawl	890.05 ± 60.39	777.05 ± 87.10	853.01 ± 76.64	890.63 ± 158.56	709.08 ± 147.16	**951.00** ± 31.29
maze	197.01 ± 34.89	309.08 ± 63.29	350.54 ± 6.98	162.14 ± 66.94	350.09 ± 6.53	**355.54** ± 10.76
pole	183.21 ± 41.49	564.71 ± 160.00	871.69 ± 105.16	281.23 ± 120.12	773.35 ± 143.52	**905.68** ± 67.33
run	55.38 ± 11.46	192.43 ± 53.47	803.42 ± 89.84	509.15 ± 155.38	830.52 ± 53.43	**878.04** ± 38.31
sit_hard	468.54 ± 296.99	477.04 ± 155.72	722.33 ± 187.68	626.20 ± 247.71	705.07 ± 255.65	**863.70** ± 55.37
sit_simple	639.18 ± 219.99	662.49 ± 165.69	905.84 ± 58.29	616.86 ± 335.66	**930.36** ± 14.62	926.68 ± 32.22
slide	126.93 ± 13.00	415.15 ± 101.01	807.12 ± 135.16	314.38 ± 63.24	**906.72** ± 47.50	867.49 ± 36.29
stair	63.72 ± 5.98	295.60 ± 116.40	269.26 ± 62.16	69.50 ± 25.87	277.22 ± 77.26	**593.26** ± 93.74
stand	696.81 ± 161.52	721.35 ± 150.11	919.88 ± 76.12	912.65 ± 67.68	921.45 ± 60.62	**921.75** ± 62.32
walk	604.75 ± 282.99	628.34 ± 158.96	893.02 ± 86.72	838.12 ± 99.57	**942.96** ± 3.13	923.64 ± 32.24
Avg Improvement	Baseline	+281.3%	+421.9%	+172.3%	+377.7%	**+581.3%**

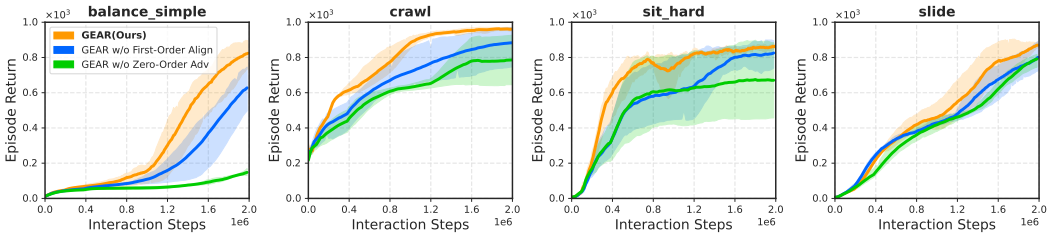

Figure 3: **Component ablation of GEAR on 4 HumanoidBench H1hand tasks.** We compare full GEAR with variants that remove first-order gradient alignment or zero-order advantage modulation while keeping all other training settings unchanged.

5.2 Experimental Results

Results on HumanoidBench. All training curves and detailed numerical results are presented in Figure 2 and Table 2. GEAR achieves the highest final returns in **8** out of 11 high-dimensional `HumanoidBench` locomotion tasks and yielding a striking **+581.3%** average improvement over the base TD-MPC2 algorithm. While these tasks pose significant challenges due to their massive state-action spaces and complex whole-body dynamics, GEAR demonstrates remarkable stability and effectiveness. This substantial margin over the unaligned baseline underscores the necessity of effectively bridging the planner-actor mismatch.

Compared to two representative baselines, BOOM and TD-M(PC)^2, GEAR achieves faster convergence and superior asymptotic performance. These methods exemplify the typical extremes of policy extraction: BOOM's *Forward KL* offers likelihood-free stability but suffers from mode-covering conflicts and weight explosions, whereas the *Reverse KL* in TD-M(PC)^2 provides mode-seeking guidance but relies on brittle proxy distributions that destabilize training. GEAR resolves this dilemma through a structurally unified, value-modulated extraction. By anchoring a likelihood-free objective with absolute, first-order geometric gating, GEAR eliminates Forward KL's mode conflicts and bypasses brittle proxies, driving focused and stable mode-seeking improvement. While baselines like BMPC and Vanilla MRS.Q show occasional successes, structural flaws cripple their overall performance. BMPC degrades into value-blind imitation, and Vanilla MRS.Q's massive Q-ensemble triggers crippling pessimism. In summary, by structurally embedding first-order value modulation, GEAR avoids these extremes and maintains clear, consistent advantages across all tasks.

Ablation Study. We evaluate the contributions of zero-order advantage modulation $A^{(0)}$ and first-order gradient alignment $A^{(1)}$ on four HumanoidBench H1hand tasks. We compare full GEAR with two variants: *GEAR w/o First-Order Align*, which relies only on the zero-order value comparison, and *GEAR w/o Zero-Order Adv*, which removes zero-order advantage modulation. All other training settings are kept unchanged.

As shown in Figure 3, full GEAR achieves the highest final return across all four tasks and generally learns faster than the two ablated variants. Removing first-order alignment noticeably slows learning on `balance_simple`, `crawl`, and `sit_hard`, although the degradation is smaller on `slide`. This

indicates that the local gradient information provides useful directional refinement beyond the zero-order value comparison.

Removing zero-order advantage modulation causes a more pronounced degradation. It makes little progress on `balance_simple` and converges to lower performance on `crawl` and `sit_hard`. On `slide`, it eventually approaches the other variants but remains less sample-efficient than full GEAR during the later stage of training. These results suggest that the zero-order term provides the primary value-based screening of candidate actions, while the first-order term further refines their extraction. Combining both signals therefore yields the most consistent learning performance across tasks.

Sensitivity to the Alignment Weight The coefficient λ_{align} controls the relative influence of the first-order directional correction with respect to the zero-order value comparison. To examine this trade-off, we vary $\lambda_{\text{align}} \in \{0.01, 0.1, 1.0\}$ on `run` and `stair`.

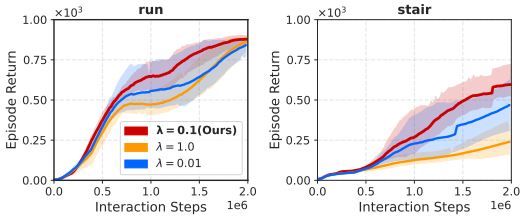

Figure 4: **Sensitivity to the first-order alignment weight.** We vary $\lambda_{\text{align}} \in \{0.01, 0.1, 1.0\}$ on `h1hand-run` and `h1hand-stair`, while fixing the zero-order scale to 1.0 and keeping all other settings unchanged.

As shown in Figure 4, the intermediate value $\lambda_{\text{align}} = 0.1$ provides the strongest overall behavior. On `run`, all three settings eventually reach relatively high returns, but $\lambda_{\text{align}} = 0.1$ improves more rapidly during the middle stage of training and maintains a slight advantage at convergence. The difference is more pronounced on `stair`, where $\lambda_{\text{align}} = 0.1$ learns faster and reaches a substantially higher final return than the other settings.

With $\lambda_{\text{align}} = 0.01$, the first-order signal is weak, so policy extraction relies mainly on zero-order value comparison and benefits less from directional refinement. In contrast, $\lambda_{\text{align}} = 1.0$ overemphasizes local critic gradients, increasing sensitivity to gradient variation and slowing learning, particularly on `stair`. The intermediate value 0.1 retains the zero-order term as the primary screening criterion while preserving a meaningful first-order correction.

Although these two tasks do not establish a universally optimal coefficient, the results suggest that $\lambda_{\text{align}} = 0.1$ balances learning efficiency and final performance. We use it as the default, while task-specific tuning may yield further improvements.

6 Conclusion

In this paper, we presented GEAR, a gradient-aligned policy extraction algorithm that effectively resolves the planner-actor mismatch in hybrid MBRL. By structurally integrating first-order Q-gradients with an absolute geometric gating mechanism, GEAR successfully eliminates the mode-covering objective conflicts inherent in standard forward KL approaches. Extensive empirical evaluations on the highly challenging `HumanoidBench` suite demonstrate that GEAR achieves SOTA performance, thoroughly outperforming existing baselines in both asymptotic returns and stability.

Limitation and Future Work. GEAR's reliance on first-order Q-gradients assumes a locally linear value landscape, limiting performance in highly non-convex environments. Future work could extend this alignment via a second-order Taylor expansion of the HJB equation. By incorporating the Hessian for curvature-aware modulation, the algorithm could better distinguish local maxima from saddle points, ensuring robust policy extraction in complex continuous control.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was partially supported by the National Key Research and Development Program of China (Grant No. 2025YFC3309100), the National Natural Science Foundation of China (Grant No. 62276024), the Beijing Natural Science Foundation (Grant No. 4262066), and the Fundamental Research Funds for the Central Universities (Grant No. 2025CX01010). This work was also supported by the Intelligent Computing Platform of Beijing Institute of Technology (BIT).

References

[1] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[2] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25476–25488. Curran Associates, Inc., 2021.

[3] Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. EfficientZero v2: Mastering discrete and continuous control with limited data. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 51041–51062. PMLR, 21–27 Jul 2024.

[4] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

[5] Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8387–8406. PMLR, 17–23 Jul 2022.

[6] Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Adrià López Escoriza, Nicklas Hansen, Stone Tao, Tongzhou Mu, and Hao Su. Multi-stage manipulation with demonstration-augmented reward, policy, and world model learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 15542–15563. PMLR, 13–19 Jul 2025.

[8] Yao-Hui Li, Zeyu Wang, Xin Li, Wei Pang, Yingfang Yuan, Zhengkun Chen, Boya Zhang, Riashat Islam, Alex Lamb, and Yonggang Zhang. Slope: Optimistic potential landscape shaping for model-based reinforcement learning. *arXiv preprint arXiv:2602.03201*, 2026.

[9] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models.

[10] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. PWM: Policy learning with multi-task world models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Nicklas Hansen, Hao Su, and Xiaolong Wang. Learning massively multitask world models for continuous control. In *The Fourteenth International Conference on Learning Representations*, 2026.

[13] Yuhang Wang, Hanwei Guo, Sizhe Wang, Long Qian, and Xuguang Lan. Bootstrapped model predictive control. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

[15] Guojian Zhan, Likun Wang, Xiangteng Zhang, Jiaxin Gao, Masayoshi Tomizuka, and Shengbo Eben Li. Bootstrap off-policy with world model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[16] Haotian Lin, Pengcheng Wang, Jeff Schneider, and Guanya Shi. TD-M $(PC)^2$: Improving temporal difference mpc through policy constraint. *arXiv preprint arXiv:2502.03550*, 2025.

[17] Scott Fujimoto and Shixiang (Shane) Gu. A minimalist approach to offline reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20132–20145. Curran Associates, Inc., 2021.

[18] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[19] Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. *arXiv preprint arXiv:2601.21306*, 2026.

[20] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[21] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[22] Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. In *International Conference on Learning Representations*, 2020.

[23] Harshit Sikchi, Wenxuan Zhou, and David Held. Learning off-policy with online planning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1622–1633. PMLR, 08–11 Nov 2022.

[24] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[25] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

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

[27] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020.

[28] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Haoran Xu, Liyuan Mao, Hui Jin, Weinan Zhang, Xianyuan Zhan, and Amy Zhang. Uni-RL: Unifying online and offline RL via implicit value regularization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[30] Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-jacobi deep q-learning for deterministic continuous-time systems with lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206):1–34, 2021.

[31] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

A Proof

A.1 Derivation of the Extraction Objective Equivalence

In Section 4.1, we define the implicit refined target distribution $\tilde{\Pi}$ as:

$$\tilde{\Pi}(\mathbf{a}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})} w_{\text{align}}(\mathbf{z}, \mathbf{a}) \Pi(\mathbf{a}|\mathbf{z}), \tag{18}$$

where $Z(\mathbf{z}) = \int w_{\text{align}}(\mathbf{z}, \mathbf{a}) \Pi(\mathbf{a}|\mathbf{z}) d\mathbf{a} \equiv \mathbb{E}_{\mathbf{a}\sim\Pi}[w_{\text{align}}(\mathbf{z}, \mathbf{a})]$ is the state-dependent partition function ensuring that $\tilde{\Pi}$ is a valid probability distribution.

Our goal is to optimize the actor parameters θ by minimizing the forward KL divergence from this target distribution to the parametric policy π_θ:

$$\begin{aligned} \arg\min_\theta \mathcal{D}_{\text{KL}}(\tilde{\Pi} \parallel \pi_\theta) &= \arg\min_\theta \int \tilde{\Pi}(\mathbf{a}|\mathbf{z}) \log \frac{\tilde{\Pi}(\mathbf{a}|\mathbf{z})}{\pi_\theta(\mathbf{a}|\mathbf{z})} d\mathbf{a} \\ &= \arg\min_\theta \Big(\mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \tilde{\Pi}(\mathbf{a}|\mathbf{z})] - \mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \pi_\theta(\mathbf{a}|\mathbf{z})] \Big). \end{aligned} \tag{19}$$

Since the first term (the entropy of $\tilde{\Pi}$) is strictly independent of the actor parameters θ, minimizing the KL divergence is mathematically equivalent to maximizing the second term (the cross-entropy):

$$\arg\min_\theta \mathcal{D}_{\text{KL}}(\tilde{\Pi} \parallel \pi_\theta) = \arg\max_\theta \mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \pi_\theta(\mathbf{a}|\mathbf{z})]. \tag{20}$$

By expanding the expectation over $\tilde{\Pi}$ using its definition, we can rewrite the objective in terms of the original planner distribution Π:

$$\begin{aligned} \arg\max_\theta \mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \pi_\theta(\mathbf{a}|\mathbf{z})] &= \arg\max_\theta \int \left(\frac{1}{Z(\mathbf{z})} w_{\text{align}}(\mathbf{z}, \mathbf{a}) \Pi(\mathbf{a}|\mathbf{z}) \right) \log \pi_\theta(\mathbf{a}|\mathbf{z}) d\mathbf{a} \\ &= \arg\max_\theta \frac{1}{Z(\mathbf{z})} \int \Pi(\mathbf{a}|\mathbf{z}) \Big(w_{\text{align}}(\mathbf{z}, \mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z}) \Big) d\mathbf{a} \\ &= \arg\max_\theta \frac{1}{Z(\mathbf{z})} \mathbb{E}_{\mathbf{a}\sim\Pi} \big[w_{\text{align}}(\mathbf{z}, \mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z}) \big]. \end{aligned} \tag{21}$$

Finally, because the partition function $Z(\mathbf{z})$ depends only on the state $\mathbf{z}$ and the static buffer distribution Π, it acts as a strictly positive constant with respect to the optimization variables θ. Scaling an objective function by a positive constant does not alter the location of its global or local optima. Therefore, we arrive at the exact unnormalized weighted likelihood objective used in our practical implementation:

$$\arg\min_\theta \mathcal{D}_{\text{KL}}(\tilde{\Pi} \parallel \pi_\theta) \equiv \arg\max_\theta \mathbb{E}_{\mathbf{a}\sim\Pi} \big[w_{\text{align}}(\mathbf{z}, \mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z}) \big]. \tag{22}$$

This concludes the derivation, proving that optimizing our computationally tractable in-sample objective strictly aligns the policy with the refined target distribution.

A.2 Proof of Theorem 4.2: Breaking Mode-Covering

In this section, we provide the formal proof for Theorem 4.2, demonstrating how our truncated weighting mechanism structurally prevents mode-covering behavior by inducing support sparsity.

Proof. Let the support set of a probability distribution P over the continuous action space $\mathcal{A}$ be defined as $\text{supp}(P(\cdot|\mathbf{z})) = \{\mathbf{a} \in \mathcal{A} \mid P(\mathbf{a}|\mathbf{z}) > 0\}$. The objective of forward KL-based policy extraction is to minimize $\mathcal{D}_{\text{KL}}(\tilde{\Pi} \parallel \pi_\theta)$, which contains the cross-entropy term $-\mathbb{E}_{\mathbf{a}\sim\tilde{\Pi}}[\log \pi_\theta(\mathbf{a}|\mathbf{z})]$. To avoid an infinite penalty (i.e., $\log \pi_\theta(\mathbf{a}|\mathbf{z}) \to -\infty$), the learned policy π_θ must satisfy $\text{supp}(\pi_\theta) \supseteq \text{supp}(\tilde{\Pi})$.

Step 1: Standard Exponential Weighting (Mode-Covering). In standard methods like BOOM, the refined target distribution is proportional to an exponentially weighted planner distribution:

$$\tilde{\Pi}_{\text{exp}}(\mathbf{a}|\mathbf{z}) \propto \exp\left(\frac{Q(\mathbf{z}, \mathbf{a})}{\tau} \right) \Pi(\mathbf{a}|\mathbf{z}). \tag{23}$$

Since the exponential function is strictly positive ($\exp(x) > 0$ for all $x \in \mathbb{R}$), the target distribution satisfies $\tilde{\Pi}_{\text{exp}}(\mathbf{a}|\mathbf{z}) > 0 \iff \Pi(\mathbf{a}|\mathbf{z}) > 0$. Therefore, the support set is strictly preserved:

$$\text{supp}(\tilde{\Pi}_{\text{exp}}) = \text{supp}(\Pi). \tag{24}$$

Consequently, the actor π_θ is forced to assign non-zero probability mass to *every* action explored by the planner, including highly suboptimal ones, thereby strictly enforcing mode-covering.

Step 2: Our Truncated Weighting (Support Sparsity). In our formulation, the refined target distribution is defined as:

$$\tilde{\Pi}_{\text{ours}}(\mathbf{a}|\mathbf{z}) \propto \left[\text{clip}\left(1 + \frac{A(\mathbf{z},\mathbf{a})}{2}, 0, w_{\max}\right)\right]^2 \Pi(\mathbf{a}|\mathbf{z}). \tag{25}$$

Due to the hard lower-bound truncation at zero, for any action $\mathbf{a}$ where the modulated advantage satisfies $A(\mathbf{z},\mathbf{a}) \leq -2$, the imitation weight evaluates to exactly 0. Let this subset of suboptimal actions be denoted as $\mathcal{A}_{\text{subopt}}(\mathbf{z}) = \{\mathbf{a} \in \text{supp}(\Pi) \mid A(\mathbf{z},\mathbf{a}) \leq -2\}$.

For any action $\mathbf{a} \in \mathcal{A}_{\text{subopt}}(\mathbf{z})$, the probability mass in our target distribution becomes:

$$\tilde{\Pi}_{\text{ours}}(\mathbf{a}|\mathbf{z}) \propto 0 \cdot \Pi(\mathbf{a}|\mathbf{z}) = 0. \tag{26}$$

This mathematically removes the suboptimal region from the support set, yielding:

$$\text{supp}(\tilde{\Pi}_{\text{ours}}) = \text{supp}(\Pi) \setminus \mathcal{A}_{\text{subopt}}(\mathbf{z}). \tag{27}$$

Conclusion. Because $\mathcal{A}_{\text{subopt}}(\mathbf{z})$ is excluded from $\text{supp}(\tilde{\Pi}_{\text{ours}})$, the learned policy π_θ is no longer penalized for assigning zero probability to these inferior actions. This frees the actor to aggressively shrink its variance and concentrate its probability mass solely on the geometrically aligned, high-value regions $\text{supp}(\tilde{\Pi}_{\text{ours}})$, thus successfully enabling in-sample mode-seeking behavior. □

A.3 Proof of Theorem 4.3: Monotonic Local Ascent

In this section, we formally prove Theorem 4.3, demonstrating that the inclusion of the first-order alignment score $A^{(1)}(\mathbf{z},\mathbf{a})$ mathematically guarantees a superior or equal local ascent in the value landscape compared to a purely zero-order baseline.

Proof. Assume the actor is parameterized as a Gaussian policy $\pi_\theta(\mathbf{a}|\mathbf{z}) = \mathcal{N}(\mu_\theta(\mathbf{z}), \Sigma)$. For simplicity of analysis, assume an isotropic covariance $\Sigma = \sigma^2 I$. The gradient of the weighted log-likelihood objective with respect to the mean μ_θ is given by:

$$\nabla_{\mu_\theta} \mathbb{E}_{\mathbf{a}\sim\Pi}[w(\mathbf{z},\mathbf{a}) \log \pi_\theta(\mathbf{a}|\mathbf{z})] = \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{a}\sim\Pi}[w(\mathbf{z},\mathbf{a})(\mathbf{a} - \mu_\theta(\mathbf{z}))]. \tag{28}$$

By defining the behavioral cloning displacement vector as $\mathbf{d}_{\text{BC}}(\mathbf{z},\mathbf{a}) \triangleq \mathbf{a} - \mu_\theta(\mathbf{z})$, the expected policy update direction $\Delta\mu_w$ driven by a weighting function w is directly proportional to this expected displacement:

$$\Delta\mu_w \propto \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w(\mathbf{z},\mathbf{a})\mathbf{d}_{\text{BC}}(\mathbf{z},\mathbf{a})\big]. \tag{29}$$

Let $\mathbf{d}_Q = \nabla_{\mathbf{a}} Q_\theta(\mathbf{z},\mathbf{a})|_{\mathbf{a}=\mu_\theta}$ be the local steepest ascent direction of the critic. Assuming the Q-function is locally linear within the support of the aligned planner samples (a standard first-order approximation), we have $\nabla_{\mathbf{a}} Q_\theta(\mathbf{z},\mathbf{a}) \approx \mathbf{d}_Q$. Projecting the expected update direction $\Delta\mu_w$ onto the value gradient $\mathbf{d}_Q$ yields the expected local ascent. Based on our definition of the alignment score $A^{(1)}(\mathbf{z},\mathbf{a}) = \langle \mathbf{d}_{\text{BC}}, \mathbf{d}_Q \rangle$, we have:

$$\mathbf{d}_Q^\top \Delta\mu_w \propto \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w(\mathbf{z},\mathbf{a})\mathbf{d}_Q^\top \mathbf{d}_{\text{BC}}(\mathbf{z},\mathbf{a})\big] = \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w(\mathbf{z},\mathbf{a})A^{(1)}(\mathbf{z},\mathbf{a})\big]. \tag{30}$$

Next, we define the final weighting functions for the baseline and our method. According to Eq. 11, the imitation weight is computed via a truncated polynomial function $f(x) = [\text{clip}(1 + x/2, 0, w_{\max})]^2$. Crucially, $f(x)$ is a *monotonically non-decreasing* function for all $x \in \mathbb{R}$.

Let $w_{\text{zero}} = f(A^{(0)})$ be the weight driven by the baseline zero-order advantage, and $w_{\text{align}} = f(A^{(0)} + \lambda \cdot \mathbb{1}_{A^{(0)}(\mathbf{z},\mathbf{a})>0} \cdot A^{(1)})$ be our aligned weight. The difference in the projected local ascent

between our method and the baseline is proportional to:

$$\begin{aligned}\mathbf{d}_Q^\top(\Delta\mu_{\text{align}} - \Delta\mu_{\text{zero}}) &= \mathbf{d}_Q^\top\Delta\mu_{\text{align}} - \mathbf{d}_Q^\top\Delta\mu_{\text{zero}} \\ &\propto \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w_{\text{align}}(\mathbf{z},\mathbf{a})A^{(1)}(\mathbf{z},\mathbf{a})\big] - \mathbb{E}_{\mathbf{a}\sim\Pi}\big[w_{\text{zero}}(\mathbf{z},\mathbf{a})A^{(1)}(\mathbf{z},\mathbf{a})\big] \\ &= \mathbb{E}_{\mathbf{a}\sim\Pi}\Big[\big(w_{\text{align}}(\mathbf{z},\mathbf{a}) - w_{\text{zero}}(\mathbf{z},\mathbf{a})\big)A^{(1)}(\mathbf{z},\mathbf{a})\Big].\end{aligned} \tag{31}$$

We now analyze the term inside the expectation, denoted as $\Psi = (w_{\text{align}} - w_{\text{zero}})A^{(1)}(\mathbf{z},\mathbf{a})$, evaluating it across all possible conditions:

- **Case 1 ($A^{(0)} \leq 0$):** The indicator function evaluates to 0. Thus, the inputs to the weight function are identical ($w_{\text{align}} = w_{\text{zero}}$), yielding $\Psi = 0$.
- **Case 2 ($A^{(0)} > 0$ and $A^{(1)} > 0$):** The positive alignment score increases the input to the weight function. Due to the monotonically non-decreasing nature of $f(x)$, we have $w_{\text{align}} \geq w_{\text{zero}}$, which means $(w_{\text{align}} - w_{\text{zero}}) \geq 0$. Multiplying by $A^{(1)} > 0$ strictly yields $\Psi \geq 0$.
- **Case 3 ($A^{(0)} > 0$ and $A^{(1)} < 0$):** The negative alignment score decreases the input to the weight function. Due to monotonicity, $w_{\text{align}} \leq w_{\text{zero}}$, resulting in $(w_{\text{align}} - w_{\text{zero}}) \leq 0$. Since $A^{(1)} < 0$, the product of two non-positive terms yields $\Psi \geq 0$.

Since $\Psi \geq 0$ holds universally for all possible sampled actions, its mathematical expectation must be non-negative:

$$\mathbb{E}_{\mathbf{a}\sim\Pi}\Big[\big(w_{\text{align}} - w_{\text{zero}}\big)A^{(1)}(\mathbf{z},\mathbf{a})\Big] \geq 0. \tag{32}$$

This concludes the proof, formally establishing that $\mathbf{d}_Q^\top\Delta\mu_{\text{align}} \geq \mathbf{d}_Q^\top\Delta\mu_{\text{zero}}$. By modulating the empirical actions based on directional consistency, our method mathematically acts as a positive drift, ensuring the policy update takes a steeper path in the value landscape compared to blind zero-order imitation. □

B Implementation Details

We compare GEAR against representative online RL algorithms, with a specific focus on planning-driven MBRL and recent policy extraction methods. To ensure a fair comparison, we utilize the official implementations and default hyperparameter settings for most baselines unless otherwise specified.

B.1 Hyperparameters

Table 3: **Hyperparameter settings.** New or changed hyperparameters are highlighted

	Hyperparameter	**Value**
GEAR(Ours)	Scaling temperature	1
	Alignment strength (λ)	0.1
Training	Learning rate	3×10^{-4}
	Target network update rate	0.5
	Encoder learning rate	1×10^{-4}
	Discount factor (γ)	0.99
	Action repeat	1
	Gradient Clipping Norm	20
	Replay batch size	256
	Optimizer	Adam
	Buffer size	1,000,000
	Loss norm	Moving (5%, 95%)
	Interaction Steps	2,000,000
	Sampling	Uniform
World Model	Reward loss coefficient (c_r)	0.1
	Dynamics loss coefficient (c_f)	20
	Value loss coefficient (c_q)	0.1
	Value functions ensemble	5
	Number of value bins	101
	Warmup steps	5000
Planner	MPPI Iterations	6 (8 if $\|\mathcal{A}\| > 20$)
	Minimum planner std	0.05
	Population size	512
	Maximum planner std	2
	Number of elites	64
	Horizon	3
	Policy prior samples	24
Actor	Minimum policy log std	Defaule: -10
		BMPC: -3
	Entropy coefficient	1×10^{-4}
	Maximum policy log std	Default: 2
		BMPC: 1
Architecture (5M)	Q-ensemble size	Default: 5
		Vanilla MRS.Q: 10
	Encoder layers	2
	Latent space dimension	512
	Encoder dimension	256
	Task embedding dimension	96
	MLP hidden layer dimension	512
	Q function drop out rate	0.01
	MLP activation	Mish
	MLP Normalization	LayerNorm

B.2 Algorithm

Algorithm 1 GEAR: Gradient-Aligned Policy Extraction

Require: Replay buffer $\mathcal{D}$, hyperparameters $\alpha, \lambda, w_{\max}$
Initialize policy π_θ, value Q_θ, world model $\mathcal{M}$
▷ Gradient-aligned policy extraction
procedure GEAR-UPDATE($\pi_\theta, Q_\theta, \mathcal{D}$)
  Sample planner transitions $(s, a) \sim \mathcal{D}$
  Compute zero-order global advantage $A^{(0)}(z, a)$ by Eq. 8
  Compute first-order local alignment $A^{(1)}(z, a)$ by Eq. 9
  Modulate advantage $A(z, a) \leftarrow A^{(0)}(z, a) + \lambda \cdot \mathbb{1}_{A^{(0)}>0} \cdot A^{(1)}(z, a)$
  Compute gradient-aligned weights $w_{\text{align}}(z, a)$ by Eq. 11
  Update π_θ by maximizing unified objective $\mathcal{J}_{\text{align}}(\theta)$ in Eq. 12
end procedure
▷ Hybrid MBRL training loop
for iteration $t = 1, 2, \cdots, N$ **do**
  Plan $a \sim \mathcal{P}(\pi_\theta, \mathcal{M})$, interact, and append to $\mathcal{D}$
  Update world model $\mathcal{M}$ and value Q_θ using $\mathcal{D}$
  GEAR-UPDATE($\pi_\theta, Q_\theta, \mathcal{D}$)
end for

B.3 Baselines

TD-MPC2 [10]. A foundational planning-driven MBRL algorithm that relies on an MPPI planner without explicit actor-planner alignment. We obtain its results of 3 seeds from the official codebase[2].

BMPC [13]. An actor-aligned variant of TD-MPC2 that distills the online MPPI planner via behavioral cloning, which heavily relies on frequent online re-planning. We reproduce its performance using the author-provided official implementation[3].

BOOM [15]. An extraction method that incorporates a likelihood-free forward KL penalty for in-sample imitation, representing inherently mode-covering extraction. We evaluate it using the default configurations from its official repository[4].

TD-M(PC)^2 [16]. An approach adopting a reverse KL regularizer, which forces reliance on an approximate parametric proxy. Results are obtained by running the official source code[5] under standard settings.

Vanilla MRS.Q [19]. For fair comparison, we reimplement its core mechanism within the TD-MPC2 architecture. Specifically, we suppress OOD overestimation via aggressive minimization across a massive Q-ensemble, matching the original paper's intent.

[2] https://github.com/carlosferrazza/humanoid-bench
[3] https://github.com/wertyuilife2/bmpc
[4] https://github.com/molumitu/BOOM_MBRL
[5] https://github.com/DarthUtopian/tdmpc_square_public

B.4 Computational Resources and Efficiency

All experiments were conducted on a server equipped with two Intel Xeon Gold 6240 CPUs (72 threads) and ten NVIDIA GeForce RTX 2080 Ti GPUs. Each individual training run used a single GPU; multiple GPUs were used only to run independent tasks and random seeds in parallel.

To quantify computational efficiency, we measure the number of learnable parameters, end-to-end wall-clock training time, and throughput under a common budget of 2M environment interaction steps. We additionally report the average performance improvement over TD-MPC2 to contextualize the performance–computation trade-off.

As shown in Table 4, GEAR contains 5.2M learnable parameters, identical to TD-MPC2, BOOM, and TD-M(PC)^2, since it introduces no additional policy, critic ensemble, or world-model component. However, computing the action-space Q-gradient during actor training incurs additional runtime. GEAR requires 50.0 hours at 11.0 FPS, compared with 38.6 hours at 14.4 FPS for TD-MPC2 and 46.7 hours at 11.9 FPS for BOOM. Thus, GEAR introduces a 29.5% wall-clock overhead relative to TD-MPC2 and a 7.1% overhead relative to BOOM, while remaining 15.0% faster than the online-replanning-based BMPC.

Despite this training-time overhead, GEAR achieves the largest average performance improvement in the comparison without increasing model size. These measurements characterize GEAR as providing a favorable performance–computation trade-off, rather than being universally the least expensive method. The additional computation is incurred only during actor training and introduces no additional test-time module.

Table 4: **Computational efficiency under a common training budget of 2M environment interaction steps.** Throughput denotes environment interaction steps per second, and Avg. Improvement is measured relative to TD-MPC2.

Method	**Params.** $\downarrow$	**Wall-clock Time (h)** $\downarrow$	**Throughput (FPS)** $\uparrow$	**Avg. Improvement vs. TD-MPC2 (%)** $\uparrow$
TD-MPC2	5.2M	**38.6**	**14.4**	–
BMPC	**3.4**M	58.8	9.4	+281.3
BOOM	5.2M	46.7	11.9	+421.9
TD-M(PC)^2	5.2M	44.2	12.6	+377.7
Vanilla MRS.Q	8.3M	45.6	12.2	+172.3
GEAR (Ours)	5.2M	50.0	11.0	**+581.3**

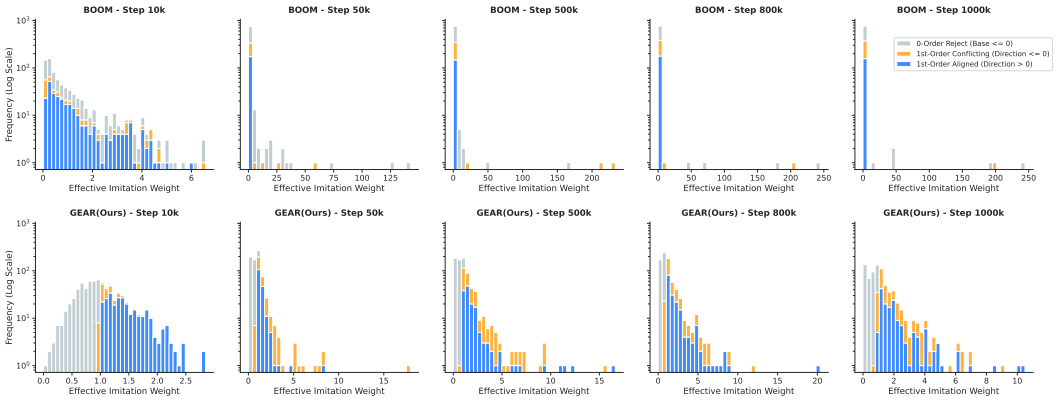

Figure 5: **Visualization of imitation weight distributions in `pole` task.** The top row illustrates BOOM, which computes relative weights via a batch-level Softmax over Q-values. This relative mechanism inevitably forces a highly skewed weight concentration (note the massive x-axis scale). The bottom row shows GEAR, which computes absolute weights using zero-order gating and first-order alignment independent of the batch context. Colors denote our geometric categorization. Notably, GEAR successfully suppresses suboptimal targets (Grey) while maintaining a stable, bounded distribution of aligned samples (Blue).

C Additional Experimental Results

C.1 Evolution of Imitation Weights

To empirically validate how our HJB-derived geometric alignment effectively isolates mode-seeking targets, we track the distribution of imitation weights assigned to buffer samples throughout the training process. We categorize the samples into three distinct types based on our proposed metric:

- **0-Order Reject (Grey)**: Suboptimal samples that are worse than the current actor's execution ($A^{(0)} \leq 0$).
- **1st-Order Conflicting (Orange)**: Promising samples ($A^{(0)} > 0$) that conflict with the local Q-gradient direction.
- **1st-Order Aligned (Blue)**: High-quality samples that are both globally superior and geometrically aligned with the Q-gradient.

Figure 5 visualizes the weight distribution of BOOM (using relative batch-level Softmax) versus GEAR (using absolute truncated polynomial) over 1M training steps. This comparison clearly highlights two critical findings:

Severe Instability and Extreme Weight Concentration in BOOM. The top row exposes the extreme numerical instability induced by BOOM's relative weighting mechanism. As training progresses from 10k to 1M steps, BOOM's weight distribution collapses into extreme sparsity, leading to a highly skewed weight allocation. The vast majority of samples are heavily diminished to near zero, while a minuscule fraction of samples absorbs the majority of the weight mass, resulting in massive weight explosions (note the BOOM x-axis scaling up to 250). In contrast, GEAR (bottom row) maintains a healthy, dense, and strictly bounded weight distribution throughout the entire training phase. By utilizing absolute polynomial truncation, GEAR ensures that the imitation weights are consistently constrained within a stable range ($[0, 20]$), completely avoiding the gradient explosions that plague relative Softmax assignments.

Absolute Gating Eradicates Spurious Imitation. Beyond numerical stability, BOOM's high-weight composition exposes a fatal flaw in sample efficiency. Because BOOM evaluates samples relative to the mini-batch, it frequently assigns absurdly high weights to the suboptimal Grey samples, simply because they happen to be the "least bad" options in a poor batch. This forces the actor to blindly imitate globally inferior actions. Conversely, GEAR resolves this issue. Due to our absolute zero-order gating, the Grey samples are rigorously suppressed and clustered strictly near zero. The

higher end of GEAR's imitation spectrum is populated by valid, geometrically aligned targets (Blue samples), ensuring that the policy is only updated toward superior behaviors.

In conclusion, the weight visualizations confirm that naive relative weighting inherently leads to unstable, sparse, and heavily flawed gradient assignments. By structurally integrating absolute gating and geometric alignment, GEAR ensures a stable, dense, and high-quality imitation process.

C.2 Comparison with Classic RL Baselines

To further contextualize the performance of GEAR, we benchmark it against two representative classic reinforcement learning paradigms: **DreamerV3** (a SOTA MBRL algorithm) and **SAC** (a standard model-free baseline).

Crucially, as depicted by the x-axis in Figure. 6, both DreamerV3 [9] and SAC [31] are trained for a full 10M environment interaction steps. In stark contrast, GEAR achieves convergence much earlier, typically requiring fewer than 2M steps.

The results highlight a profound gap in sample efficiency and capability. As a model-free method, SAC struggles with the high dimensionality and complex contact dynamics of the `HumanoidBench` tasks, largely flatlining near zero returns on challenging environments like `run`, `sit_hard`, and `stair`. While the model-based DreamerV3 eventually discovers competent behaviors on several tasks (e.g., `crawl`, `stand`, and `walk`) toward the end of its 10M-step training budget, its learning pace is exceedingly slow. GEAR not only matches or surpasses the final asymptotic performance of DreamerV3 on almost all tasks but does so using approximately **one-fifth** of the interaction steps.

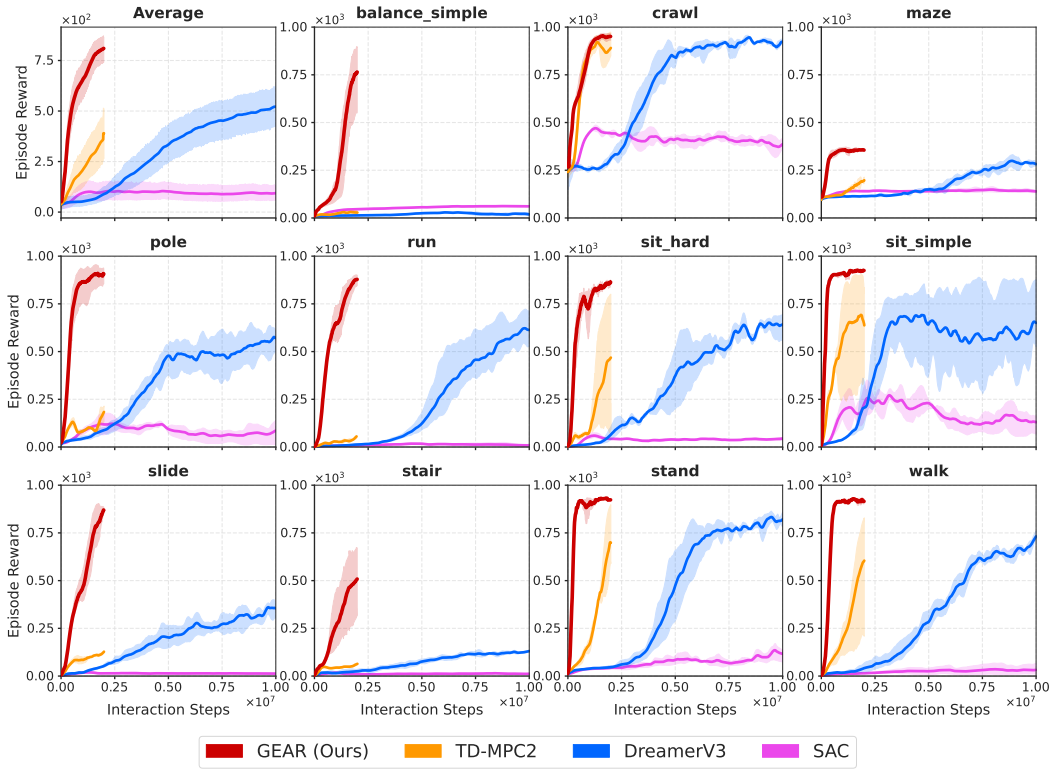

Figure 6: **Comparison with classic RL baselines.** Training curves of our method alongside DreamerV3 and SAC. The experimental results for DreamerV3 and SAC are directly sourced from the official benchmark repository: https://github.com/carlosferrazza/humanoid-bench.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline our primary contributions, theoretical findings, and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See detailed proofs in Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details are provided in Appendix and the code is linked anonymously in the Abstract.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is linked anonymously in the Abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See implementation details in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The curves and shaded areas in our paper represent the mean and 95% confidence intervals (CIs) over 5 independent runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section Computational Resources in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and ensure that our paper complies with them.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper focuses on fundamental algorithmic improvements to MBRL in simulated environments. As foundational research without a direct path to specific, deployable applications, it does not present immediate positive or negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cite the original paper that produced the code package or dataset correctly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for editing and formatting.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.