Iterative Gumbel Planning for Continuous Control

Shaohuai Liu[1,*] **Weirui Ye**[2,*] **Yilun Du**[3,†] **Le Xie**[3,†]
[1]Texas A&M University [2]MIT [3]Harvard University
liushaohuai5@tamu.edu
[*]Equal contribution. [†]Co-corresponding authors.
https://liushaohuai5.github.io/ez_igp/

Abstract

Search-based planning remains a powerful policy improvement paradigm in reinforcement learning, driving milestone successes in discrete domains. However, extending this paradigm to continuous control exposes a structural mismatch: search evaluates a finite set of action candidates, whereas continuous RL typically relies on Gaussian policies. We show that winner-action distillation into a continuous Gaussian induces a *projection error* and a stepwise variance-expanding gradient tendency when the selected action is sufficiently far from the current mean. Empirically, this tendency is associated with elevated terminal variance and a return gap on the evaluated tasks. To address this, we propose **Iterative Gumbel Planning (IGP)**. By adopting a per-dimension factorized categorical policy, IGP naturally aligns the policy support with the search candidates, eliminating the projection error and reducing the residual approximation to a bounded discretization error. To bypass the combinatorial explosion of high-dimensional discrete action spaces, IGP pairs Gumbel Top-K candidate sampling with an MPPI-style multi-round iterative search to refine the policy at each state. Empirically, IGP achieves state-of-the-art aggregate performance and sample efficiency on DMControl and HumanoidBench. Furthermore, it produces more stable action sequences with lower temporal variance, demonstrating its practical potential for real-world robotic deployment.

1 Introduction

Continuous control is a key challenge for reinforcement learning, with applications across robotics, control, and beyond. The dominant difficulty lies in credit assignment over long horizons in high-dimensional action spaces. Search-based planning offers a principled response: by combining a learned model with value-guided lookahead, it propagates value estimates over multiple steps and selects actions through explicit comparison rather than single-step bootstrapping. This paradigm has driven major progress on discrete-action benchmarks, with the AlphaGo and MuZero families [1–5] setting the state of the art. Extending it to continuous control is therefore natural, and recent work has begun to explore this direction [3, 6]. Yet the continuous setting has not seen comparable dominance, and we argue that the reason lies in a structural mismatch between search and the policy class.

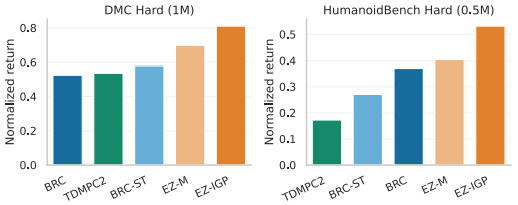


Figure 1: **Factorized policies achieve SOTA performance.** Across high-dimensional continuous-control benchmarks, our iteratively refined policy consistently attains strong returns.

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

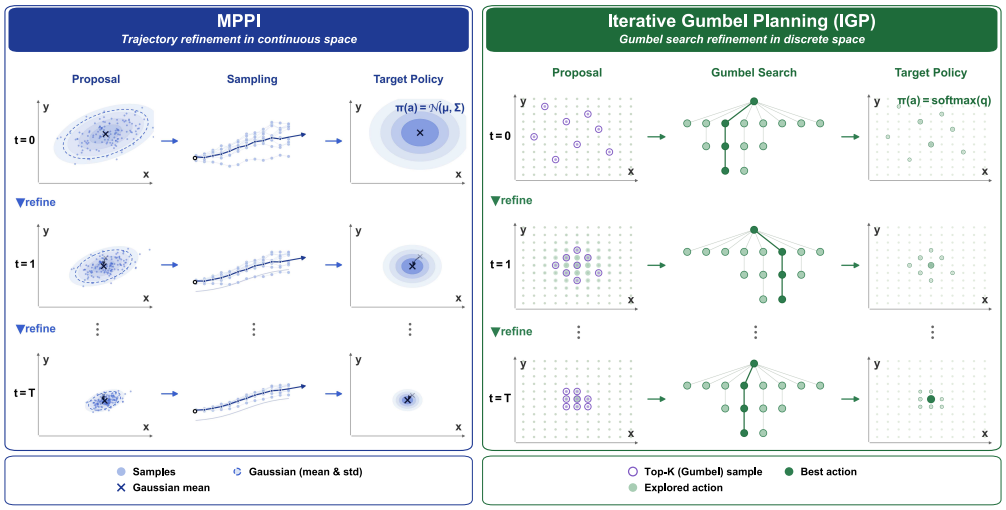


Figure 2: **Comparison of MPPI and Iterative Gumbel Planning (IGP).** While standard MPPI refines trajectories over an unbounded continuous action space, IGP natively operates in a factorized discrete policy space. By utilizing per-dimension Gumbel Top-K sampling, IGP executes multi-round iterative search to systematically refine a categorical root policy at each state.

Search is fundamentally a discrete improvement operator: at each state, it evaluates a finite set of candidate actions and returns a categorical distribution over them. In contrast, existing search-based continuous-control methods inherit Gaussian policies from policy-gradient RL [7, 8] and distill the search output into them via maximum likelihood. Since the selected action is sampled from the Gaussian and rarely coincides with its mean, maximum likelihood not only shifts the mean toward the target but can also encourage larger variance to cover the sampled candidate. We show that winner-action distillation produces a stepwise variance-expanding gradient tendency when the selected action is sufficiently far from the current mean.

To remove this projection error, the policy should be structurally aligned with the discrete search output. A natural solution is to discretize the continuous action space. Although discretization introduces its own approximation gap, we show that this gap is bounded and controllable by the grid resolution, replacing the non-vanishing Gaussian projection error with an explicitly characterized discretization error. The key remaining challenge is the $V^{|A|}$ combinatorial explosion of joint discretization, which has prevented this idea from scaling to high-dimensional control domains.

To resolve this bottleneck, we propose **Iterative Gumbel Planning (IGP)**. IGP operates on a per-dimension factorized categorical policy, keeping the parameter count linear in $|A|$ while aligning the policy support with the search operator. Each dimension is sampled without replacement and equally ranked samples are aligned to form K joint candidates. This is a scalable structured approximation, not exact without-replacement sampling from the exponentially large joint factorized distribution. Crucially, IGP treats search not as a single-pass procedure, but as an MPPI-style multi-round refinement process that repeatedly improves the root policy at the same state.

We implement IGP on EfficientZero-Multitask (EZ-M) [9] and evaluate it on DMControl [10] and HumanoidBench [11], covering high-dimensional locomotion and whole-body humanoid control. IGP achieves state-of-the-art performance, improves sample efficiency over strong model-based baselines, and produces action sequences with lower temporal variance at evaluation, which is relevant to real-world robotic deployment. **Our contributions are**: (1) We characterize the variance-expanding gradient tendency induced by winner-action Gaussian distillation and contrast its projection error with the bounded, controllable error of discretization; (2) We propose Iterative Gumbel Planning (IGP), which performs MPPI-style multi-round refinement over a factorized categorical policy via Gumbel Top-K sampling, bypassing Gaussian projection; (3) IGP achieves state-of-the-art sample efficiency on DMControl and HumanoidBench, yielding significantly lower temporal action variance and smoother control behaviors at evaluation.

2 Related Work

Discrete policies for continuous control. Discrete action representations have long been explored to enable value-based learning and stabilize continuous control. To address the combinatorial explosion in high-dimensional action spaces, prior work often relies on per-dimension factorization [12, 13] or hierarchical action selection [14, 15]. More recently, action discretization has also been adopted in vision-language-action (VLA) models [16–19], mainly due to the autoregressive format of pretrained transformers. In contrast, our work studies discretization from the perspective of search-based RL optimization, aiming to align the policy support with finite-support search targets in high-dimensional continuous control.

Search-based policy improvement. Search-based methods treat lookahead planning as a policy improvement operator and have driven major progress in discrete-action domains [1, 2, 5, 4]. Existing adaptations to continuous control typically sample candidates from a Gaussian proposal and perform search over this continuous subset [3, 6]. However, distilling the resulting finite-support search target back into a continuous Gaussian introduces projection errors and variance inflation, a structural limitation that we formally analyze in §4.

Iterative refinement in model-based planning. Model-based planning methods improve decisions through iterative refinement. MPPI [20], TD-MPC2 [21], and related trajectory optimization methods such as CEM repeatedly sample, evaluate, and update continuous action proposals. These approaches naturally operate on continuous distributions, whereas our work translates iterative refinement to a discrete, factorized policy space, enabling search-style improvement without Gaussian projection.

3 Preliminaries

We build IGP on MuZero-style search-based policy improvement. We briefly review two relevant components: Sampled MuZero, which extends MuZero to large action spaces through sampling, and Gumbel MuZero, which formulates search as a finite-budget policy-improvement operator.

Sampled MuZero with Gaussian proposals. Sampled MuZero [3] extends MuZero to large or continuous action spaces by sampling a small candidate set from a proposal distribution β_θ, typically a Gaussian policy. Given candidates $\mathcal{A}_K(s) = \{a_1, \dots, a_K\} \sim \beta_\theta(\cdot \mid s)$, MCTS is applied only over this finite set, producing an improved search policy π_{imp} summarized by visit counts, with $a_{\text{sel}} = \arg\max_k \pi_{\text{imp}}(a_k \mid s)$. This finite-support target is distilled back into the proposal policy, either through a cross-entropy/KL loss restricted to $\mathcal{A}_K(s)$ or by maximizing $\log \beta_\theta(a_{\text{sel}} \mid s)$.

Policy improvement with Gumbel search. Gumbel MuZero [4] casts search as a policy-improvement operator under a finite simulation budget. Given a prior $\beta_\theta(\cdot \mid s)$, it perturbs the prior logits with i.i.d. Gumbel noise and applies sequential halving over the selected candidates, allocating simulations to actions that are both likely under the prior and promising under value-guided search. Under suitable assumptions, such as reliable value estimates and correct backups, the resulting search policy $\pi_{\text{imp}}(\cdot \mid s)$ improves over the prior and is used as the training target for β_θ. We adopt this construction for the inner search step throughout this paper.

4 Analysis and Methodology

In this section, we present the structural analysis and algorithmic design of our approach. As conceptually illustrated in **Figure 3**, we analyze a winner-action Gaussian distillation objective that exhibits a variance-expanding gradient tendency under the stated local assumptions, and contrast its continuous projection with the bounded discretization error of a support-aligned categorical policy (§4.1). To resolve this mismatch, we introduce Iterative Gumbel Planning (IGP), which natively aligns the policy support with the discrete search candidates (§4.2), trading the continuous projection error for a rigorously bounded discretization error (§4.3).

4.1 Why Gaussian policies fail under search-based distillation

Consider a Gaussian policy with diagonal covariance $\beta_\theta(\cdot \mid s) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$ at a fixed state s. Action samples are drawn via reparameterization, $a_k = \mu + \sigma \odot \epsilon_k, \quad \epsilon_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I), \quad k =$

$1, \ldots, K$, and the winner action is the candidate maximizing the search-estimated value $\hat{Q}$: $a_{\text{sel}} \in \arg\max_{k \in \{1,\ldots,K\}} \hat{Q}(s, a_k)$. Following the winner-based maximum-likelihood loss in Gumbel MuZero [4], the policy update maximizes the log-likelihood at the winner:

$$\mathcal{L}(\mu, \sigma; s) = -\log \beta_\theta(a_{\text{sel}} \mid s). \tag{1}$$

Distilling a single winner into the Gaussian via (1) is itself a projection step: a finite-support search target is fit by a continuous density. The following result characterizes the expected directional pressure on the scale parameter under a local first-order model.

Proposition 1 (Directional inflation of Gaussian policy variance)**.** *Assume the action value admits a local first-order approximation around μ,*

$$\hat{Q}(s, a) \approx \hat{Q}(s, \mu) + g^\top (a - \mu) + o(\|a - \mu\|), \quad g \neq 0.$$

Let $w = \sigma \odot g$, and let the winner a_{sel} be treated as a fixed target in the policy update. Then under (1),

$$\mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial \log \sigma_i}\right] = -\frac{w_i^2}{\|w\|^2}\left(\mathbb{E}[\epsilon_{(K)}^2] - 1\right) \leq 0,$$

where $\epsilon_{(K)} = \max_{1 \leq k \leq K} \epsilon_k$ with $\epsilon_k \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$. The inequality is strict whenever $w_i \neq 0$ and $K \geq 3$.

Remark 1 (Weighted search targets)**.** *The same scale tendency is not restricted to a one-hot winner target. For fixed normalized search weights q_k and*

$$\mathcal{L}_q = -\sum_{k=1}^{K} q_k \log \mathcal{N}\left(a_k; \mu, \operatorname{diag}(\sigma^2)\right), \qquad \sum_k q_k = 1,$$

the per-dimension scale gradient is

$$\frac{\partial \mathcal{L}_q}{\partial \log \sigma_i} = 1 - \sum_{k=1}^{K} q_k \epsilon_{k,i}^2.$$

Thus the update expands variance whenever search places sufficient mass on candidates with above-baseline standardized displacement; winner-action MLE is the one-hot special case. A quadratic variance penalty adds $2\lambda\sigma_i^2$ to this gradient, creating a task- and stage-dependent trade-off between suppressing inflation and preserving proposal diversity.

The winner is sampled from β_θ and almost surely does not coincide with μ, so the maximum-likelihood objective raises $\beta_\theta(a_{\text{sel}} \mid s)$ via two mechanisms: shifting μ toward a_{sel}, and enlarging σ to raise the density at a_{sel}. Proposition 1 identifies this update-level variance-expanding mechanism, while Figure 4(a) shows that it accumulates into a sustained training-time effect under practical finite-budget learning. The full multi-dimensional proof is in Appendix A.1.

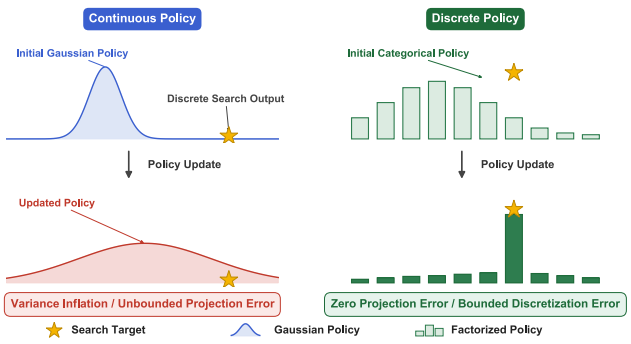


Figure 3: **Error trade-off in search-guided policy improvement.** (*Left*) Gaussian distillation inherently inflates policy variance due to an unbounded projection error. (*Right*) Our factorized discrete policy naturally aligns with the search support for exact distillation, replacing this unbounded error with a bounded discretization error.

We next show that variance inflation translates into a return gap, even at a converged policy mean.

Proposition 2 (Inflated variance lowers Q)**.** *Suppose the proposal policy is converged in the sense that $\mu = a^*$, and $Q(s, a)$ is locally m-strongly concave around a^*, a standard regularity assumption near a local optimum. Then*

$$\mathbb{E}_{a \sim \beta_\theta(\cdot|s)}[Q(s, a)] \leq Q(s, a^*) - \frac{m}{2} \sum_i \sigma_i^2.$$

Corollary 1 (Return degradation under persistent variance inflation)**.** *If the local concavity condition in Proposition 2 holds along states in the visitation distribution d^π, then*

$$J(\pi^*) - J(\pi) \geq \frac{m}{2(1-\gamma)} \mathbb{E}_{s\sim d^\pi}\left[\sum_i \sigma_i^2\right].$$

Under the local concavity assumptions, sustained policy variance induces a return gap proportional to $\sum_i \sigma_i^2$. Figure 4(a) empirically demonstrates that the elevated variance persists over the practical training horizon, making this bound operational for the evaluated setting. Proofs are in Appendix A.2. This motivates aligning the policy support with the search candidates, which we develop next.

4.2 Iterative Gumbel Planning

To avoid the variance inflation analyzed in §4.1, we represent the policy on the same finite support that the search operates on. We instantiate this idea as **Iterative Gumbel Planning (IGP)**, with four components: a per-dimension factorized categorical policy, Gumbel Top-K candidate sampling, search-output distillation, and multi-round root refinement.

Per-dimension factorized categorical policy. For a continuous-control task with $|A|$ action dimensions and bounded action space $[-1,1]^{|A|}$, we discretize each dimension into V bins with centers $\{c_v\}_{v=1}^V \subset [-1,1]$, giving a grid resolution $\Delta = 2/V$. The policy network outputs per-dimension logits $p_\theta(s) \in \mathbb{R}^{|A|\times V}$, where $p_{\theta,j,v}(s)$ is the logit of bin v on dimension j. The policy is the product of $|A|$ independent categorical distributions over V bins, and a discrete code $\mathbf{z} = (z_1, \ldots, z_{|A|}) \in \{1,\ldots,V\}^{|A|}$ is decoded into a continuous action via

$$\mathrm{dec}(\mathbf{z}) = \frac{2\mathbf{z}-1}{V} - 1 \in [-1,1]^{|A|}. \tag{2}$$

This factorization keeps the parameter count strictly linear in $|A| \cdot V$, fundamentally avoiding the combinatorial explosion of a joint discrete action space. Per-dimension factorization has been used in prior on-policy work [13]. Following standard practice in continuous control RL, where proposal distributions are almost universally parameterized as diagonal Gaussians $\mathcal{N}(\mu, \mathrm{diag}(\sigma^2))$ (e.g., SAC [8], TD-MPC2 [21], Sampled MuZero [3], EfficientZeroV2 [6]), our per-dimension factorized categorical policy adopts the *same independence structure* across action dimensions. Both proposal families therefore omit explicit residual cross-dimensional covariance, although their support and optimization geometry differ. Search evaluates every proposed joint action as a whole, but factorization can still limit proposal quality when strong dependencies must be represented before search.

Gumbel Top-K candidate sampling. At each state s, IGP constructs K joint candidates using a per-dimension Gumbel Top-K trick. For each dimension j, we draw i.i.d. Gumbel noise $g_{j,v} \sim \mathrm{Gumbel}(0,1)$ and form perturbed scores

$$\tilde{p}_{j,v}(s) = p_{\theta,j,v}(s) + g_{j,v}, \tag{3}$$

then take the top-K indices along the bin dimension:

$$\mathcal{I}_j(s) = \mathrm{TopK}\big(\tilde{p}_{j,1:V}(s),\, K\big) \subset \{1,\ldots,V\}, \qquad |\mathcal{I}_j(s)| = K. \tag{4}$$

We then construct K joint candidates by aligning the k-th selected bin across all dimensions:

$$z_j^{(k)}(s) = \mathcal{I}_j(s)[k], \qquad \mathbf{z}^{(k)} = \big(z_1^{(k)}, \ldots, z_{|A|}^{(k)}\big), \qquad a^{(k)} = \mathrm{dec}\big(\mathbf{z}^{(k)}\big). \tag{5}$$

The Top-K step samples distinct bins without replacement within each dimension, then uses rank alignment to form joint candidates. It is not equivalent to exact Top-K sampling without replacement from the full product distribution; it is a parallelizable structured approximation that avoids enumerating the $V^{|A|}$ joint grid.

Distillation from the search-improved policy. Given the candidate set $\{a^{(k)}\}_{k=1}^K$, the search procedure produces a distribution $\pi_{\mathrm{imp}}(a^{(k)} \mid s)$ supported on the K candidates, with $\sum_{k=1}^K \pi_{\mathrm{imp}}(a^{(k)} \mid s) = 1$. We distill π_{imp} into the per-dimension categorical policy by marginalizing it onto each action dimension:

$$t_{j,v}(s) = \sum_{k=1}^K \pi_{\mathrm{imp}}(a^{(k)} \mid s)\, \mathbb{I}\Big[z_j^{(k)} = v\Big], \tag{6}$$

yielding a valid distribution $t_{j,\cdot}(s)$ on $\{1, \ldots, V\}$ for each j. The distillation loss is the sum of per-dimension cross-entropies:

$$\mathcal{L}_{\text{distill}}(\theta) = -\mathbb{E}_{s\sim\mathcal{D}}\left[\sum_{j=1}^{|A|}\sum_{v=1}^{V} t_{j,v}(s)\log p_{\theta,j}(v \mid s)\right]. \tag{7}$$

For training stability, we adopt the hard-label special case as default: setting the target to the selected candidate $a^{(k_{\text{sel}})}$ with $k_{\text{sel}} = \arg\max_k \pi_{\text{imp}}(a^{(k)} \mid s)$ reduces (6) to one-hot labels $t_{j,v}(s) = \mathbb{I}[z_j^{(k_{\text{sel}})} = v]$, and (7) becomes a multi-class classification objective across the $|A|$ action dimensions.

Multi-round root refinement. Because the support of π_{imp} coincides with the policy support, the distillation step in (7) is exact, as we formalize in §4.3. This allows IGP to iterate search and refinement at the same state without accumulating projection error. At iteration t, MCTS evaluates the current candidate set and returns $\pi_{\text{imp}}^{(t)}(a^{(k)} \mid s)$ together with search scores $q^{(k)}(s)$. These outcomes refine the root logits. Let $a_t^\star = \arg\max_{a\in\mathcal{A}_t(s)} \widehat{Q}(s,a)$. The next round retains this predecessor and samples only the remaining $K-1$ candidates:

$$\mathcal{A}_{t+1}(s) = \{a_t^\star\} \cup \widetilde{\mathcal{A}}_{t+1}(s), \qquad |\widetilde{\mathcal{A}}_{t+1}(s)| = K-1.$$

For a fixed scoring function $\widehat{Q}$, this gives $\max_{a\in\mathcal{A}_{t+1}} \widehat{Q}(s,a) \geq \max_{a\in\mathcal{A}_t} \widehat{Q}(s,a)$, so refinement cannot discard the best candidate already found. This is only a candidate-retention guarantee, not monotonic policy improvement: changing value estimates, finite search budgets, and later learning updates can still make realized performance non-monotonic. This is an MPPI-style outer loop in discrete policy space: unlike TD-MPC2 [21], which refines on continuous support, IGP refines the per-dimension categorical logits via Gumbel-sampled discrete candidates.

For the per-step logit update, we use a local kernel around the selected bin index $z_j^{(k_{\text{sel}})}$:

$$\kappa_r\left(v, z_j^{(k_{\text{sel}})}\right) = \max\left(0,\, 1 - \tfrac{|v - z_j^{(k_{\text{sel}})}|}{r+1}\right)\mathbb{I}\left[|v - z_j^{(k_{\text{sel}})}| \leq r\right], \tag{8}$$

with $r = 1$ in practice. After normalizing κ_r to sum to one over v, we use it as a local bonus $b_{j,v}(s)$, center it under the current categorical $\bar{p}_{\theta,j}(\cdot \mid s) = \text{softmax}(p_{\theta,j,\cdot}(s))$, and update the logits:

$$\tilde{b}_{j,v}(s) = b_{j,v}(s) - \sum_{u=1}^{V} \bar{p}_{\theta,j}(u \mid s)\, b_{j,u}(s), \qquad p'_{\theta,j,v}(s) = p_{\theta,j,v}(s) + \eta\, \tilde{b}_{j,v}(s), \tag{9}$$

with $\eta = 1.0$. This local update spreads probability mass to neighbors of the selected bin, yielding a smooth search-guided refinement rather than an abrupt one-hot replacement.

4.3 Error trade-off: projection error vs. discretization error

During the outer learning phase, the target policy $\pi_{\text{imp}}(\cdot \mid s)$ is strictly supported on the finite candidate set evaluated by search. Distilling this discrete target back into the parametric proposal fundamentally acts as a projection onto the policy class:

$$\beta_{\theta^+}(\cdot \mid s) \approx \arg\min_\theta \mathbb{E}_{a\sim\pi_{\text{imp}}(\cdot|s)}[-\log \beta_\theta(a \mid s)]. \tag{10}$$

For a categorical policy, this objective reduces to cross-entropy on the candidate support; for a Gaussian, it is the negative log-density at the candidates.

Gaussian policies: non-vanishing projection error. For a Gaussian $\beta_G(\cdot \mid s) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$ and a one-hot target $\pi_{\text{imp}}(a \mid s) = \delta(a - a_{\text{sel}})$, the projection objective

$$\mathcal{L}_G(s) = \tfrac{1}{2}\sum_i \tfrac{(a_{\text{sel},i} - \mu_i)^2}{\sigma_i^2} + \sum_i \log\sigma_i + C \tag{11}$$

has the same scale-parameter behavior as the winner-MLE loss in Proposition 1: whenever the residual $|a_{\text{sel},i} - \mu_i| > \sigma_i$, gradient descent on $\mathcal{L}_G$ enlarges σ_i. In our experiments, the winner target frequently remains far from the current mean and the Gaussian variance stays elevated throughout training, connecting the stepwise mechanism to a sustained long-horizon optimization effect.

Discretized policies: zero projection error on the search support. For the per-dimension factorized categorical policy β_D of §4.2, parameterized by softmax over $\{p_{\theta,j,v}\}$, when search selects a grid action a_{sel}, the one-hot target is achievable in the limit of the policy class:

$$\inf_{\theta} \mathbb{E}_{a\sim\delta_{a_{\text{sel}}}}[-\log \beta_D(a \mid s;\theta)] = 0, \tag{12}$$

where the infimum reflects the standard softmax parameterization (achieved as logits go to infinity). The projection step is therefore exact on the shared support: there is no residual that the policy must absorb as variance.

Bounded discretization error. Discretization introduces its own approximation gap when the optimal continuous action $a^*(s)$ does not lie on the grid. We bound this gap below.

Lemma 1 (Discretization error)**.** *Let $\tilde{a}^*(s)$ be the nearest grid point to $a^*(s)$ at resolution $\Delta = 2/V$, and assume $Q(s,\cdot)$ is L-Lipschitz in a. Then*

$$\left|Q(s,a^*(s)) - Q(s,\tilde{a}^*(s))\right| \le L\,\frac{\sqrt{|A|}}{2}\,\Delta, \quad J(\pi^*) - J(\tilde{\pi}^*) \le \frac{L}{1-\gamma}\cdot\frac{\sqrt{|A|}}{2}\,\Delta.$$

For one-hot targets on the grid, the categorical projection error can vanish, whereas Gaussian winner distillation may retain a scale-dependent residual during finite training. The discretization error is explicitly characterized and shrinks at rate $\Delta = 2/V$ as the bin count V increases. The trade-off is favorable: IGP exchanges a continuous-support projection mismatch for a controllable representation error.

5 Experiments

In this section, we evaluate IGP by implementing it on top of EfficientZero-Multitask (EZ-M) [9], and refer to the resulting method as EZ-IGP. We continue to adopt the multi-task training setting. Our experiments aim to answer the following questions: (1) How does EZ-IGP perform compared with established baselines? (2) What is the empirical variance distribution when training with a Gaussian policy? (3) What does the learned policy behavior look like in practical applications? (4) How does performance vary with the number of discretization bins? (5) How does performance change with the number of search rounds? We answer these questions in the following subsections, with additional experimental details and training curves provided in Appendix C.

5.1 Experimental Settings

Benchmarks. We evaluate EZ-IGP on two challenging continuous-control benchmarks: DMControl [10] and HumanoidBench [11]. Following [22, 9], we focus on *DMC-Hard* and *HB-Hard*. *DMC-Hard* includes 7 high-dimensional locomotion tasks: humanoid `stand`, `walk`, and `run`, and dog `stand`, `walk`, `run`, and `trot`. *HB-Hard* consists of hand-control humanoid tasks, including 9 locomotion tasks as well as `sit-simple`, `sit-hard`, `balance-simple`, `balance-hard`, and `reach`. We train all methods for 1 million environment steps; DMControl uses an action repeat of 2, while HumanoidBench uses no action repeat.

Baselines. On HumanoidBench, we compare EZ-IGP with representative single-task and multi-task baselines, including TD-MPC2 [21], MH-SAC [23], BRC [22], and EZ-M [9]. On DMControl, we compare against TD-MPC2 and EZ-M. EZ-IGP follows the same architectures and training objectives as EZ-M, but replaces its Gaussian policy with our iteratively refined factorized policy. Baseline scores are taken from published results when available, and otherwise from re-evaluations of official implementations. Hyperparameter details are provided in Appendix E.

Evaluation Metrics. We report normalized aggregate scores as described in Appendix D, and provide full training curves with original returns in Appendix C.

5.2 Overall Performance

Main results on DMControl dog and humanoid tasks. Figure 6 and Table 1 show results on the DMControl dog and humanoid locomotion tasks. EZ-IGP consistently outperforms or matches all baselines across the seven environments. The improvement is especially pronounced on the dog tasks. On the four dog tasks, EZ-IGP learns faster and achieves the highest reported per-seed

Table 1: Scores for DMC Hard (1M) tasks, reported as mean ± standard deviation. Each score is computed by first taking the maximum return within each seed, and then computing the mean and standard deviation across seeds. We **bold** the best result for each task.

Task	TD-MPC2	BRC	BRC-ST	EZ-M-ST	EZ-M	EZ-IGP(ours)
dog-stand	817.367±23.354	945.557±35.713	913.862±57.743	947.730±5.045	945.741±7.064	**974.790±8.358**
dog-walk	719.833±68.214	809.929±147.561	764.241±97.900	778.395±110.429	915.123±6.744	**958.983±5.187**
dog-run	228.800±22.439	281.717±49.678	299.228±73.987	241.965±70.275	405.874±50.808	**613.257±6.546**
dog-trot	500.033±16.631	516.288±130.027	588.662±95.487	466.416±122.449	815.958±76.640	**942.814±3.341**
humanoid-stand	663.733±32.515	737.042±201.735	751.334±166.116	589.152±97.187	772.273±8.943	**903.719±7.734**
humanoid-walk	628.233±16.280	314.038±271.480	569.773±92.383	531.614±5.920	779.404±39.126	**911.542±7.005**
humanoid-run	190.200±14.062	62.398±73.964	161.155±42.949	153.026±33.236	259.030±25.534	**369.468±5.633**

peak scores. This indicates that the proposed factorized iterative policy is not only effective for dexterous manipulation-like tasks, but also transfers well to high-dimensional locomotion domains with complex contact dynamics.

The same pattern appears on the humanoid tasks: EZ-IGP obtains the best reported peak performance and generally shows stronger sample efficiency than BRC, BRC-ST, TD-MPC2, and EZ-M, with especially clear gains on `humanoid-walk` and `humanoid-run`, where several baselines plateau at lower scores. This suggests that iterative action refinement becomes more beneficial as tasks require more accurate long-horizon coordination.

Taken together, these results show that EZ-IGP provides a simple but effective alternative to continuous Gaussian policy parameterization by factorizing the action space into discrete per-dimension choices and improving candidate actions through iterative refinement. The resulting policy achieves strong sample efficiency, competitive stability, and state-of-the-art or near state-of-the-art performance across both HumanoidBench and DMControl high-dimensional control tasks.

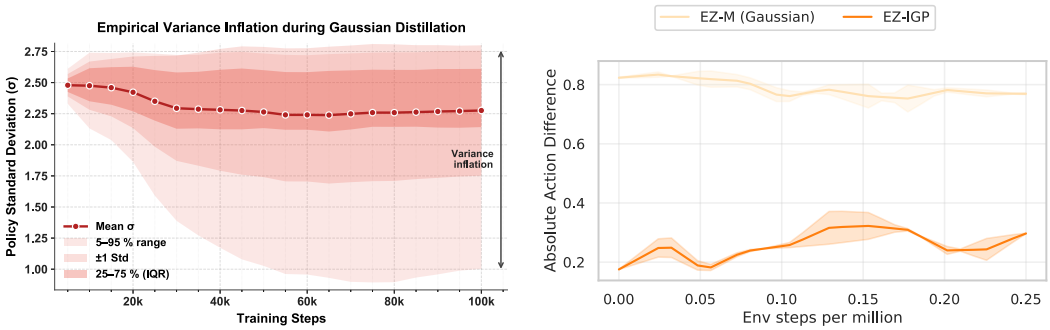
Main results on HumanoidBench. Figure 7 reports learning curves on the high-dimensional HumanoidBench tasks, while full per-task scores and uncertainty are provided in Table 2. Overall, EZ-IGP achieves strong and consistent performance across a broad set of challenging dexterous-control environments. Compared with EZ-M, EZ-IGP often learns substantially faster and reaches higher final returns, especially on tasks such as `h1hand-walk`, `h1hand-run`, `h1hand-pole`, and `h1hand-maze`. These tasks require coordinated control over many action dimensions, where Gaussian policy optimization can be sensitive to variance and exploration scale; in contrast, EZ-IGP provides a structured factorized action proposal and further improves selected actions through iterative planning.

The advantage of EZ-IGP is particularly clear in tasks where successful behavior requires precise multi-dimensional coordination rather than simply discovering a coarse locomotion pattern. For example, in `h1hand-run` and `h1hand-pole`, EZ-IGP rapidly improves after the early training phase and eventually reaches a much higher score than BRC, BRC-ST, and TD-MPC2. On `h1hand-reach`, EZ-M achieves the highest score, while EZ-IGP remains stronger than TD-MPC2, BRC, and BRC-ST.

There are also clear failure cases. EZ-IGP trails the strongest baseline on `crawl`, `sit`, and `balance`. Capacity is a substantial confound: reducing BRC from approximately 1B to 16M parameters lowers its HB-Hard normalized score from 0.576 to 0.487 ± 0.038, while EZ-IGP reaches 0.743 at 16M parameters. Nevertheless, BRC-16M remains stronger on the sitting tasks, suggesting a residual task-specific inductive-bias gap. The present comparisons do not isolate finite bin resolution, action factorization, exploration, and objective choice, so we avoid attributing the gap to a single cause.

5.3 Policy variance comparisons

Figure 4(a) empirically demonstrates the sustained training-time consequence of the update-level mechanism in Proposition 1: the standard deviation rises early and remains high throughout the observed training horizon, including its final stage. The median σ stabilizes around a high level, and both the interquartile range and the 5–95% range suggest that this behavior is not caused by a few outlier dimensions or tasks. Figure 4(b) further shows that this elevated variance is reflected in the realized policy behavior: EZ-M with a Gaussian policy produces much larger temporal action differences, while EZ-IGP maintains substantially smoother action sequences. Such action fluctuations can weaken control precision and lead to jittery behaviors in continuous-control tasks. The qualitative comparison videos can be accessed at the project website.



(a) Policy variance distribution in Gaussian distillation. (b) Average $|\Delta a|$ across three humanoid tasks.

Figure 4: **Temporal variance comparisons. (a)** The Gaussian policy standard deviation (σ) rises early and remains elevated throughout the observed training horizon; the solid line shows the median and shaded regions show the interquartile and 5–95% ranges. **(b)** The Gaussian policy maintains much larger action changes across adjacent timesteps during evaluation, whereas EZ-IGP consistently produces smaller action differences, suggesting smoother and less jittery control behavior.

Refinement-matched Gaussian control. To isolate policy parameterization from multi-round refinement, we train a Gaussian proposal with the same backbone, $K = 8$ candidates, $N = 16$ simulations per round, two refinement rounds, winner-action target, and unchanged SEM/categorical path. EZ-IGP improves all three DMC humanoid tasks by approximately 139–150 points (Table 3). In the matched Gaussian runs, `dist/policy_sigma` rises from 1.76 ± 0.16 to an early peak of 3.02 ± 0.16 and remains at 2.11 ± 0.04 over the final 10% of training. A separate two-seed control that distills a search-score-weighted mixture of all candidates also retains elevated final variance (2.67 ± 0.04) and underperforms winner-target Gaussian distillation on the same tasks. Thus, neither proposal refinement nor distributing target mass across candidates removes the observed scale pressure.

5.4 Ablation Study

Number of factorized bins. We investigate the effect of action discretization granularity by varying the number of bins V per action dimension. Figure 5(a) reports unscaled evaluation returns on three DMC tasks. While different bin counts yield comparable final performance on `humanoid_stand` and `humanoid_walk`, $V = 41$ achieves the highest mean on `humanoid_run`, suggesting that finer discretization can be beneficial but does not consistently improve performance. To assess whether this trend generalizes to more challenging control problems, we further evaluate different bin counts across all 14 HumanoidBench-Hard (HB-Hard) tasks, using three seeds per setting, 8 candidates, and 16 simulations per round. As shown in Table 4, $V = 31$ achieves the highest mean performance at $R = 2$ after 300k updates, with no further improvement from increasing V to 41. Together, these results suggest that moderate discretization provides sufficient action resolution for effective planning, without requiring an increasingly fine-grained action space. We therefore adopt $V = 31$ as a robust choice across benchmarks.

Number of refinement rounds. We next examine the effect of iterative proposal refinement by varying the number of rounds R while fixing $V = 31$. Unlike the bin count, which determines the resolution of the action support, R controls how often the search updates its proposal distribution and resamples candidates. With $R = 1$, the search performs no proposal refinement, whereas $R = 2$ introduces one refinement-and-resampling step. Figure 5(b) shows that two rounds achieve the highest final mean returns across all three DMC tasks. Increasing to four rounds reduces the terminal means relative to two rounds, although it does not consistently underperform a single round. The broader HB-Hard evaluation in Table 4 reinforces this pattern: increasing R from 1 to 2 improves the mean performance by 0.056, whereas further increasing to four rounds yields no additional gain. These findings indicate that a small number of refinement steps is sufficient to benefit from proposal adaptation, while excessive refinement may concentrate the proposal distribution and limit exploration. We thus select $R = 2$ to balance planning effectiveness and computational cost.

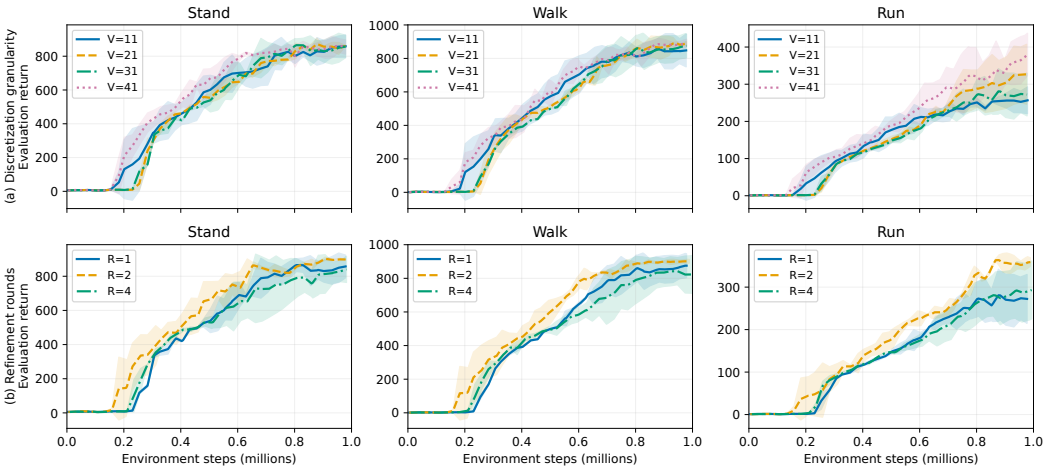


Figure 5: **Ablation studies on three DMC humanoid tasks.** Top: bin count; bottom: refinement rounds. Lines show mean unscaled evaluation returns over two runs, and shading denotes sample standard deviation. Curves are aligned by collected environment steps without endpoint extrapolation. Larger bin counts are not uniformly better; two rounds perform well across these tasks. See Table 4 for HB-Hard.

Sensitivity to the simulation budget. We vary only the candidate/simulation budget while fixing $V = 31$ and two refinement rounds. IGP remains trainable at $2/4$, but performance degrades monotonically as fewer candidates reduce root coverage and fewer simulations weaken candidate ranking (Table 5). The default $8/16$ setting is therefore an empirical quality–compute trade-off rather than a claim of budget invariance.

6 Discussion

This work identifies a structural mismatch between search-based policy improvement and Gaussian policies in continuous control. Since search improves over a finite set of candidate actions, distilling its output into a continuous Gaussian introduces a non-vanishing projection error that can inflate policy variance and degrade returns. IGP addresses this mismatch with a per-dimension factorized categorical policy whose support is aligned with search candidates, replacing Gaussian projection error with bounded discretization error. Empirically, EZ-IGP achieves strong performance on both DMControl and HumanoidBench, showing that factorized discrete policies can serve as a practical alternative to Gaussian policies in high-dimensional continuous control. The ablations show task-dependent effects of discretization and refinement. Together, DMC and the preliminary HB-Hard sweep support a moderate bin count and two rounds as a practical performance–computation trade-off, rather than establishing a universal optimum or a causal explanation for the effect of additional rounds.

Several **limitations** remain. IGP uses fixed uniform discretization and a factorized action representation, which may be suboptimal when action dimensions require different resolutions or exhibit strong dependencies. Its performance also depends on search hyperparameters such as bin count, candidate number, and refinement rounds. Future work could explore adaptive discretization, structured or autoregressive discrete policies, adaptive refinement schedules, and validation on real robots.

Acknowledgments and Disclosure of Funding

We thank Jeffrey Polasek and the Texas A&M University Department of Electrical and Computer Engineering IT team for their support of the Olympus computing cluster, which provided computational resources for this work. Funding is supported in part by Harvard Salata Institute for Climate and Sustainability.

Disclaimer

The views expressed in this paper are those of the authors and do not reflect the views of PJM Interconnection, L.L.C. or its Board of Managers, of which Le Xie is a member.

References

[1] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017. 1, 3

[2] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 3

[3] Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Mohammadamin Barekatain, Simon Schmitt, and David Silver. Learning and planning in complex action spaces. In *International Conference on Machine Learning*, pages 4476–4486. PMLR, 2021. 1, 3, 5

[4] Ivo Danihelka, Arthur Guez, Julian Schrittwieser, and David Silver. Policy improvement by planning with gumbel. In *International Conference on Learning Representations*, 2022. 3, 4

[5] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021. 1, 3

[6] Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. Efficientzero v2: Mastering discrete and continuous control with limited data. *arXiv preprint arXiv:2403.00564*, 2024. 1, 3, 5, 21

[7] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019. 2

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018. 2, 5

[9] Shaohuai Liu, Weirui Ye, Yilun Du, and Le Xie. Scaling tasks, not samples: Mastering humanoid control through multi-task model-based reinforcement learning. *arXiv preprint arXiv:2603.01452*, 2026. 2, 7

[10] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2, 7, 27

[11] 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. 2, 7, 27

[12] Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, volume 32, 2018. 3

[13] Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 5981–5988, 2020. 3, 5

[14] Weichao Mao, Kaiqing Zhang, Qiaomin Xie, and Tamer Basar. Poly-hoot: Monte-carlo planning in continuous space mdps with non-asymptotic analysis. *Advances in Neural Information Processing Systems*, 33:4549–4559, 2020. 3

[15] Younggyo Seo and Pieter Abbeel. Coarse-to-fine q-network with action sequence for data-efficient reinforcement learning. *arXiv preprint arXiv:2411.12155*, 2024. 3

[16] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3

[17] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[19] Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. Oat: Ordered action tokenization. *arXiv preprint arXiv:2602.04215*, 2026. 3

[20] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1714–1721. IEEE, 2017. 3

[21] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023. 3, 5, 6, 7

[22] Michal Nauman, Marek Cygan, Carmelo Sferrazza, Aviral Kumar, and Pieter Abbeel. Bigger, regularized, categorical: High-capacity value functions are efficient multi-task learners. *arXiv preprint arXiv:2505.23150*, 2025. 7

[23] Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *Advances in Neural Information Processing Systems*, 33:4767–4777, 2020. 7

Appendix Contents

A Proofs **14**

A.1 Proof of Proposition 1 14

A.2 Proof of Proposition 2 and Corollary 1 15

B Additional Experimental Results **17**

C Training Curves **19**

D Score Normalization **19**

E Hyperparameters and Details **21**

A Proofs

A.1 Proof of Proposition 1

Proof: Inflation of Gaussian policy variance (multidimensional perspective). Under the local first-order approximation,

$$\hat{Q}(s,a) = \hat{Q}(s,\mu) + g^\top(a-\mu) + o(\|a-\mu\|_2), \qquad g \neq 0.$$

Since

$$a_k = \mu + \sigma \odot \epsilon_k, \qquad \epsilon_k \overset{i.i.d.}{\sim} \mathcal{N}(0, I),$$

the first-order term becomes

$$g^\top(a_k - \mu) = g^\top(\sigma \odot \epsilon_k) = (\sigma \odot g)^\top \epsilon_k.$$

Define

$$w \coloneqq \sigma \odot g.$$

Then, up to the local first-order approximation, the winner index is

$$k^* \in \arg\max_{1 \le k \le K} w^\top \epsilon_k.$$

Let

$$u \coloneqq \frac{w}{\|w\|_2}.$$

For each sample ϵ_k, decompose it as

$$\epsilon_k = z_k u + \xi_k,$$

where

$$z_k \coloneqq u^\top \epsilon_k \sim \mathcal{N}(0,1), \qquad \xi_k \perp u, \qquad z_k \text{ is independent of } \xi_k.$$

Since

$$w^\top \epsilon_k = \|w\|_2 z_k,$$

the winner index is equivalently

$$k^* = \arg\max_{1 \le k \le K} z_k.$$

Hence the selected noise can be written as

$$\epsilon^* \coloneqq \epsilon_{k^*} = z_{(K)} u + \xi^*, \qquad z_{(K)} \coloneqq \max_{1 \le k \le K} z_k,$$

where ξ^* has the same distribution as the orthogonal component of a standard Gaussian sample and remains independent of $z_{(K)}$.

Now consider the winner-MLE loss

$$\mathcal{L}(\mu, \sigma; s) = -\log \pi_{\mu,\sigma}(a^*|s), \qquad a^* = \mu + \sigma \odot \epsilon^*,$$

with a^* treated as a fixed target. For a diagonal Gaussian policy,

$$\mathcal{L} = \sum_i \left(\log \sigma_i + \frac{(a_i^* - \mu_i)^2}{2\sigma_i^2} \right) + C,$$

where C is constant with respect to (μ, σ). Therefore,

$$\frac{\partial \mathcal{L}}{\partial \log \sigma_i} = 1 - \frac{(a_i^* - \mu_i)^2}{\sigma_i^2} = 1 - (\epsilon_i^*)^2.$$

Taking expectation,

$$\mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial \log \sigma_i} \right] = 1 - \mathbb{E}[(\epsilon_i^*)^2].$$

Using $\epsilon_i^* = u_i z_{(K)} + \xi_i^*$, we obtain

$$\mathbb{E}[(\epsilon_i^*)^2] = u_i^2 \mathbb{E}[z_{(K)}^2] + 2u_i \mathbb{E}[z_{(K)} \xi_i^*] + \mathbb{E}[(\xi_i^*)^2].$$

The cross term vanishes by independence and zero mean of the orthogonal component:

$$\mathbb{E}[z_{(K)} \xi_i^*] = 0.$$

Moreover, since a standard Gaussian has unit covariance and u is unit norm,

$$\mathbb{E}[(\xi_i^*)^2] = 1 - u_i^2.$$

Therefore,

$$\mathbb{E}[(\epsilon_i^*)^2] = u_i^2 \alpha_K + (1 - u_i^2), \qquad \alpha_K := \mathbb{E}[z_{(K)}^2].$$

Substituting back,

$$\mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial \log \sigma_i} \right] = 1 - \left(u_i^2 \alpha_K + 1 - u_i^2 \right) = -(\alpha_K - 1) u_i^2.$$

For $K \geq 3$, we have $\alpha_K > 1$, so

$$\mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial \log \sigma_i} \right] \leq 0,$$

with strict inequality whenever $u_i \neq 0$, equivalently whenever $(\sigma_i g_i) \neq 0$. This proves the claim. □

A.2 Proof of Proposition 2 and Corollary 1

Proof: Inflated variance lowers Q and return. For an arbitrary fixed state s, assume that $Q(s, a)$ is locally m-strongly concave around the optimal action $a^*(s)$, and that there exists a constant $m > 0$ such that

$$Q(s, a) \leq Q\big(s, a^*(s)\big) - \frac{m}{2} \|a - a^*(s)\|^2, \tag{13}$$

for all a in a neighborhood of $a^*(s)$.

Consider a Gaussian policy $a \sim \pi(\cdot|s) = \mathcal{N}\big(\mu(s), \Sigma(s)\big)$, where $\Sigma(s) = \mathrm{diag}(\sigma^2(s))$. Assume the policy is locally converged in the sense that $\mu(s) = a^*(s)$. Then

$$\mathbb{E}_{a \sim \pi(\cdot|s)}[Q(s, a)] \leq Q\big(s, a^*(s)\big) - \frac{m}{2} \mathbb{E}_{a \sim \pi(\cdot|s)} \|a - a^*(s)\|^2. \tag{14}$$

For the Gaussian policy above,

$$\mathbb{E}_{a \sim \pi(\cdot|s)} \|a - a^*(s)\|^2 = \mathrm{tr}\big(\Sigma(s)\big) = \sum_i \sigma_i^2(s), \tag{15}$$

and therefore

$$\mathbb{E}_{a \sim \pi(\cdot|s)}[Q(s, a)] \leq Q\big(s, a^*(s)\big) - \frac{m}{2} \sum_i \sigma_i^2(s). \tag{16}$$

Under the assumptions of Proposition 2, suppose additionally that for every state s visited by π, the action $a^*(s)$ maximizes $Q^{\pi^*}(s, a)$, i.e.,

$$a^*(s) \in \arg\max_a Q^{\pi^*}(s, a), \tag{17}$$

and the local quadratic upper bound holds on the support of $\pi(\cdot \mid s)$. Then

$$J(\pi^*) - J(\pi) \geq \frac{m}{2(1-\gamma)} \mathbb{E}_{s \sim d^\pi} [\mathrm{tr}(\Sigma(s))] \,. \tag{18}$$

For diagonal covariance $\Sigma(s) = \mathrm{diag}(\sigma_1^2(s), \ldots, \sigma_d^2(s))$, this becomes

$$J(\pi^*) - J(\pi) \geq \frac{m}{2(1-\gamma)} \mathbb{E}_{s \sim d^\pi} \left[\sum_{i=1}^{d} \sigma_i^2(s) \right] . \tag{19}$$

□

B Additional Experimental Results

Table 2: Scores for HumanoidBench-Hard (1M) tasks, reported as mean ± standard deviation across three seeds after taking the maximum return within each seed. We **bold** results above the success score, or the SoTA when all methods are below it.

Task	Random	Success	TD-MPC2	BRC	BRC-ST	EZ-M	EZ-IGP(ours)
h1hand-stand	11.973	800.0	209.256±52.474	**890.627±64.554**	344.159±309.290	**914.460±2.127**	**899.706±11.895**
h1hand-walk	2.505	700.0	235.559±93.954	622.684±139.376	**858.652±106.977**	**923.384±3.694**	**923.752±3.092**
h1hand-run	1.927	700.0	35.943±3.973	189.786±25.663	51.847±13.318	**822.343±43.723**	**889.872±7.244**
h1hand-stair	3.161	700.0	51.033±3.398	191.848±45.344	65.302±11.846	**441.692±19.294**	381.660±25.700
h1hand-crawl	278.868	800.0	**898.560±35.968**	740.718±59.136	**854.395±19.127**	665.573±70.116	647.794±59.109
h1hand-slide	3.142	700.0	97.656±16.324	332.524±58.340	151.583±39.211	666.507±65.815	**870.521±46.518**
h1hand-pole	19.721	700.0	150.233±21.203	551.138±104.690	439.295±28.894	**887.945±25.582**	**879.091±9.683**
h1hand-hurdle	2.371	700.0	42.457±6.239	110.333±11.177	55.887±9.599	**471.050±38.269**	465.969±140.063
h1hand-maze	106.233	1200.0	142.249±11.927	362.677±4.216	348.915±5.388	451.575±12.825	**479.622±72.949**
h1hand-sit_simple	10.768	750.0	675.884±196.728	**935.483±3.844**	**901.682±25.333**	**872.353±12.648**	**814.452±51.049**
h1hand-sit_hard	2.477	750.0	148.216±36.804	**885.141±33.683**	710.754±258.261	358.350±10.554	426.718±30.742
h1hand-balance_simple	10.170	800.0	27.412±3.310	101.305±8.963	106.005±6.567	103.400±12.201	**310.167±17.553**
h1hand-balance_hard	10.032	800.0	33.042±6.165	72.918±2.266	**77.788±5.622**	74.962±4.581	70.913±8.696
h1hand-reach	-50.024	12000.0	3610.454±617.640	4317.026±1527.201	4931.460±597.736	**5820.811±501.354**	5114.852±40.931

Table 3: Refinement-matched comparison on DMC humanoid tasks (mean ± sample standard deviation over three seeds).

Task	Gaussian + Refine	EZ-IGP
humanoid-stand	763.8 ± 18.7	$\mathbf{903.7 \pm 7.7}$
humanoid-walk	761.6 ± 35.5	$\mathbf{911.5 \pm 7.0}$
humanoid-run	230.5 ± 37.5	$\mathbf{369.5 \pm 5.6}$

Table 4: **Preliminary HB-Hard sensitivity study.** Benchmark normalized scores (mean ± sample standard deviation over three seeds), evaluated near 300k training updates across 14 tasks. Candidates/simulations are fixed at 8/16 per round; total search computation grows with R.

Bins V	Rounds R	Normalized score
11	2	0.622 ± 0.017
21	2	0.663 ± 0.026
31	2	0.743 ± 0.041
41	2	0.690 ± 0.013
31	1	0.667 ± 0.037
31	4	0.689 ± 0.047

Table 5: DMC humanoid performance under strict search budgets (mean ± sample standard deviation).

Candidates / simulations	Stand	Walk	Run	Average
$2/4$	375.193 ± 10.490	406.443 ± 37.224	108.607 ± 0.046	296.748 ± 15.889
$4/8$	772.111 ± 15.872	790.306 ± 47.157	221.021 ± 31.538	594.479 ± 31.523
$8/16$	903.719 ± 7.734	911.542 ± 7.005	369.468 ± 5.633	728.243 ± 1.635

C Training Curves

This section presents the full training dynamics of EZ-IGP and the baselines on the two benchmark suites used in our evaluation: the high-dimensional subset of DMControl (DMC-Hard) and HumanoidBench-Hard. While the per-task scores reported in Tables 1 and 2 summarize *peak* performance, they do not directly reflect how each method behaves during the course of training. The curves below complement those tables by exposing two additional dimensions of method quality: *learning speed*, i.e., how quickly returns rise as a function of environment interactions, and *learning stability*, i.e., how consistently performance is maintained across seeds and across the training budget.

In all figures, the y-axis denotes the unnormalized episodic return and the x-axis denotes environment steps in millions. Each curve is aggregated over 5 random seeds; solid lines correspond to the seed mean and shaded regions to 95% bootstrap confidence intervals. We report *unnormalized* returns here so that readers can directly compare to the raw scores used by prior work and to the per-task random/success thresholds listed in Table 2.

Fig. 6 reports the curves on the 7-task DMC-Hard suite, comprising 4 dog tasks (*dog-stand*, *dog-walk*, *dog-run*, *dog-trot*) and 3 humanoid tasks (*humanoid-stand*, *humanoid-walk*, *humanoid-run*). These two embodiment families differ markedly in morphology and physical modeling, so their joint inclusion lets us assess whether the gains from EZ-IGP transfer across embodiments rather than being tied to a specific robot. Reading the curves alongside Table 1 also clarifies an effect that peak-score tables can hide: on the more dynamic tasks such as *dog-run* and *dog-trot*, EZ-IGP's advantage opens up early in training and is preserved throughout, indicating better sample efficiency rather than only a higher asymptote.

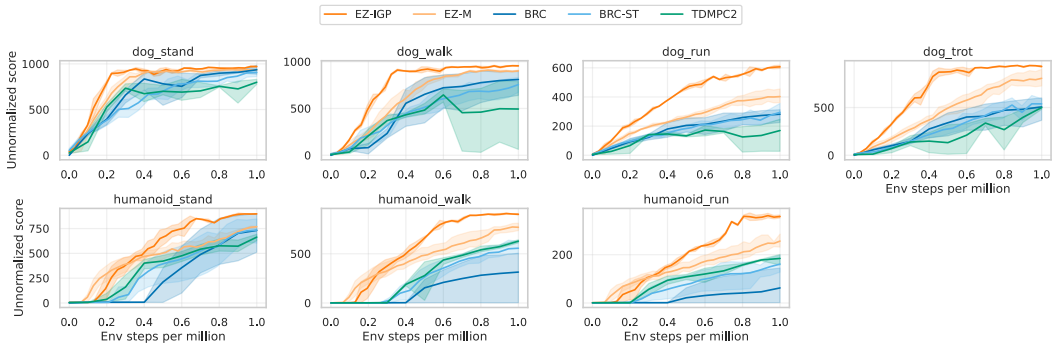


Figure 6: **Unnormalized curves on DMControl Hard tasks.** The y-axis denotes episodic return, and the x-axis denotes environment steps (millions). Curves are aggregated over five random seeds, and shaded regions denote 95% confidence intervals. Per-task scores are listed in Table 1.

Fig. 7 reports the analogous curves on the 14-task HumanoidBench-Hard suite (H1hand-* with hand control). Compared with DMC-Hard, this setting introduces additional degrees of freedom from finger actuation, more demanding manipulation-style tasks (*sit_simple*, *sit_hard*, *balance_simple*, *balance_hard*, *reach*), and substantially higher-dimensional observation and action spaces. Beyond confirming the per-task scores in Table 2, the curves are useful for diagnosing where the gap between EZ-IGP and the strongest baselines opens up during training, and conversely for identifying tasks—such as *h1hand-stair* and *h1hand-sit_hard*—where the gap is smaller.

D Score Normalization

Why normalize. The benchmarks we consider span tasks whose raw return scales differ substantially—for example, `h1hand-reach` has a success score of 1.2×10^4 while `h1hand-walk` has a success score of 700 (Table 2). An unweighted average of raw returns across such tasks would be dominated by a handful of large-scale tasks and would obscure relative progress on the rest. Following standard practice in the multi-task RL literature, we therefore normalize each task to a comparable scale before aggregating across tasks.



Figure 7: **Unnormalized curves on the HumanoidBench-Hard benchmark.** The y-axis denotes episodic return, and the x-axis denotes environment steps (millions). Curves are aggregated over five random seeds, and shaded regions denote 95% confidence intervals. Per-task random and success thresholds are listed in Table 2.

HumanoidBench normalization. For each HumanoidBench task we map raw episode return to a normalized score using

$$\text{Return}_{\text{norm}} = \frac{\text{Return} - \text{Random Score}}{\text{Success Score} - \text{Random Score}}, \tag{20}$$

where the *Random Score* is the mean episode return obtained by a uniform-random policy on that task and the *Success Score* is the task-specific completion threshold provided by the benchmark authors. Under this definition, a normalized score of 0 corresponds to the random baseline and a score of 1 corresponds to nominal task success; values above 1 are possible and indicate that the agent exceeds the success threshold. The per-task random and success scores are listed in the first two numerical columns of Table 2.

DMControl normalization. DMControl returns are by construction bounded in $[0, 1000]$ on each task, so we adopt the simpler convention of dividing the cumulative reward by 1000 to obtain a normalized score in $[0, 1]$. This avoids introducing a benchmark-specific random-baseline subtraction that is not part of standard DMControl protocol while keeping DMC and HumanoidBench scores on a comparable scale for cross-benchmark comparisons.

Bolding convention and aggregation. In Table 1 we **bold** the best result per task. In Table 2 we additionally bold any score that surpasses the per-task success threshold, since multiple methods may simultaneously satisfy this criterion; when no method reaches success, we bold only the SoTA. For both benchmarks and every method, we take the maximum evaluation return within each seed and task, then report the mean and sample standard deviation across seeds. Benchmark aggregates normalize these per-seed task maxima, average uniformly across tasks within each seed, and finally average across seeds.

E Hyperparameters and Details

This section documents the implementation details of EZ-IGP needed to reproduce our results. All experiments are conducted on a single machine with 4 NVIDIA A100 GPUs. EZ-IGP is built on top of EZ-M and inherits its multi-task training setup, network architecture, and learning objectives; the only methodological change is replacing the Gaussian policy head with the per-dimension factorized categorical policy and the iterative Gumbel planning procedure. The hyperparameters reported here therefore split naturally into two groups: those inherited from EZ-M (optimization, world-model unrolling, replay) and those specific to IGP (bin count, Gumbel candidate count, refinement rounds).

Computational efficiency. Our original within-implementation comparison measures 0.368 s per decision for EZ-IGP (2 rounds, 16 simulations and 8 candidates per round), versus 0.375 s for EZ-M (1 round, 32 simulations and 16 candidates). These decision-time measurements indicate comparable planning overhead; they do not by themselves establish equal training cost. We separately report the broader baseline comparison below, using a common V100 for decision latency rather than combining timings from different measurement setups.

Baseline decision-time and training-time comparison. Table 6 reports a synchronized, warm-started microbenchmark on one NVIDIA V100-SXM2-32GB GPU using HB-Hard observation/action dimensions. Latency includes complete action selection for a single observation, not just a network forward pass. Training times refer to HB-Hard at 1M environment steps per task; BRC timings are read at that budget, rather than at the end of its longer 2M runs. The released TD-MPC2 and BRC-ST curves lack wall-clock timestamps, so we leave these entries unavailable.

Table 6: **Baseline compute comparison on HB-Hard.** Decision latency uses the same V100 GPU; training times use the recorded hardware and are not hardware-normalized. BRC-ST latency was not separately measured. Training-time uncertainties are across seeds.

Method	Decision procedure	Latency/action (ms)	Training time at 1M	Training setup
BRC	Actor only	0.162 ± 0.015	40.58 ± 0.03 h (3 seeds)	1 RTX 3090/run
BRC-ST	Actor only, no task embedding	Not measured	Not timestamped	Separate run/task
TD-MPC2	Full MPPI planning	47.10 ± 0.49	Not timestamped	Official curves
EZ-IGP	Prediction + two-round Gumbel search	247.23 ± 2.63	24.59 ± 0.93 h (2 seeds)	Distributed A100 setup

BRC is fastest because model-free action selection only evaluates the actor; BRC-ST has the same actor-only structure. TD-MPC2 batches candidate trajectories on the GPU, whereas successive tree expansions depend on preceding search results. Consequently, lower operation counts need not imply lower latency. For context, EfficientZero-v2 [6] reports 9,216 latent predictions and 3.6×10^{10} FLOPs for TD-MPC2, versus 32 expansions and 4.7×10^{7} FLOPs for Gumbel search. EZ-IGP also uses $RN = 2 \times 16 = 32$ expansions in this comparison; those published FLOP counts are not new measurements of EZ-IGP.

At the matched interaction budget, EZ-IGP has shorter recorded training wall-clock time than BRC. Its Ray implementation overlaps data collection, reanalysis, and optimization. Because the hardware differs, this is a practical turnaround comparison, not evidence of lower hardware-normalized training compute. BRC-ST additionally requires a separate training run for each task.

Configuration philosophy. Unless otherwise specified, we keep the same hyperparameter configuration across *all* tasks within a benchmark. This is a deliberate choice. A central claim of this paper is that the projection-error issue we identify is structural rather than tuning-dependent, so per-task hyperparameter searches would muddy that claim by allowing implicit task-specific adaptation. The only configuration that varies between settings is the number of training steps, which we scale with the size and difficulty of the task set: 2×10^5 for DMC-Hard (7 tasks) and 3×10^5 for HumanoidBench-Hard (14 tasks, with finger actuation and a higher-dimensional action space).

World-model and optimization settings. We use Adam with a learning rate of 1×10^{-4} and weight decay of 2×10^{-5}, which we found to be stable across all task combinations we tried; we did not perform a per-task learning-rate sweep. The discount factor $\gamma = 0.99$ and batch size of 512 are standard for continuous-control RL at this scale. We use a relatively short unroll of 5 steps and a matching TD horizon of 5, matching the choice in EfficientZeroV2 and shorter than what fully

Table 7: Key hyperparameters used in HumanoidBench experiments

Hyperparameter	**Value**
Discount factor γ	0.99
Unroll steps	5
TD steps	5
Batch size	512
Replay buffer size	1×10^6
Training steps	2×10^5 (DMC) / 3×10^5 (HumanoidBench)
Network hidden dimension	512
Task embedding dimension	128
Task embedding usage	Rep / Dyn / Rew / Policy-Value
Number of res blocks	3
MCTS simulations	16
Top actions	8
Sampled actions	8
num bins	31
num refined rounds	2
Optimizer	Adam
Learning rate	1×10^{-4}
Weight decay	2×10^{-5}

model-based methods such as DreamerV3 use; this reflects our reliance on MCTS to extend planning depth at decision time rather than at training time.

IGP-specific hyperparameters. Three hyperparameters are introduced by IGP. The number of bins per dimension, $V = 31$, controls the discretization resolution and trades off representation precision against the per-dimension softmax size. A finer grid yields a tighter discretization-error bound but enlarges the categorical support; our ablations show that returns improve from $V = 11$ up to $V = 31$ and then plateau, motivating $V = 31$ as a default. The number of Gumbel Top-K candidates per planning call is $K = 8$ (*Sampled actions*); together with 8 *Top actions* for sequential halving and 16 MCTS simulations per inner search, this fixes the per-round search budget. We use two refinement rounds as a practical performance–compute trade-off. One round disables iterative refinement, while four rounds provide only a small, higher-variance gain in the broader HB-Hard sweep and double the search expansions; the preferred setting is therefore not universal.

Task conditioning and shared backbone. We retain EZ-M's task-conditioning mechanism unchanged: a 128-dimensional task embedding is injected into all four learned components of the world model—representation, dynamics, reward, and policy-value. Injection at all four sites keeps the embedding small while ensuring that task-specific physics (in dynamics and reward) and task-specific behavior (in representation and policy-value) can both be expressed. The shared backbone uses a hidden dimension of 512 and 3 residual blocks, also inherited from EZ-M.

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: Abstract and Section 1 state our three contributions: identifying the projection error problem (formalized in Section 4.1), proposing IGP (Section 4.2), and demonstrating state-of-the-art aggregate performance on DMControl and HumanoidBench (Section 5.2).

   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: Limitations are discussed in Section 6, including fixed uniform discretization, the per-dimension factorized assumption, dependence on search hyperparameters (bin count, candidate number, refinement rounds), and the lack of real-robot validation.

   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: Proposition 1, Proposition 2, Corollary 1, and Lemma 1 are stated in Section 4 with full assumptions; complete proofs are provided in Appendix A.1 and Appendix A.2.

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: Algorithmic details are described in Section 4.2; experimental setup in Section 5.1; full hyperparameters in Appendix E, Table 7. EZ-IGP is built on the open-source EfficientZero-Multitask codebase, with only the policy head replaced as described in Section 4.2.

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 publicly available at https://github.com/liushaohuai5/EZ-IGP, including dependency specifications, benchmark configurations, training launch commands, and periodic evaluation.

   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: All training and evaluation details, including optimizer, batch size, learning rate, network architecture, and IGP-specific hyperparameters (bin count, candidate count, refinement rounds), are documented in Appendix E, Table 7.

   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: All learning curves in Figures 6 and 7 are aggregated over five random seeds, with solid lines denoting seed means and shaded regions denoting 95% bootstrap confidence intervals. Aggregation conventions for tables are described in Appendix D.

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: Compute resources are reported in Appendix E: all experiments are run on a single machine with 4 NVIDIA A100 GPUs.

   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 confirm that our work conforms with it. The research is purely methodological reinforcement learning, evaluated only on simulated benchmarks.

   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: [Yes]

Justification: Our method is foundational continuous-control RL research evaluated only in simulation. Potential downstream impacts (e.g., real-world robotic deployment) are shared with the broader continuous-control RL community and are not specific to our contribution.

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: [N/A]

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 use DMControl [10] and HumanoidBench [11], both released for research use, and reproduce baselines from their official open-source implementations with appropriate citations.

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: The paper introduces a new method, but it does not currently release a new dataset, codebase, or model artifact alongside the submission.

    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: [N/A]

    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: [N/A]

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: [N/A]

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.