ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

Task-Consistent Bayesian Domain Inference via Performance Distributions for Deep Reinforcement Learning Policy Deployment

Xiang Fu[1] *and Kewei Chen*[1,*]

[1] Faculty of Mechanical Engineering & Mechanics, Ningbo University, Ningbo, 315211, China

SUMMARY: *Although Deep reinforcement learning (DRL) has achieved remarkable success in robotics, the policies learnt in simulation often experience severe performance drops in the real world because of the reality gap. To reduce this performance gap, we propose a Task-Consistent Bayesian Inference (TCBI) framework for sim-to-real transfer. Rather than relying on intractable dynamics likelihoods or matching on high-dimensional trajectories, TCBI builds the task-level pseudo-likelihood based on the divergence of simulated and real performance distribution. In our formulation, the reward statistics, the body posture distributions and the contact time ratios are all compact task-oriented performance statistics of the distribution that characterizes task-specific domain discrepancy. Our design thus supports likelihood-free Bayesian inference effectively and robustly and also improves computational efficiency. We demonstrate the proposed framework on a six-legged robot in both balance task and forward locomotion task. Our experimental results show that TCBI always lowers the reward distribution disparity and also achieves better real-world performance than domain randomization, ABC, and simulation optimization (SimOpt) do. Ablation studies further show that incorporating reward, posture, and contact statistics can further improve the posterior identifiability and policy stability compared with using the reward distributions. Moreover, posterior variance analysis tells us that parameter concentrations in the inference process are progressive, and wall-clock time comparison also demonstrates that the computational cost of the method is much lower than that of trajectory-based methods. Robustness experiments under sensor noises further verify the stability and generalization capability of the method that we presented. All of these experimental results clearly point out that task-level probabilistic inference gives us an efficient, robust and scalable solution for sim-to-real deployment of reinforcement learning methods.*

KEYWORDS: *Deep Reinforcement Learning; Sim-to-Real Transfer; Domain Adaptation; Bayesian Inference; Simulation Optimization*

1 Introduction

Deep reinforcement learning (DRL) has had success for robotics. Deep RL can enable agents to learn complex control policies from high-dimensional observations. DRL using large simulation environments enables efficient policy learning. This includes simulation for locomotion, manipulation, and navigation. However, policies trained in simulation can suffer a severe performance gap when applied in the real world. This is because of the well-known reality gap. The gap comes from the dynamics, sensing and environment difference.

*chenkewei@nbu.edu.cn
https://doi.org/10.65102/is20261261
Published: 10 June, 2026

1.1 Sim2Real Transfer Methods: From Randomization to Inference

Closing this gap has led to a plethora of sim-to-real (Sim2Real) transfer methods [1-9]. Early methods are dominated by domain randomization (DR). DR improves robustness by sampling simulation parameters from set ranges. Parameters could be mass, friction or actuator dynamics. Although DR is simple and effective, it often relies on manually set parameter distributions. This results in either a poor coverage or extremely conservative policies. To solve this disadvantage, adaptive domain randomization and simulation optimization methods (SimOpt, for example) are proposed. They update the parameter distribution iteratively according to the difference between simulated and real observations [10]. Such methods improve sample efficiency and adaptability. But they often treat the problem as a deterministic optimization problem. They don't have principled quantification of uncertainty.

These approaches explicitly model uncertainty. Bayesian domain inference methods have been introduced for this purpose. These methods infer a posterior distribution over simulation parameters. The distribution is conditioned on real-world observations. In principle, these methods have to evaluate the likelihood of the observations given candidate parameters. This is infeasible for complex robots. To address this problem, likelihood-free inference methods have been proposed. Likelihood free inference methods are, for instance, approximate Bayesian computation (ABC), simulation-based inference. Likelihood evaluation is replaced by distance evaluation. Those are comparison distances between simulated and observed trajectories [11-13]. These methods are theoretically effective but have disadvantages:

1. They rely on high-dimensional distributions over trajectory or state-action distributions.
2. They have to implement nice distance metrics.
3. They are sensitive to distribution drifts induced by the policy.

A unifying insight across all these methods is quite clear. Most existing Sim2Real methods do trajectory-level or state-distribution matching. This matching is the main consistency metric. Such metrics need to minimise the discrepancy in dynamics. Such metrics generally do not distinguish task-relevant performance difference. This is particularly true for multiple trajectories that give similar task results. Moreover, trajectory-based metrics poorly scale with dimensionality. They are also sensitive to stochasticity and policy variations. In contrast, new recent advances are moving beyond explicit dynamics matching. They address task-oriented representations for Sim2Real transfer. A growing line of work suggests successful transfer does not have to make explicit trajectory-level matching. All one needs to preserve is task-relevant information in a more compact representation.

Of these studies on recent work on abstract Sim2Real via Approximate Information States is particularly formal. Policies can be learned in simple or abstract simulators. That is possible if the representation still has enough task-relevant information [14]. By viewing the state abstraction as information compression, these approaches prove full state matching or full trajectory matching need not be performed. Keeping a low-dimensional representation is enough. This representation consists of the information needed for decisions, but abstraction inevitably leads to partial observability. It requires history-dependent correction and careful grounding of ground information from the real world. Complementary to such viewpoint, recent Sim2Real frameworks propose more reward-driven or task specification-driven representations. Trajectory matching becomes not explicit. For example, Iterative Keypoint Reward (IKER) directly constructs task specifications. It uses spatial relations between keypoints. IKER also iteratively optimizes the reward function with real-world feedback [15].

It implicitly encodes task-relevant structure. It does not compare full state-action trajectories. Similarly, Sim2Real locomotion experiments clearly show carefully designed reward functions matter. Reward functions matter more than exactly matching dynamics. These

reward functions directly shape learned behaviors and adaptation abilities [16]. Conventional trajectory-based methods (e.g., ABC, simulation-based inference) rely on matching between high-dimensional states or trajectories. This matching is computationally costly. It is also sensitive to stochasticity. Moreover, it conflicts with task objectives.

1.2 Task-Oriented Representations and Approximate Information States

Such limitations are reflected in recent research, which has moved to task representations. Many modern RL control methods make use of criteria of performance. Such criteria are return distributions, entropy regularized objectives, and task-level reward signals. They suggest learning and evaluation. The task-level representations are a better choice. They are low-dimensional, task-level and robust. Examples include reward distributions, keypoint objectives and abstract information states. We can consider such representations to be Approximate Information States. They are summaries of trajectories that retain important information. This information is needed for optimality. Information-theoretically, these task-level quantities can be considered to be Approximate Information States. They compress a trajectory from high dimensions into lower dimensional statistics. These statistics retain information relevant to the task. Compared to measures of distance between trajectories, such representations have the following advantages:

- Low dimensionality: reward or return distributions have much lower dimensionality than the full trajectories.
- Task relevance: they directly reflect the control objectives and performance goals.
- Robustness: they are less sensitive to stochastic changes and distribution shifts introduced by policies.
- Computational simplicity: dissimilarities between scalar or low-dimensional distributions are easy to calculate.

Nevertheless, the benefits discussed above exist despite the fact that existing efforts to resolve these issues have two key drawbacks:

(1) They do not capture a principled probabilistic framework, such as reward engineering methods.

(2) They do not link the task-level measures to the inference of domain parameters, like zero-shot RL and abstraction-based methods.

1.3 Proposed Method and Contributions

Here, we propose a new Task-Consistent Bayesian Inference (TCBI) approach. We move the emphasis from consistency in dynamics level to consistency in task level. Instead of matching trajectories or state distributions, we formulate a pseudo-likelihood function. Our pseudo-likelihood function is based on divergence of the reward distributions from simulation to the real environment. Reward signals represent the task performance. Meanwhile, reward signals depend implicitly on the system dynamics. Therefore, alignment of reward distributions serves as a compact and informative proxy to domain consistency. Moreover, the proposed method provides a principled mapping of the performance metric to Bayesian inference. In this paper, we treat the divergence in distribution (e.g., Jensen–Shannon divergence) as the surrogate likelihood. It allows efficient posterior inference of domain parameters, avoiding explicit modelling of the domain dynamics. Compared with the trajectory-based likelihood-free approaches, our method significantly reduces the computational complexity, and also enhances robustness to variation in policy.

Indeed, to validate our proposed framework, we run experiments on the hexapod robot platform. Our experiments range from simulation and real deployment. We iteratively adjust

domain parameters via task-consistent pseudo-likelihood. The learned policies have better robustness and transfer performance. Compared with commonly-used domain randomization and likelihood-free inference methods, they outperform others. The main contributions of this paper include the following:

1. proposed a task-consistent Bayesian inference framework;
2. replaced trajectory-based likelihood with the divergence of reward distribution;
3. proposed pseudo-likelihood formulation, which connects the task-level performance distribution with the probabilistic estimation of domain parameter;
4. made systematic comparisons between two types of trajectory-level metrics and task-level (Approximate Information State) metrics. We do this through real-world experiments on a physical hexapod robot. The results show improved transfer performance and robustness. And comparison reflects the advantages of trajectory-based metric in Sim2Real transfer; validated the effectiveness of our method through real-world experiments, using a physical hexapod robot platform. They do achieve better transfer performance and robustness.

2 Materials and Methods

2.1 Value Function and Sim-to-Real Mismatch

Reinforcement learning, the action-value function $Q(s,a)$ has a definite meaning. It represents the expected cumulative reward. This reward comes from performing action a in state s using some policy. The value function depends on the environment dynamics. The environment dynamics is characterized by the transition probability:

$$p(s'|s,a,\xi) \tag{1}$$

where ξ represents the domain parameters of an environment, such as mass, friction, actuator characteristics, etc. In sim-to-real transfer, the gap between simulation and real environment induces the gap in transition dynamics:

$$p(s'|s,a,\xi_{sim}) \neq p(s'|s,a,\xi_{real}) \tag{2}$$

These gaps cause the change in state distribution. They also affect the reward function $r_\xi(s,a)$. Concretely, the action-value function learned in simulation will be mismatched with the true value function in the real environment:

$$\hat{Q}(s,a) = Q(s,a) + \epsilon(s,a) \tag{3}$$

where $\epsilon(s,a)$ represents the estimation error induced by domain mismatch.

2.2 Trajectory-Based Domain Inference and Limitations

Recent studies have looked at distribution matching methods. These methods compare behaviours simulated and real behaviours. They are used particularly in imitation learning and policy evaluation [17, 18]. These methods aim at making simulation state-action or occupancy distributions resemble the real behaviour. Such alignment enhances domain transfer performance. From state abstraction theory viewpoint, the notion of a trajectory or state-action distributions is not minimal sufficient statistic for control. Abstraction brings partial observability. Under this setting, multiple possibly different underlying dynamics can yield equivalent trajectory distributions. This many to one situation gives rise to an identifiability

problem. Matching trajectories does not guarantee matching task-specific dynamics. In contrast, task-level performance metrics behave differently. Such metrics as r reward distributions represent compact task-oriented performance measures for the task-level policy evaluation. This holds for a given task.

More recent AIS studies implied that effective control doesn't always require explicit preservation of the full trajectory information, but compact representations of task characteristics that preserve control-relevant information are often all that is needed for decision making. From this point of view, reward distributions can be regarded as performance-sensitive summaries of the evolution of the interaction. They explicitly aggregate the impact of the transition dynamics, state visits and policy behaviour to one low-dimensional representation related to long-term control performance. Unlike raw trajectory distributions, reward distributions suppress many variations in the trajectories that have limited impact on task success. As a result, they provide a more compact and task-oriented description of system behaviour.

Following the intuition of AIS-based abstraction methods, we regard reward distributions as compressed control-oriented representations induced by interaction history. Reward distributions are not strict minimal sufficient statistics in the formal statistical sense. They are only practical task-oriented indicators of the information that needs to be preserved for policy evaluation and control tasks. This view has several desirable properties:

(i) task relevance: reward distributions directly account for the expected performance of control and long-term reward.

(ii) dynamics dependence: parameters of dynamics that significantly change the way in which a task is executed change the reward distribution too.

(iii) robustness: Compared with trajectory distribution, reward distributions are less affected by local stochastic behaviours and/or trajectory noise.

Importantly, the reward function is a task-oriented projection of the underlying dynamics. Different physical parameters change the transition dynamics and state visitation, which are eventually reflected by the reward statistics obtained upon execution of a policy. Conversely, trajectory distributions are heavily mixed by both task-relevant and task-irrelevant variations. A small trajectory deviation should not always be the result of a performance difference. Reward distributions rather propagate behaviour components that are crucial to control goals and long horizon returns. Therefore, a match in reward distributions for simulation and real systems provides a task-consistent criterion for domain inference. This motivates us to regard sim-to-real domain adaptation as a problem of matching task-level distributions and is the basis for our proposed Task-Consistent Bayesian Inference Framework.

Given a prior distribution $p(\xi)$, the posterior is:

$$p(\xi|\mathcal{D}_{real}) \propto p(\mathcal{D}_{real}|\xi)p(\xi) \tag{4}$$

where:

- $\mathcal{D}_{real}$ is data collected from the real system
- $p(\mathcal{D}|\xi)$ is a domain likelihood
- $p(\xi)$ is a prior distribution

Bayesian methods provide a principled approach to estimating domain parameters. They estimate the posterior distribution for a parameter conditioned on data observed [19]. For sim-to-real transfer, this means estimating simulation parameters. The simulation parameters that best fit the data observed in the real world.

Computing the likelihood of the observed data is often intractable. This is because the robotic dynamics are complex (high dimensional and nonlinear contact interactions).

To overcome this challenge, likelihood-free inference methods are proposed. Approximate Bayesian Computation (ABC) and simulation-based inference (SBI) are examples [20-22]. The methods replace explicit likelihood evaluations. Instead, they compare simulated and real data. These comparisons rely on summary statistics and distance functions. In robotics, BayesSim has domain inference methods that match trajectory level statistics between the simulated and real environments. Doing so enables estimation of posterior without explicit likelihood functions.

$$p(\tau^{real}|\xi) \approx \exp(-d(\tau^{real}, \tau^{sim}(\xi))) \tag{5}$$

where:

- d($\cdot,\cdot$) is a distance function(e.g., L2, MMD, KL, JS)
- $\tau^{sim}(\xi)$ is a simulated trajectory generated under the parameter ξ

SimOpt performs sim-to-real transfer as a simulation parameter inference problem. It minimizes the deviation between real and simulated trajectories, i.e. the domain parameter distribution $p(\xi)$ is optimized. This optimization minimizes the divergence $D(\tau_{sim}, \tau_{real}(\xi)))$.

The optimization adopts the Relative Entropy Policy Search (REPS). The updating constraints with REPS are bounded in KL-divergence, so that the learning is always stable.

This method makes system identification a reinforcement learning problem. This problem is based on simulation parameters.

$$p^*(\xi) = \underset{p(\xi)}{argmin}\, D(\tau_{\xi\sim p(\xi)}, \tau_{real}) \tag{6}$$

The simulation parameter distribution is updated through a REPS-style constrained optimization. Given samplesξ_k , and their sample costs $c(\xi_k)$, we obtain importance weights:

$$w_k \propto \exp(-\frac{c(\xi_k)}{\eta}) \tag{7}$$

η is the estimate of solved dual problem under KL constraint, and the updated Gaussian distribution is obtained by weighted maximum likelihood method. Divergence calculation formula is shown in equation 8.

$$D\left(\tau_{\xi}^{ob}, \tau_{real}^{ob}\right) = w_{\ell 1} \sum_{i=0}^{T} \left|W\left(o_{i,\xi} - o_{i,real}\right)\right| + w_{\ell 2} \sum_{i=0}^{T} \left\|W\left(o_{i,\xi} - o_{i,real}\right)\right\|_2^2 \tag{8}$$

where:

- τ_{real}^{ob} is a trajectory observed in real world.
- τ_{ξ}^{ob} is a simulated trajectory observed under the parameter ξ.
- $w_{\ell 1}$ and $w_{\ell 2}$ are the weights of One-norm and two-norm.
- W is the weight of each observation dimension.

These approaches implicitly view trajectory distribution as sufficient statistic for the domain. But with abstraction and partial observability, trajectories cannot form minimal sufficient statistic, and thus the parameter estimation is ambiguous. However, these approaches all have high-dimensional trajectory or state-action representation hidden, they are difficult to calculate, sensitive to policy, and may contain redundant information irrelevant to task performance, etc.

2.3 Reward-Distribution-Based Inference and Algorithm

We then aim to infer domain parameters ξ. The aim is to make the simulated environment

consistent with the real world system. Instead of constructing likelihoods in the high-dimensional space of trajectories, we build a pseudo-likelihood based on reward distributions. Reward distributions are a task-level performance representation. This changes the aim of inference. It changes from behavior matching to task-consistent inference. This perspective is compatible with theory of Approximate Information States (AIS). In AIS theory, optimal decision should be made based on compressed representations. The representations compress things down to what is relevant to the reward. Our formulation in this paper makes reward distribution play the role of such compressed representations. Reward distributions can be regarded as a practical task-level representation inspired by minimal sufficient statistics for sim-to-real transfer. It would be unproductive to rely on the intractable dynamics likelihood $p(\mathcal{D}|\xi)$ and only consider task-level pseudo-likelihood aligned with reward distribution.

Let $\mathcal{D}_{real}$ be data from the real system, and $\mathcal{D}_{sim}(\xi)$ be simulated rollouts under domain parameters ξ. Let us next introduce the reward statistics distributions:

$$P_r^{real} = p(r|\mathcal{D}_{real}), P_r^{sim}(\xi) = p(r|\mathcal{D}_{sim}(\xi)) \tag{9}$$

Now, we replace intractable likelihood by divergence-based surrogate:

$$p(\mathcal{D}_{real}|\xi) \approx \exp(-\alpha \cdot D(P_r^{real}, P_r^{sim}(\xi))) \tag{10}$$

where: $D(\cdot,\cdot)$ i is a distribution divergence (for example, Jensen–Shannon divergence), $\alpha > 0$ is a temperature parameter.

This is fundamentally different from ABC-based likelihoods. These likelihoods use raw or handcrafted statistics summaries. Here, the statistics are task induced and policy dependent. This guarantee inferred posterior overlap with task performance.

Our approach defines likelihood on the reward distribution differences instead of a true generative model. That is, $p(\mathcal{D}_{real}|\xi)$ is a relative compatibility between different ξ. Temperature parameter α controls the sensitivity of the measure.

Finally, we do not optimize α explicitly. Instead, we implicitly adjust α by distance scaling and acceptance thresholds. The two aspects implicitly affect the ‘effective sharpness’ of the posterior distribution.

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}}\, Var(w_\alpha) \tag{11}$$

Equation 10 affords greater likelihoods for domain parameters. For domain parameters, simulated reward distributions are close to the real ones. This formulation is fundamentally different from BayesSim, SimOpt and standard ABC. The latter methods approximate likelihoods as trajectory-level divergences. Ours is different. We define an energy-like likelihood over task-level representations of performance. As a consequence, the inferred posterior is shaped directly by task-level discrepancies. It is not shaped by raw observation discrepancies.

And can thus be viewed as Bayesian inference in a reduced task-oriented representation space defined by reward statistics. Compared with trajectory based inference, this representation creates a more concentrated posterior. It is also more aligned with the task. Trajectory based inference works in an over-parameterised observation space. In this view, the divergence induces an energy function over domain parameters. This enables posterior inference. We do not need to actually evaluate the true likelihood. Introducing task variation enriches the space of task-conditioned observations and diversifies behaviours inferred. It moves from dynamics-only representation to dynamics–task space. This enriches the task-

conditioned diversity of observations. More fundamentally, task variations induce diverse state visitation distributions and make reward distributions more sensitive to underlying dynamics parameters.

Moreover, reward distributions implicitly marginalize over trajectory variation. This trajectory variation stems from stochastic policy and environment noise. Such marginalization effect lowers the variance of likelihood estimation. It also renders better robustness against the trajectory-based distances. The distances are susceptible to alignment errors and noise in high dimensions. As a result, the divergence $D(P_r^{real}, P_r^{sim}(\xi))$ becomes more discriminative. This alleviates the practicable ambiguity in the identification of task-based inference under limited task conditions. For limited task conditions, different dynamics parameters can lead to similar reward distributions. More importantly, the task variation increases the excitation of system dynamics. This guarantees that different domain parameters tend to induce distinguishable task-level performance distributions under sufficient task-variation conditions. This eases the identifiability issue in the trajectory matching. In the absence of excitation, trajectories become indistinguishable under different parameters.

Under limited task conditions, multiple dynamics parameters would produce indistinguishable trajectories. Therefore, inference from trajectory space is non-identifiable. However, one can tell these dynamics parameters apart if one compares their corresponding induced reward distributions over different tasks. These are distinguishable in task-based performance representation. Therefore, aligning reward distribution regularizes the inverse problem well. It improves the posterior concentration. Inserting pseudo-likelihood into Equation 4:

$$p(\mathcal{D}_{real}|\xi) \propto \exp(-\alpha \cdot D(P_r^{real}, P_r^{sim}(\xi)))p(\xi) \tag{12}$$

This leads to a generalized likelihood-free Bayesian inference scheme controlled by task consistency. Strict Bayesian consistency cannot be achieved. In our work, the use of summary statistics and approximate inference are employed. The posterior concentrates empirically. It converges into stable posterior regions. We choose the Jensen–Shannon (JS) divergence. It is symmetric and bounded:

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M), M = \frac{1}{2}(P+Q) \tag{13}$$

Compared with the trajectory-based distance such as L2 and MMD, the JS divergence over reward distribution directly measures the statistics of task-based performance. It leads to a discrepancy measure more stable, easily interpretable and task consistent.

Algorithm 1: Task-Consistent Bayesian Inference (TCBI)

Input: Prior distribution $p(\xi)$
Real-world reward data D_{real}
Number of particles N
Number of max generations T
Temperature parameter α
Perturbation scale σ
Output: Approximate posterior $p(\xi|\mathcal{D}_{real})$
1： Compute real reward distribution: $P_r^{real} = \text{EstimateDistribution}(D_{real})$
2： Initialization: Sample particles: $\{\xi_i^{(0)}\}_{i=1}^N \sim p(\xi), w_i^{(0)} = \frac{1}{N}$
3： For generation t=1,…,T:

4: For each particle i=1,…,N:

5: (a) Resample & perturb: $\xi_i^{(t)} \sim \sum_j w_j^{(t-1)} \mathcal{N}(\xi_i^{(t-1)}, \sigma^2 \mathrm{I})$

6: (b) Simulation rollout: $D_{sim}^{(i)} \sim \mathrm{Simulate}(\xi_i^{(t)})$

7: (c) Estimate reward distribution: $P_r^{sim(i)} = \mathrm{EstimateDistribution}(P_{sim}^{(i)})$

8: (d) Compute discrepancy (JS divergence): $D_i = D_{JS}(P_r^{real} || P_r^{sim(i)})$

9: (e) Compute importance weight: $w_i^{(t)} = \exp(-\alpha D_i)$

10: Normalize weights: $w_i^{(t)} \leftarrow \frac{w_i^{(t)}}{\sum_j w_j^{(t)}}$

11: Compute effective sample size (ESS): $ESS = \frac{1}{\sum_i (w_i^{(t)})^2}$

12: if ESS<τN: Resample particles according to $w_i^{(t)}$, Reset weights: $w_i^{(t)} = \frac{1}{N}$

13: **return** posterior approximation: $p(\xi|\mathcal{D}_{real}) \approx \sum_{i=1}^{N} w_i^{(t)} \delta(\xi - \xi_i^{(\mathrm{T})})$

Algorithm converges when the difference between the current and the previous iteration of the posterior mean value is below a threshold. The algorithm is also confirmed to converge when improvement of the task reward is below a threshold over iterations. In addition, we set the maximum number of iterations to bound the computation cost. Overall, our framework switches from trajectory-dependent inference. It performs inference based on the reward distributions that are performance-dependent summaries. This guarantees effective, task-sensitive and theoretically-grounded sim-to-real domain inference.

3 Experiments

Experiments We test the proposed Task-consistent Bayesian Inference (TCBI) framework. The performance we examine is sim-to-real transfer for a hexapod robot control task. The virtual training environment for deep RL uses the MuJoCo physics simulator. In the simulator, robot models are made for simulating the physical robots with sizes of the real robot as shown in Figure 1. Simulation is set up with MuJoCo physics engine, and domain parameters are:

Dynamics parameters ξ_{dym}: mass, friction coefficient, actuator gain

Task parameters ξ_{task}: terrain slope, obstacle distribution, and initial robot state

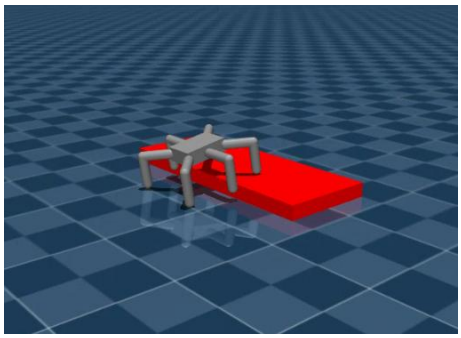


Figure 1: MuJoCo Sim Environment.

The physical robot for policy deployment is shown in Figure 2. It has 18 DoF. Control commands are computed on a PC and sent by Bluetooth for policy deployment. We collect real rollouts $\mathcal{D}_{real} = \{(s_t, a_t, r_t)\}$ with fixed policy. The related reward distribution is estimated by $P_r^{real} = p(r|\mathcal{D}_{real})$.

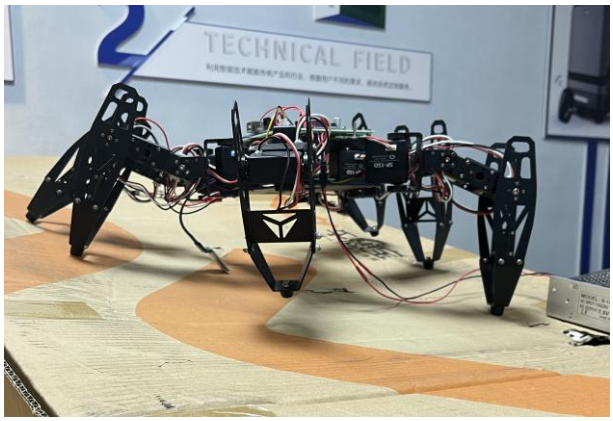


Figure 2: Real Environment used to collect data.

Our robot's state space only contains joint angle and the posture of the body. It corresponds to the sensors of the physical robot and its static gait locomotion modes. The control frequency of our robot is set to 5 Hz. This means that the policy commands control 5 times a second, which agrees with the real robot settings. Related constraints of robot both in real and virtual are shown in Table 1.

Table 1: Constraint parameters related to robot and tasks.

Properties	Value
Terrain slope	$(-\pi/6, \pi/6)$
Terrain friction	0.5~1
Block height	0.06~0.08 m
Block length	0.1~0.2 m
Block friction	0.5~1
Hexapod joint range	$(-\pi/6, \pi/6)$
Hexapod mass	0.4~1.2 kg
Motor response gain	0.1~8 N·m/rad

The specific training architecture is shown in Figure 3. Policy is implemented using an MLP. The policy has 2 hidden layers each with 256 units.

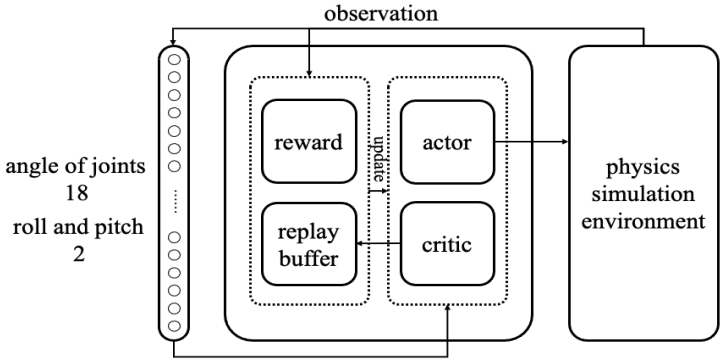


Figure 3: Deep Reinforcement Learning Training Framework.

Two training tasks are set to test the effectiveness and transfer ability of the policy, whose end episodes will never be turned off neither with the success of each task.

- Terrain is randomized with different slopes. And the policy is trained to balance body in terrain, as shown in Figure 4. The length of each episode is 8 s.

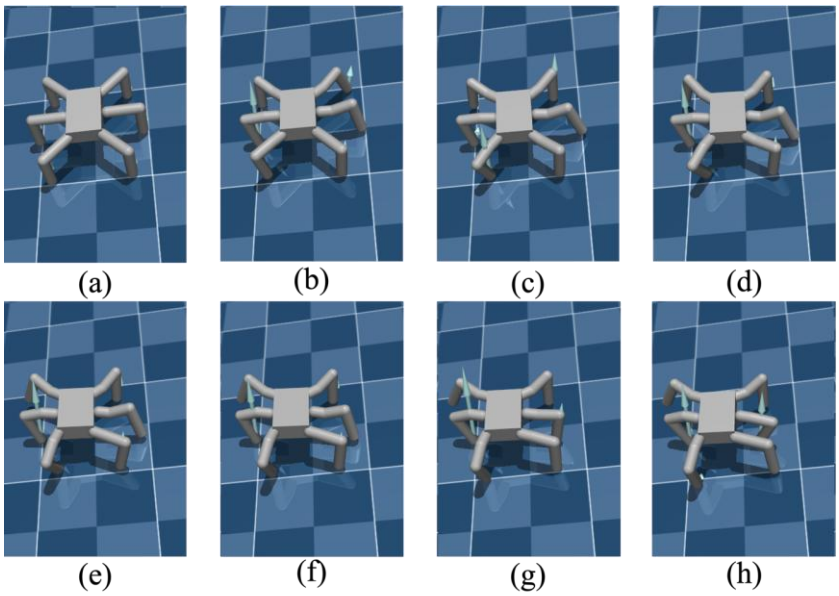


Figure 4: The process of controlling the tilted plane body posture in a virtual environment.

- Random obstacle of different sizes is placed anywhere, and the policy is trained for locomotion forward, as shown in Figure 5. The length of each episode is 30 s.

The setting of reward function is dependent on the torso attitude, traveled forward distance and the action cost. The statistics distribution of the reward function $p(r|\mathcal{D})$ used in TCBI consists of distribution of reward, distribution of body attitude and ratio of contact time. We compare our method with the following baselines:

- Domain Randomization (DR): uniform sampling over predefined parameter ranges
- Domain Adaptation via Simulation Optimization
- Normal Bayesian Optimization

We first check both distribution alignment and real-world performance:

(1) Reward Distribution Divergence We calculate the Jensen–Shannon divergence $D_{JS}(P_r^{real}, P_r^{sim}(\xi))$, which is used to measure consistency between simulation reward distribution and real-world reward distribution.

(2) Real-World Task Performance We deploy trained policy on real robot to get the average return.

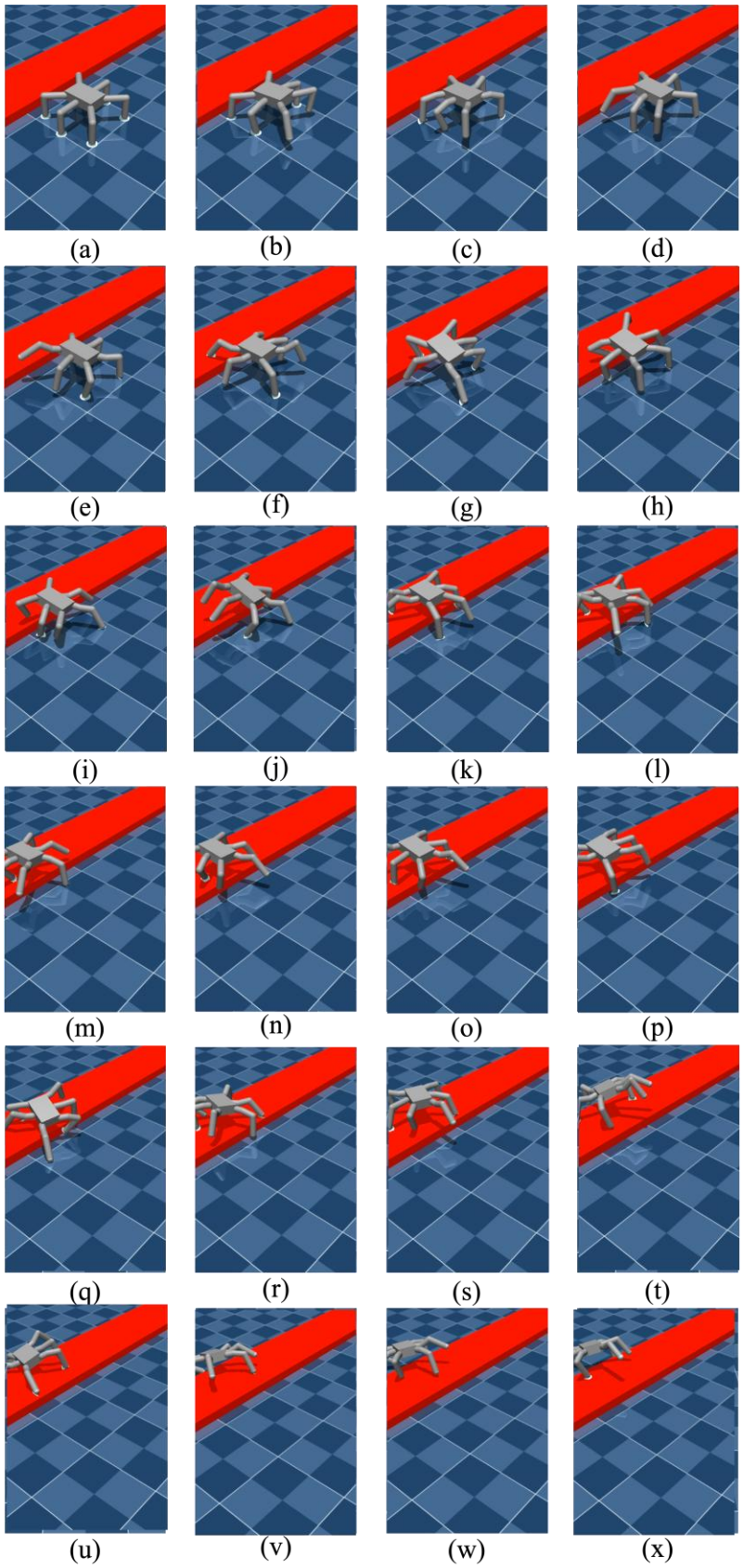


Figure 5: The process of barrier terrine forward task in a virtual environment.

4 Results

To test the performance of TCBI algorithm, we used the trajectory of the robot in deployment of RL to obtain domain parameters inference. We then used the inferred domain parameters to re-train the policy. After the re-training, the policy was used to deploy on the robot and tested the performance of the optimized policy. Related settings of baselines and RL are shown in Table 2.

Table 2: TCBI, Simopt and Reinforcement Learning setup.

Properties	Value
TCBI	
Posterior sample size every iteration	50
Reward distribution sample size (Random environment setting)	20
Max iteration	15
α	10
Simopt	
Number of simulation parameter samples per update	1200
Discrepancy function threshold	0.06
Minimum temperature of sample weights (perturb scale)	0.04
L1-cost weight of discrepancy function	0.5
L2-cost weight of discrepancy function	1
RL	
Distance reward	90000
Attitude reward	9000
Action cost	-600
Training Steps of Balance task	3e6
Training Steps of Forward task	5e6
Learning rate	1e-3
Batch size	256
γ	0.99
τ	0.005

Figure 6-8 shows the mean and variance of the domain parameter ξ during the calculation process. 3 domain parameters are iterated in balance tasks for 10 times and forward task for 12 times.

Posterior variance contraction from iteration 1 to final iteration is shown in Table 3.

Table 3: Variance reduction during the iterative process of inferring the domain parameters (compare to initial settings).

	ABC	Simopt	TCBI
	Balance Task		
Friction	31.82%	92.84%	82.66%
Mass	1.83%	67.90%	62.95%
Gain	86.90%	62.10%	54.39%
	Forward Task		
Friction	-6.59%	72.50%	73.06%
Mass	-24.48%	77.23%	79.63%
Gain	6.37%	47.47%	67.96%

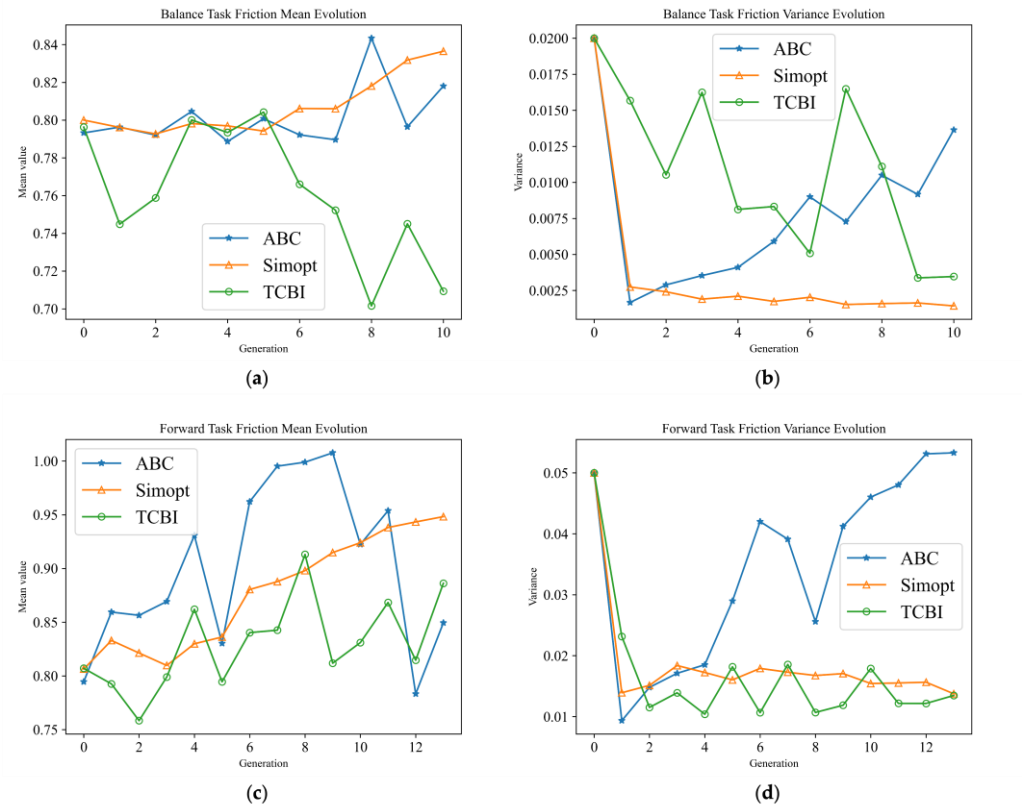
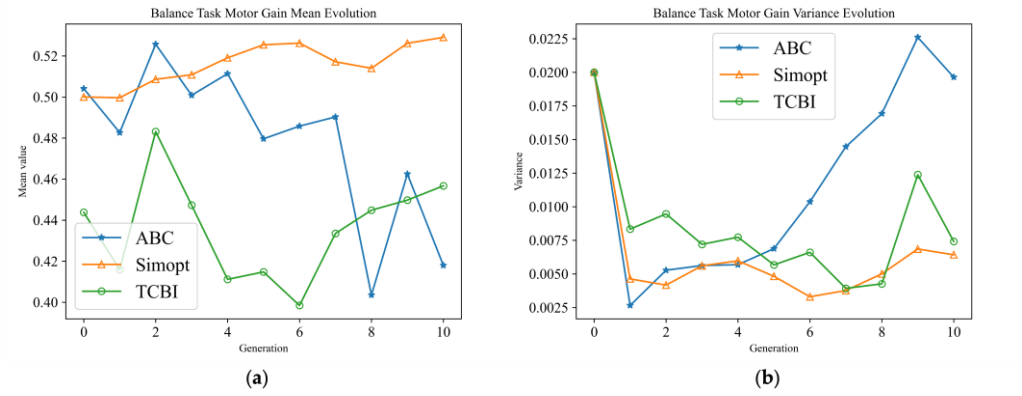


Figure 6: (a) The mean value of the friction coefficient distribution during the iterative process of inferring the domain parameters of the balancing task; (b) The variance value of the friction coefficient distribution during the iterative process of inferring the domain parameters of the balancing task; (c) The mean value of the friction coefficient distribution during the iterative process of inferring the domain parameters of the forward task; (b) The variance value of the friction coefficient distribution during the iterative process of inferring the domain parameters of the forward task.



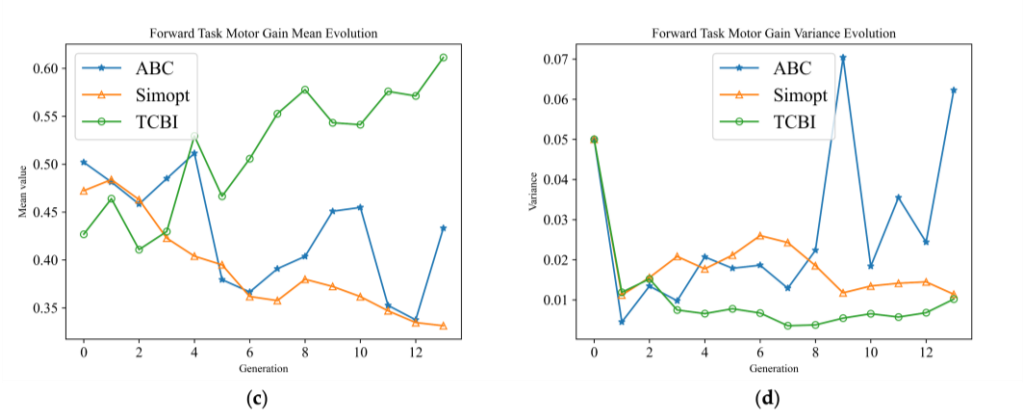


Figure 7: (a) The mean value of the motor gain distribution during the iterative process of inferring the domain parameters of the balancing task; (b) The variance value of the motor gain distribution during the iterative process of inferring the domain parameters of the balancing task; (c) The mean value of the motor gain distribution during the iterative process of inferring the domain parameters of the forward task; (b) The variance value of the motor gain distribution during the iterative process of inferring the domain parameters of the forward task.

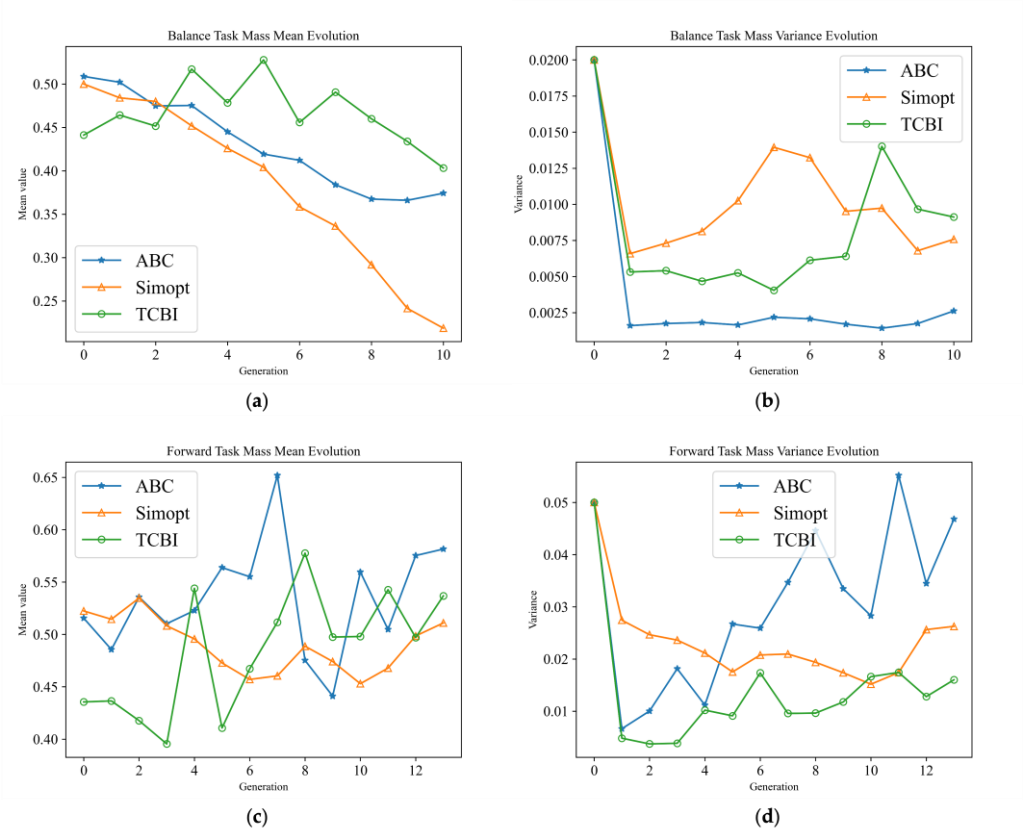


Figure 8: (a) The mean value of the mass distribution during the iterative process of inferring the domain parameters of the balancing task; (b) The variance value of the mass distribution during the iterative process of inferring the domain parameters of the balancing task; (c) The mean value of the mass distribution during the iterative process of inferring the domain parameters of the forward task; (b) The variance value of the mass distribution during the iterative process of inferring the domain parameters of the forward task.

Compared to early iterations, parameter distributions are wide. Later iterations converge towards a narrow region. This demonstrates that the inferred parameters become closer to the real system dynamics. Moreover, the means show task-dependent convergence. For example, in the balance task, the friction coefficient converges earlier. In the forward locomotion task, convergence is slower and has mild oscillations. This difference reflects a fact. Forward locomotion contains more complicated contact dynamics and longer temporal dependencies. Such factors make the inference problem harder. Similarly, motor gain and mass parameters converge more smoothly for the balance task compared with forward task. This means that static or quasi-static tasks give more informative and stable signals of the domain for the inference problem. Unlike dynamic tasks, dynamic tasks introduce more variability to reward distributions.

Overall, these results prove that TCBI can indeed gradually refine the distributions of domain parameters. This refines to match a task's features instead of collapsing to a single constant parameter setting. The real-world balance task performance of the policy trained with retrained using the inferred domain parameters are presented in Figure 9 and Table 4. Experiment is repeated 30 times.

Table 4: The balance task performance of the policy for re-training based on the domain parameters of the inference (repeat the experiment 30 times).

	Fixed DR	ABC	Simopt	TCBI (reward only)	TCBI
Reward	386.80±411.85	617.08±352.27	667.06±318.38	623.04±333.99	839.6±156.27
Final inclination angle (degree)	7.83±3.87	5.39±2.97	4.83±2.53	4.85±2.14	2.28±0.99

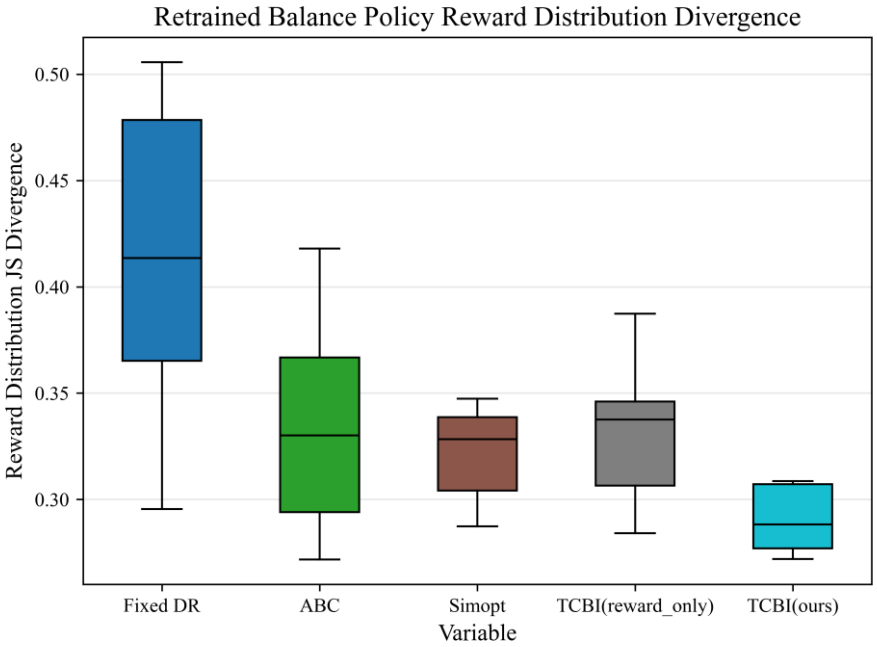


Figure 9: Retrained Balance Policy Reward Distribution Divergence across Baselines.

The real-world forward task performance of the policy trained with retrained using the inferred domain parameters are presented in Figure 10 and Table 5.

Table 5: The forward task performance of the policy for re-training based on the domain parameters of the inference (repeat the experiment 30 times).

	Fixed DR	ABC	Simopt	TCBI (reward only)	TCBI
Reward	7602.1±6867.9	10822.6±4504.4	12530.9±3356.8	12008.2±4176.95	15262±2880.3
Travel Distance (cm)	29.3±14.6	36.5±9.3	38.1±11.1	37.8±12.6	44.3±8.1

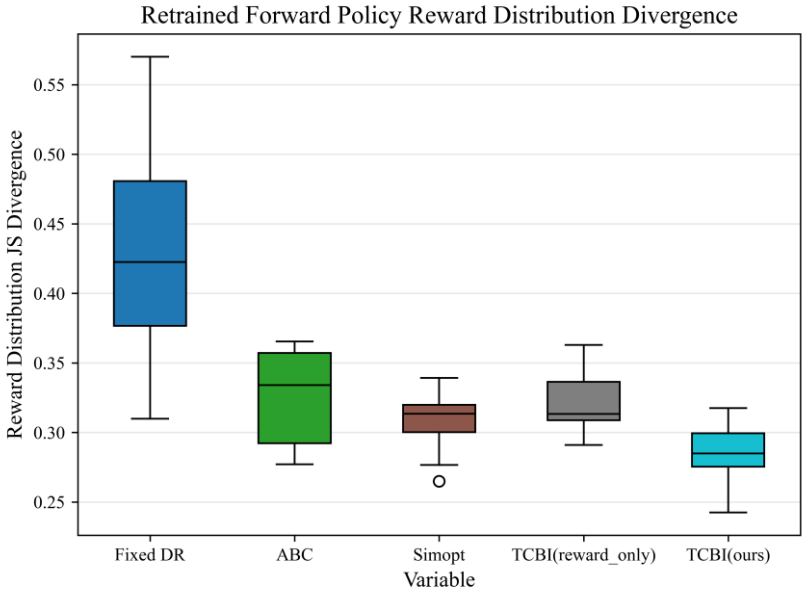


Figure 10: Retrained Forward Policy Reward Distribution Divergence across Baselines.

To compare the reward and reward distribution divergence, we perform a two-sample Welch's t-test between them. The results are shown in Table 6. TCBI significantly outperforms ABC and SimOpt in both reward ($p < 0.05$), which rejects their hypothesis that their means are equal.

Table 6: P-values of Pairwise Comparisons of Reward and JS Distribution Divergence.

	Fix-DR vs TCBI	ABC vs TCBI	Simopt vs TCBI
	Reward		
Balance Task	0.00006289	0.00344813	0.01217334
Forward Task	0.00000000	0.00033816	0.00693335
	Reward Distribution Divergence		
Balance Task	0.00029974	0.02412016	0.00302445
Forward Task	0.00014992	0.00490966	0.02593419

These results also indicate that TCBI always obtains the smallest JS divergence. That is to say, it leads to the highest degree of match between simulation and real reward distribution. Fixed DR has the largest divergence. That is, manually-designed parameter range cannot reproduce reality well. ABC is better than Fixed DR because it uses data-driven inference. But its performance is limited due to simulation trajectory-level statistics. SimOpt further decrease the divergence by optimizing simulations directly. Still, SimOpt cannot improve as much as

TCBI does. One remark is that TCBI's divergence curves fall down steadily and faster than the other curves. It proves the reward-based pseudo-likelihood we propose provides a more informative and stable optimization signal. It facilitates effective domain adaptation.

Another ablation study is given by TCBI (reward only), which drops the auxiliary task-level statistics and retains only the reward distribution in pseudo-likelihood building. TCBI (reward only) performs consistently poorer than full TCBI in both balance and forward tasks, but still far from worst results, which is outperformed by SimOpt. This means that the reward distribution alone already encodes a substantial amount of the dynamics information related to the task, which confirms the effectiveness of this compact task-like description of performance. The difference of results of TCBI (reward only) and full TCBI points to the importance of task statistics including richer descriptions of performance.

Specifically, the full TCBI jointly models the distribution of reward, the distribution of body attitude and the ratio of contact time, yielding a multi-view task-consistent representation of system behaviours. The distribution of body attitude contains direct geometric constraints on stability and system body posture consistency. The ratio of contact time models the temporal interactivity between robot and environment (especially with respect to gait regularity and contact stability for locomotion), and encodes distribution of contact time in interaction. When all these complementary statistics are combined, the inferred pseudo-likelihood is more sensitive to subtle domain mismatches that are not perceptible from the reward signals.

From Table 4, we can read this effect in both performance magnitude and variance decrease. While TCBI (reward only) improves on baseline inference methods, its variance is larger and its average return is lower than that of the full TCBI. This indicates that a reward-only alignment of tasks can admit many degenerate domain parameter explanations that yield the same return but different internal dynamics. In contrast, full TCBI de-am-biguate this ambiguity by tying inference over more task-consistent statistical views, giving rise to sharper empirical posterior concentration and more stable outcomes of the policy retraining. Overall, this ablation confirms that superiority of TCBI does not solely come from reward distribution alignment, but from integrating reward, posture and contact-based statistics in a structured way, improving identifiability and robustness of sim-to-real transfer.

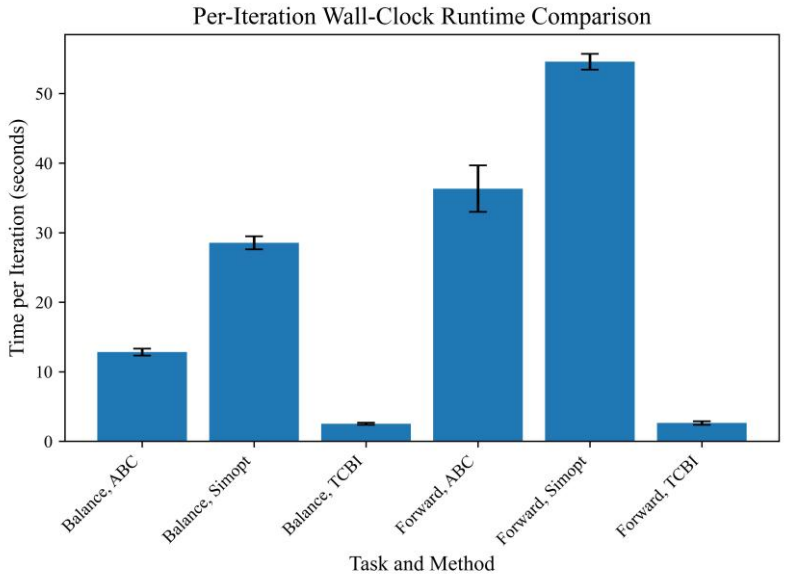


Figure 11: Per-Iteration Trajectory/Reward Statistics Wall-Clock Runtime Comparison. All methods are evaluated under the same hardware configuration, rollout horizon, and particle budget.

Wall-clock runtime comparison also shows evident computational benefits of TCBI to trajectory-based inference in Figure 11. In the balance task, ABC and SimOpt need approximately 12.8 and 28.5 s per iteration, respectively, but TCBI takes only about 2.5 s per iteration. Similar behaviors are observed for the forward locomotion task: ABC and SimOpt consume approximately 36.2 and 54.3 s per iteration, respectively, whereas TCBI takes only about 2.6 s per iteration. These results show that training trajectory-level discrepancy optimization in general suffers a considerable computational cost, particularly in dynamic tasks with longer horizons. In contrast, TCBI offers orders of magnitude reduction in runtime by updating distributions in a compact task-level distribution space without costly high-dimensional trajectory matching. This shows that our proposed reward-distribution-based inference framework provides both higher computational efficiency and higher scalability for the sim-to-real deployment.

For robustness test, state noise was added to model the uncertainty of sensor measurement of real robot, for example. The specific one can be seen from Equation 14:

$$s^* = s + x \qquad x \leftarrow \mathcal{N}(\mu, \sigma^2) \tag{14}$$

where:

s represents the original state vector

s^*represents the observed state after adding noise

x represents the random noise that follows a Gaussian distribution.

The mean of the noise μ is set to 0. By varying the covariance of different dimensions, σ^2, the measurement errors of sensors (such as encoders, IMUs), and the random disturbance in practical running mode are represented.

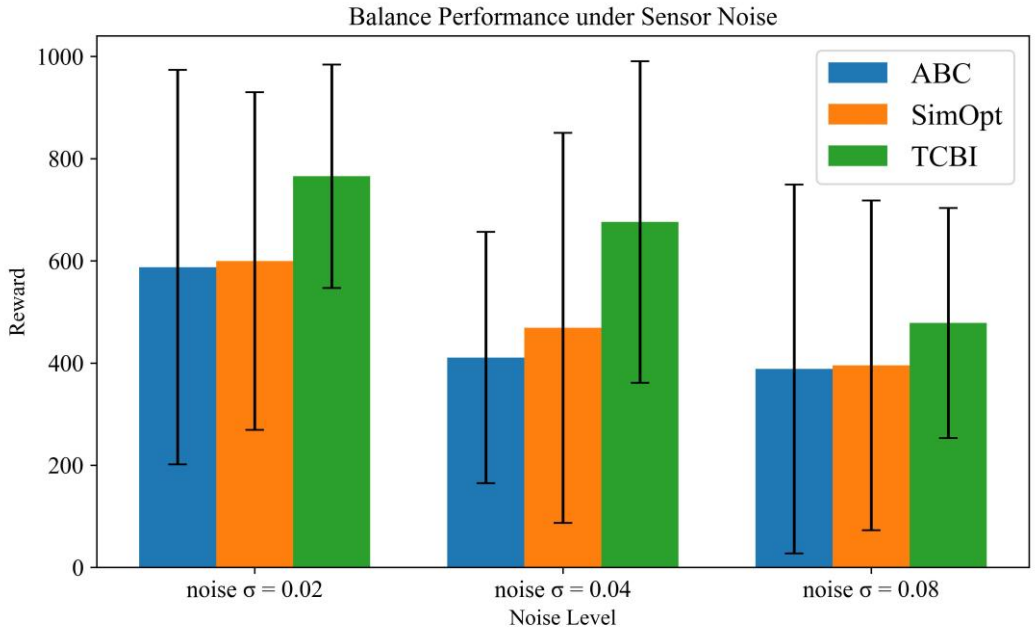


Figure 12: Balance Task Performance under Sensor Noise.

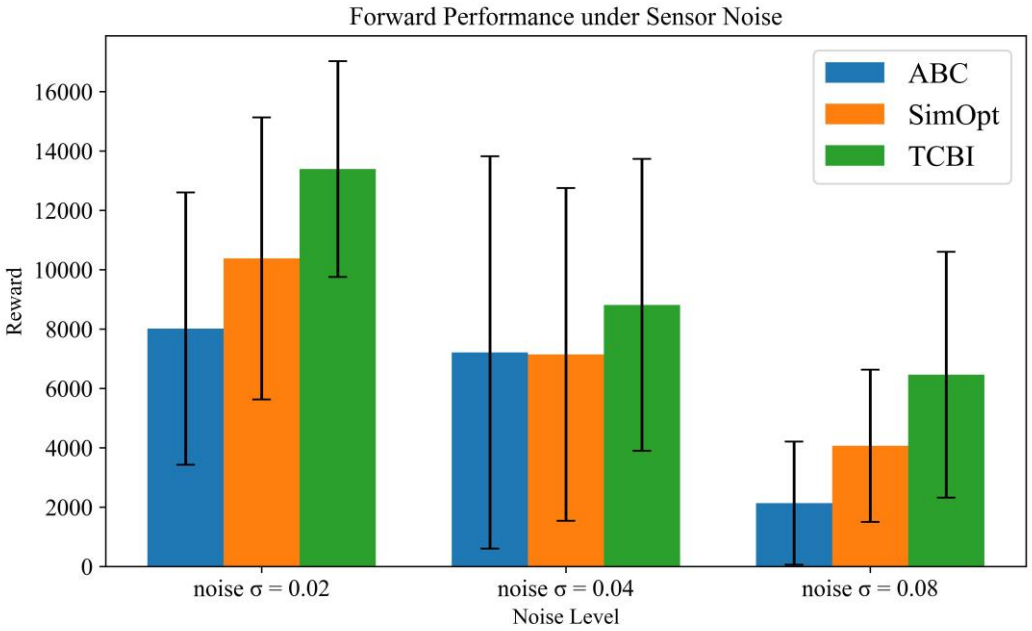


Figure 13: Forward Task Performance under Sensor Noise.

Results are displayed in Figure 12 and 13. Under sensor noise disturbances, TCBI has much stronger robustness than ABC and SimOpt in both balance and forward locomotion tasks. When the noise changes from $\sigma = 0.02$ to $\sigma = 0.08$, rewards of ABC and SimOpt vary largely and some trials even have negative return. But TCBI has stable average rewards and lower reward variance. In the balance task, TCBI can maintain the stability of the posture control under more noise disturbance. In forward locomotion task, TCBI has much smaller performance loss than ABC and SimOpt, showing that the task-consistent performance distribution inference well avoids the effect of sensor noise on the execution of policies. Overall, TCBI achieves better robustness, stability and generalization effect in noisy sensing environments.

Overall, the experimental results also show that the TCBI not only improves the accuracy of the domain parameter estimation, but also connects distribution alignment with the performance on real-world tasks. It motivates more effective, reliable sim-to-real transfer.

5 Discussion

The experimental results demonstrate that the proposed TCBI framework achieves better performance both on balance and forward walking tasks over existing baselines by obtaining higher rewards, lower variance and improving stability on sim-to-real transfer. Several observations can be made from the experimental results.

First, the reduction of posterior variance over iterations shows that the proposed task-consistent pseudo-likelihood consolidates the domain parameter distribution towards the stable regions. Compared with early iterations, the posterior distributions later have a lot less variance, demonstrating that the estimated parameters converge increasingly closer to real system dynamics. This shows that the TCBI not only improves parameter estimation accuracy but also provides a stable probabilistic inference process.

Second, the convergence of domain parameters (Figures 4–6) unveils obvious task-specific differences on several domain parameters. The inferred parameters converge with different trends in the balance and forward tasks, which means the optimal domain parameters are different for each task. This verifies the view that a set of global domain parameter values

cannot serve for multiple tasks. By contrast, TCBI implicitly adjusts the distribution of domain parameters at the task level with the task-level feedback. TCBI can make better matching between the simulated reality and real reality to serve for different tasks.

Third, compared to trajectory-level matching approaches such as ABC and SimOpt, TCBI is more stable and efficient. Trajectory-level mismatches accumulate over time. Small mismatches for an early state can lead to large mismatches later. This error accumulation makes trajectory-level matching very susceptible to noise and policy randomness. Instead, a reward distribution is a compressed view of long-horizon performance. It automatically reduces temporal error accumulation. This explains the fact that TCBI is matched to a few fewer iterations.

Fourth, an ablation study comparing the TCBI (reward only) and full TCBI explains the importance of combining various task-level performance statistics. Despite reward distributions already contain useful task-level information, adding in body attitude distributions and contact time ratios additionally improved posterior discriminability and reduced ambiguity in parameter inference. These complementary statistics can be viewed as implicit knowledge about a task's geometric stability and temporal interaction properties that are not learnt fully by reward signals alone. As such, the full TCBI obtains higher reward, lower variance and more stable deployment of policy than its reward-only variant.

Finally, compare with trajectory based approaches such as ABC and SimOpt, TCBI is much more computationally efficient. The wall-clock runtimes indicate that trajectory-level discrepancy optimization introduces large runtime overhead, in particular for locomotion control problems over long horizons. By contrast, TCBI operates in a much more compact task-level distribution space and no expensive high-dimensional comparison of trajectories is required. This dramatically reduces the run time per iteration while maintaining very good transfer performance, which indicates better practical scalability for robotic deployment. The superiority of TCBI derives from the combined effect of task-consistent probabilistic inference, multiple statistic representation of performance, better empirical posterior concentration, lower computational complexity and enhanced robustness under uncertainty.

6 Conclusions

In this paper, we proposed a Task-Consistent Bayesian Inference (TCBI) framework for sim-to-real transfer for deep reinforcement learning. In contrast to the conventional trajectory-based domain inference methods, TCBI transforms domain adaptation to a task-level probabilistic inference problem by building a pseudo-likelihood on the likelihood of the performance distribution of task-level performance. Reward distribution, body posture statistic, and contact time ratio are jointly adopted as compact and task-oriented performance statistic that represents task-relevant discrepancy between sim environment and real world. Unlike usual strategies that inferring domain information from single trajectories or single reward realization, the presented framework studies the distribution of performance statistics collected from multiple rollouts, has stronger capability of characterizing long-horizon control behaviour, and reduces dependence on the randomness of individual trajectory.

Experiments on a hexapod robot platform, however, show that TCBI consistently outperform domain randomization, A Bayesian approximate computation (ABC) and simulation optimization in both balance and forward walking tasks. Our proposed method obtains higher rewards, lower reward variance and a better reward distribution match of real environment. The posterior variance demonstrates a progressive parameter concentration in the course of inference, which verifies the effectiveness of our proposed Bayesian updating algorithm. In the ablation study, we find that by combining the reward, posture and contact

based statistics, inference identifiability and the policy robustness improve significantly when compared with only taking the reward distributions into account. In the wall-clock runtime ablation study, TCBI exhibits much lower computational time than other trajectory-based approaches by working in a small task-level representation. Experiments on robustness to the sensor noises again verify that our proposed approach has good stability in a sensing uncertain environment, exhibiting improvement of generalization ability and deployment reliability.

Despite these strengths, this approach has weaknesses. Choosing the task-level statistics still requires prior knowledge and choices can still degrade the quality of inference. Additionally, TCBI improves the data efficiency but still requires a small amount of real-world interaction. This can be undesirable in highly safety critical situations. We would like to do future work in automatically learning informative task-level representations from data. It will also further improve sample efficiency using more advanced simulation-based inference. We also plan to extend our framework to more advanced robotic systems and to multi-task scenarios. In short, TCBI offers a principled, practical approach for bridging the reality gap. We have shown that task-level probabilistic inference can greatly enhance both data efficiency and transfer performance in reality-aware robotic applications.

References

[1] Tiwari R, Khapre S, Singh A. Reinforcement learning in robotic systems: A review on sim-to-real transfer[J]. Robotics and Au-tonomous Systems, 2026: 105327.

[2] Da L, Turnau J, Kutralingam T P, et al. A survey of sim-to-real methods in rl: Progress, prospects and challenges with founda-tion models[J]. arXiv preprint arXiv:2502.13187, 2025.

[3] Tobin J, Fong R, Ray A, et al. Domain randomization for transferring deep neural networks from simulation to the real world[C]//2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2017: 23-30.

[4] Ranaweera M, Mahmoud Q H. Bridging Reality Gap Between Virtual and Physical Robot through Domain Randomization and Induced Noise[C]//Canadian AI. 2022.

[5] Shakerimov A, Alizadeh T, Varol H A. Efficient sim-to-real transfer in reinforcement learning through domain randomization and domain adaptation[J]. IEEE Access, 2023, 11: 136809-136824.

[6] Muratore F, Ramos F, Turk G, et al. Robot learning from randomized simulations: A review[J]. Frontiers in Robotics and AI, 2022, 9: 799893.

[7] Muratore F, Treede F, Gienger M, et al. Domain randomization for simulation-based policy optimization with transferability assessment[C]//Conference on Robot Learning. PMLR, 2018: 700-713.

[8] Peng X B, Andrychowicz M, Zaremba W, et al. Sim-to-real transfer of robotic control with dynamics randomization[C]//2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018: 3803-3810.

[9] Andrychowicz O A I M, Baker B, Chociej M, et al. Learning dexterous in-hand manipulation[J]. The International Journal of Robotics Research, 2020, 39(1): 3-20.

[10] Chebotar Y, Handa A, Makoviychuk V, et al. Closing the sim-to-real loop: Adapting simulation randomization with real world experience[C]//2019 international conference on robotics and automation (ICRA). IEEE, 2019: 8973-8979.

[11] Muratore F, Eilers C, Gienger M, et al. Data-efficient domain randomization with bayesian optimization[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 911-918.

[12] Ramos F, Possas R C, Fox D. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators[J]. arXiv preprint arXiv:1906.01728, 2019.

[13] Eckman D J, Henderson S G, Shashaani S. SimOpt: A testbed for simulation-optimization experiments[J]. INFORMS Journal on Computing, 2023, 35(2): 495-508.

[14] Deng Y, Li Y, Hanna J P. Abstract sim2real through approximate information states[J]. IEEE Robotics and Automation Letters, 2026.

[15] Patel S, Yin X, Huang W, et al. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards[C]//2025 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2025: 8258-8266.

[16] Hu X, Sun Q, He B, et al. Impact of Static Friction on Sim2Real in Robotic Reinforcement Learning[C]//2025 IEEE/RSJ Interna-tional Conference on Intelligent Robots and Systems (IROS). IEEE, 2025: 17107-17114.

[17] Ho J, Ermon S. Generative adversarial imitation learning[J]. Advances in neural information processing systems, 2016, 29.

[18] Tiboni G, Klink P, Peters J, et al. Domain randomization via entropy maximization[C]// International Conference on Learning Representations. 2024, 2024: 19841-19863.

[19] Ghavamzadeh M, Mannor S, Pineau J, et al. Bayesian reinforcement learning: A survey[J]. Foundations and Trends® in Ma-chine Learning, 2015, 8(5-6): 359-483.

[20] Papamakarios G, Sterratt D, Murray I. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows[C]//The 22nd international conference on artificial intelligence and statistics. PMLR, 2019: 837-848.

[21] Lueckmann J M, Bassetto G, Karaletsos T, et al. Likelihood-free inference with emulator networks[C]//Symposium on ad-vances in approximate Bayesian inference. PMLR, 2019: 32-53.

[22] Csilléry K, Blum M G B, Gaggiotti O E, et al. Approximate Bayesian computation (ABC) in practice[J]. Trends in ecology & evolution, 2010, 25(7): 410-418.