Bridging Imagination and Reality for Model-Based Deep Reinforcement Learning

Guangxiang Zhu*
IIIS
Tsinghua University
guangxiangzhu@outlook.com

Minghao Zhang*
School of Software
Tsinghua University
mehoozhang@gmail.com

Honglak Lee
EECS
University of Michigan
honglak@eecs.umich.edu

Chongjie Zhang
IIIS
Tsinghua University
chongjie@tsinghua.edu.cn

Abstract

Sample efficiency has been one of the major challenges for deep reinforcement learning. Recently, model-based reinforcement learning has been proposed to address this challenge by performing planning on imaginary trajectories with a learned world model. However, world model learning may suffer from overfitting to training trajectories, and thus model-based value estimation and policy search will be prone to be sucked in an inferior local policy. In this paper, we propose a novel model-based reinforcement learning algorithm, called **BrIdging Reality** and **D**ream (**BIRD**). It maximizes the mutual information between imaginary and real trajectories so that the policy improvement learned from imaginary trajectories can be easily generalized to real trajectories. We demonstrate that our approach improves sample efficiency of model-based planning, and achieves state-of-the-art performance on challenging visual control benchmarks.

1 Introduction

Reinforcement learning (RL) is proposed as a general-purpose learning framework for artificial intelligence problems, and has led to tremendous progress in a variety of domains [1, 2, 3, 4]. Model-free RL adopts a trail-and-error paradigm, which directly learns a mapping function from observations to values or actions through interactions with environments. It has achieved remarkable performance in certain video games and continuous control tasks because of its simplicity and minimal assumptions about environments. However, model-free approaches are not yet sample efficient and require several orders of magnitude more training samples than human learning, which limits its applications on real-world tasks [5].

A promising direction for improving sample efficiency is to explore model-based RL, which first builds an action-conditioned world model and then performs planning or policy search based on the learned model. The world model needs to encode the representations and dynamics of an environment is then used as a "dreamer" to do multi-step lookaheads for planning or policy search. Recently, world models based on deep neural networks were developed to handle dynamics in complex high-dimensional environments, which offers opportunities for learning model-based polices with visual observations [6, 7, 8, 9, 10, 11, 12, 13].

*Equal Contribution

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Model-based frameworks can be roughly grouped into four categories. First, Dyna-style algorithms alternate between building the world model from interactions with environments and performing policy optimization on simulated data generated by the learned model [14, 15, 16, 17, 11]. Second, model predictive control (MPC) and shooting algorithms alternate model learning, planning and action execution [18, 19, 20]. Third, model-augmented value expansion algorithms use model-based rollouts to improve targets for model-free temporal difference (TD) updates or policy gradients [21, 9, 6, 10]. Fourth, analytic-gradient algorithms leverage the gradients of the model-based imaginary returns with respect to the policy and directly propagate such gradients through a differentiable world model to the policy network [22, 23, 24, 25, 26, 27, 13]. Compared to conventional planning algorithms that generate numerous rollouts to select the highest performing action sequence, analytic-gradient algorithm is more computationally efficient, especially in complex domains with deep neural networks. Dreamer [13] as a landmark of analytic-gradient model-based RL, achieves state-of-the-art performance on visual control tasks.

However, most existing breakthroughs on analytic gradients focus on optimizing the policy on imaginary trajectories and leave the discrepancy between imagination and reality largely unstudied, which often bottlenecks their performance on real trajectories. In practice, a learning-based world model is not perfect, especially in complex environments. Unrolling with an imperfect model for multiple steps generates a large accumulative error, leaving a gap between the generated trajectories and reality. If we directly optimize policy based on the analytic gradients through the imaginary trajectories, the policy will tend to deviate from reality and get sucked in an inferior local solution.

Evidence from humans' cognition and learning in the physical world suggests that humans naturally have the capacity of self-reflection and introspection. In everyday life, we track and review our past thoughts and imaginations, introspect to further understand our internal states and interactions with the external world, and change our values and behavior patterns accordingly [28, 29]. Inspired by this insight, our basic idea is to leverage information from real trajectories to endow policy improvement on imaginations with awareness of discrepancy between imagination and reality. We propose a novel reality-aware model-based framework, called **B**r**I**dging **R**eality and **D**ream (**BIRD**), which performs differentiable planning on imaginary trajectories, as well as enables adaptive generalization to reality for learned policy by optimizing mutual information between imaginary and real trajectories. Our model-based policy optimization framework naturally unifies confidence-aware analytic gradients, entropy regularization maximization, and model learning. We conduct experiments on challenging visual control benchmarks (DeepMind Control Suite with image inputs [30]) and the results demonstrate that BIRD achieves state-of-the-art performance in terms of sample efficiency. Our ablation study further verifies the superiority of BIRD benefits from mutual information maximization rather than from the increase of policy entropy.

2 Related Work

Model-Based Reinforcement Learning Model-based RL exhibits high sample efficiency and has been widely used in several real-world control tasks, such as robotics [31, 32, 7]. Dyna-style algorithms [14, 15, 16, 17, 11] optimize policies with samples generated from a learned world model. Model predictive control (MPC) and shooting methods [18, 19, 20] leverage planning to select actions, but suffer from expensive computation. In model-augmented value expansion approaches, MVE [21], VPN [6] and STEVE [9] use model-based rollouts to improve targets for model-free TD updates. MuZero [10] further incorporates Monte-Carlo tree search (MCTS) and achieves remarkable performance on Atari and board games. To manage visual control tasks, VisualMPC [33] introduces a visual prediction model to keep track of entities through occlusion by temporal skip connections. PlaNet [12] improves the model learning by combining deterministic and stochastic latent dynamics models. [34] presents a summary of model-based approaches and benchmarks popular algorithms for comparisons and extensions.

Analytic Value Gradients If a differentiable world model is available, analytic value gradients are proposed to directly update the policy by gradients that flow through the world model. PILCO [24] and iLQR [25] compute an analytic gradient by assuming Gaussian processes and linear functions for the dynamics model, respectively. Guided policy search (GPS) [26, 35, 36, 37, 38] uses deep neural networks to distill behaviors from the iLQR controller. Value Gradients (VG) [22] and Stochastic Value Gradients (SVG) [23] provide a new direction to calculate analytic value gradients through a generic differentiable world model. Dreamer [13] and IVG [27] further extend SVG by

generating imaginary rollouts in the latent space. However, these works focus on improving the policy in imaginations, leaving the discrepancy between imagination and reality largely unstudied. Our approach enables policy generalization to real-world interactions by maximizing mutual information between imagination and real trajectories, while optimizing the policy on imaginary trajectories. In addition, alternative end-to-end planning methods [39, 40] leverage analytic gradients, but they either focus on online planning in simple tasks [39] or require goal images and distance metrics for the reward function [40].

Information-Based Optimization In addition to maximizing the expected return objective, a reliable RL agent may exhibit more characteristics, like meaningful representations, strong generalization, and efficient exploration. Deep information-based methods [41, 42, 43, 44] recently show progress towards this direction. [45, 46, 47] are proposed to learn more efficient representations. Maximum entropy RL maximizes the entropy regularized return to obtain a robust policy [48, 49] and [50, 51] further connect policy optimization under such regularization with value based RL. [52] learns a goal-conditioned policy with information bottleneck to identify decision states. IDS [53] estimates the information gain for a sampling-based exploration strategy. These algorithms mainly focus on facilitating policy learning in the model-free setting, while BIRD aims at bridging imagination and reality by mutual information maximization in the context of model-based RL.

3 Preliminaries

3.1 Reinforcement Learning

A reinforcement learning agent aims at learning a policy to maximize the cumulative rewards by exploring in a Markov Decision Processes (MDP) [54]. Normally, we use denote time step as t and introduce state $s_t \in \mathcal{S}$, action $a_t \in \mathcal{A}$, reward function $r(s_t, a_t)$, a policy $\pi_\theta(s)$, and a transition probability $p(s_{t+1}|s_t, a_t)$ to characterize the process of interacting with the environment. The goal of the agent is to find a policy parameter θ that maximizes the long-horizon summed rewards represented by a value function $v_\phi(s_t) \doteq \mathbb{E}\left(\sum_{i=t}^{t+H} \gamma^{i-t} r_i\right)$ parameterized with ϕ. In model-based RL, the agent builds a world model p_ψ parameterized by ψ for environmental dynamics p and reward function r, and then performs planning or policy search based on this model.

3.2 World Model

Considering that several complex tasks (e.g., visual control tasks [30]) are partially observable Markov decision process (POMDP), this paper adopts a similar world model with PlaNet [12] and Dreamer [13], which learns latent states from the history of visual observations and models the latent dynamics by LSTM-like recurrent networks. Specifically, the world model consists of the following modules:

$$\begin{aligned} \text{Representation model}: &\quad s_t \sim p_\psi(s_t|s_{t-1}, a_{t-1}, o_t) \\ \text{Transition model}: &\quad s_t \sim p_\psi(s_t|s_{t-1}, a_{t-1}) \\ \text{Observation model}: &\quad o_t \sim p_\psi(o_t|s_t) \\ \text{Reward model}: &\quad r_t \sim p_\psi(r_t|s_t). \end{aligned} \tag{1}$$

The representation model encodes the image input into a compact latent space and the long-horizon dynamics on latent states are captured by a latent transition model. We use RSSM [12] as our transition model, which combines deterministic and stochastic transition model in order to learn dynamics more accurately and efficiently. For each latent state on the predicted trajectories, observation model learns to reconstruct its visual observations, and the reward model predicts the immediate reward. The entire world model $\mathcal{J}_\psi^{\text{Model}}$ is optimized by a VAE-like objective [55]:

$$\begin{aligned} \mathcal{J}_\psi^{\text{Model}}(\tau^{\text{img}}, \tau^{\text{real}}) = \sum_{(a_{t-1}, o_t, r_t) \sim \tau^{\text{real}}} \Big[&\ln(p_\psi(o_t|s_t)) + \ln(p_\psi(r_t|s_t)) \\ &- \beta D_{\text{KL}}(p_\psi(s_t|s_{t-1}, a_{t-1}, o_t) || p_\psi(s_t|s_{t-1}, a_{t-1})) \Big]. \end{aligned} \tag{2}$$

3.3 Stochastic Value Gradients

Given a differentiable world model, stochastic value gradients (SVG) [22, 23] can be applied to directly compute policy gradient on the whole imaginary trajectory, which is a recursive composition of policy, transition, reward, and value function. According to the stochastic Bellman Equation, we have:

$$v(s) = \mathbb{E}_{\rho(\eta)}\left(r(s, \pi_\theta(s, \eta)) + \gamma \mathbb{E}_{\rho(\xi)}\left(v(p(s, \pi_\theta(s, \eta), \xi))\right)\right), \tag{3}$$

where $\eta \sim \rho(\eta)$ and $\xi \sim \rho(\xi)$ are noises from a fixed noise distribution for re-parameterization. So the gradients through trajectories can be iteratively computed as:

$$\begin{aligned} \frac{\partial v}{\partial s} &= \mathbb{E}_{\rho(\eta)}\left(\frac{\partial \mathrm{r}}{\partial s} + \frac{\partial r}{\partial a}\frac{\partial \pi}{\partial s} + \gamma \mathbb{E}_{\rho(\xi)}\left(\frac{\partial v}{\partial s'}\left(\frac{\partial p}{\partial s} + \frac{\partial p}{\partial a}\frac{\partial \pi}{\partial s}\right)\right)\right) \\ \frac{\partial v}{\partial \theta} &= \mathbb{E}_{\rho(\eta)}\left(\frac{\partial r}{\partial a}\frac{\partial \pi}{\partial \theta} + \gamma \mathbb{E}_{\rho(\xi)}\left(\frac{\partial v}{\partial s'}\frac{\partial p}{\partial a}\frac{\partial \pi}{\partial \theta} + \frac{\partial v}{\partial \theta}\right)\right), \end{aligned} \tag{4}$$

where s' denotes the next state given by the transition function. Intuitively, policy can be improved by propagating analytic gradients with respect to the policy network through the imaginary trajectories.

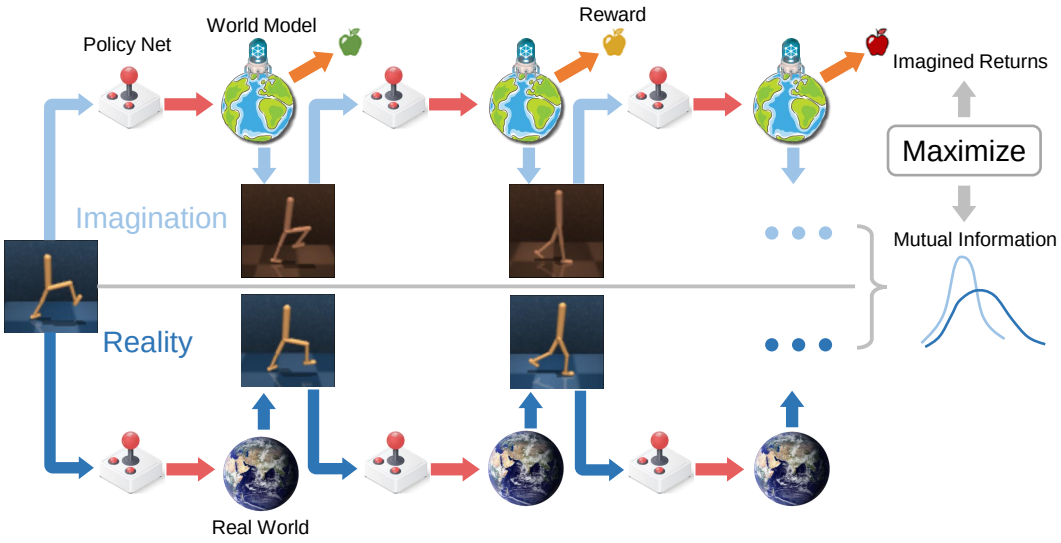

Figure 1: Overall framework of BIRD.

4 Reality-Aware Model-Based Policy Improvement

In this section, we present a novel model-based RL framework, called **BrIdging Reality and Dream** (**BIRD**), as shown in Figure 1. The agent represents its policy function with a policy network ( ). To estimate the future effects of its policy and enable potential policy improvement, it unrolls trajectories based on its world model ( ) using the current policy and optimizes the accumulative rewards on the imaginary trajectories. The policy network and differentiable world model connect to one another forming a larger trainable network, which supports differentiable planning and allows the analytic gradients of accumulative rewards with respect to the policy flow through the world model. In the meantime, the agent also interacts with the real world ( ) and generates real trajectories. BIRD maximizes the mutual information between real and imaginary trajectories to endow both the policy network and the world model with adaptive generalization to real-world interactions. In summary, BIRD maximizes the total objective function:

$$\mathcal{J}_{\mathrm{BIRD}} = \mathcal{J}_\theta^{\mathrm{SVG}}(\tau^{\mathrm{img\_roll}}) - \mathcal{L}_\phi^{\mathrm{TD}}(\tau^{\mathrm{img\_roll}}) + w I_{\theta,\psi}(\tau^{\mathrm{img}}, \tau^{\mathrm{real}}), \tag{5}$$

where τ^{real} and τ^{img} indicate the real trajectories and the corresponding imaginary trajectories under the same policy, and $\tau^{\mathrm{img\_roll}}$ indicate the rolled out imaginary trajectories during the optimization

of policy improvement. θ, ϕ, ψ, are parameters of policy network π_θ, value network v_ϕ, and world model p_ψ, respectively. The first two terms $\mathcal{J}_\theta^{\mathrm{SVG}}(\tau^{\mathrm{img\_roll}}) - \mathcal{J}_\phi^{\mathrm{TD}}(\tau^{\mathrm{img\_roll}})$ account for policy improvement on imaginations, the last term $I_{\theta,\psi}(\tau^{\mathrm{img}}, \tau^{\mathrm{real}})$ optimizes the mutual information, and w is a weighting factor between them.

In conventional model-based RL approaches, real-world trajectories are normally used to optimize model prediction error, which is quite different from BIRD. In complex domains, optimizing model prediction error cannot guarantee a perfect predictive model. Unrolling with such an imperfect model for multiple steps will generate a large accumulative error, leaving a large gap between the generated trajectories and real ones. Thus, policy optimized by such a model may overfit undesirable imaginations and have a low generalization ability to reality, which is also shown in our experiments (Figure 3). This problem is further exacerbated in analytic-gradient RL that performs differentiable planning by gradient-based local search. This is because even a small gradient step along the imperfect model can easily reach a non-generalizable neighbourhood and lead to a direction of incorrect policy improvement. To address this problem, our method optimizes mutual information with respect to both the model and the policy, which makes policy improvement aware of the discrepancy between real and imaginary trajectories. Intuitively, BIRD optimizes the world model to be more real and reinforces the actions whose resulting imaginations not only have large accumulative rewards, but also resemble real trajectories. As a result, BIRD learns a policy from imaginations with easier generalization to the real-world environment.

4.1 Policy Improvement on Imaginations

As a model-based RL algorithm, BIRD improves the policy by maximizing the accumulative rewards of the imaginary trajectories unrolled by the world model. Conventional model-based approaches [18, 7, 11] perform policy improvement by selecting the optimal action sequence that maximizes the expected planning reward, that is $\max_{a_{t:t+H}} \mathbb{E}_{s_x \sim p_\psi} \sum_{x=t}^{t+H} r(s_x, a_x)$. If the world model is differentiable, we use stochastic value gradients (SVG) to directly leverage the gradients through the world model for policy improvement. Similar with Dreamer [13], our objective of maximizing the model-based value expansion within horizon H is given by:

$$\begin{aligned} &\mathcal{J}_\theta^{\mathrm{SVG}}(\tau^{\mathrm{img}}) = \max_\theta \sum_{x=t}^{t+H} \mathrm{V}_\lambda(s_x), \\ &\mathrm{V}_\lambda(s_x) = \mathbb{E}_{a_i \sim \pi_\theta, s_i \sim p_\psi(s_i|s_{i-1}, a_{i-1})} \sum_{k=1}^{H} \lambda_k \left[\left(\sum_{i=t}^{h-1} \gamma^{i-t} r_i \right) + \gamma^{h-t} v_\phi(s_h) \right], \end{aligned} \tag{6}$$

where r_i represents the immediate reward at timestep i predicted by the world model ψ. For each expand length k, we expand the expected value from current timestep x to timestep $h-1$ ($h = \min(x+k, t+H)$) and use learned value function $v_\phi(s_h)$ to estimate returns beyond h steps, i.e., $v_\phi(s_h) = \mathbb{E}\left(\sum_{i=h}^{H} \gamma^{i-h} r_i \right)$. Here, we use the exponentially-weighted average of the estimates for different values of k to balance bias and variance, and the exponential weighting factor is indicated by λ_k. As shown in the Equation 6, we alternate the policy network π_θ and the differentiable world model p_ψ, connect them to one another to form a large end-to-end trainable network, and then back-propagate the gradients of expected values with respect to policy parameters θ though this large network. Intuitively, a gradient step of the policy network encourages the world model to obtain a gradient step of new states, and in turn affect the future value. As a result, the states and policy will be optimized sequentially based on the feedback on future values. To optimize the value network, we use TD updates as actor-critic algorithms [54, 56, 21], instead of Monte Carlo estimation:

$$\mathcal{L}_\phi^{\mathrm{TD}}(\tau^{\mathrm{img}}) = \sum_{x=t}^{t+H} \| v_\phi(s_x) - \mathrm{V}_\lambda(s_x) \|^2, \tag{7}$$

4.2 Bridge Imagination and Reality by Mutual Information Maximization

To ensure the policy improvement based on the learned world model is equally effective in the real world, we introduce an information-theoretic objective, that optimizes mutual information between

real and imaginary trajectories with respect to the policy network and the world model:

$$\begin{aligned} I_{\theta,\psi}(\tau^{\text{img}}, \tau^{\text{real}}) &= \mathcal{H}(\tau^{\text{real}}) - \mathcal{H}(\tau^{\text{real}} \big| \tau^{\text{img}}) \\ &= \mathcal{H}(\tau^{\text{real}}) - \sum_u P(u) \mathcal{H}(\tau^{\text{real}} \big| \tau^{\text{img}} = u) \\ &= \mathcal{H}(\tau^{\text{real}}) + \sum_u P(u) \sum_v P(v|u) \log(P(\tau^{\text{real}} = v|u)) \\ &= \mathcal{H}(\tau^{\text{real}}) + \sum_{u,v} P(u,v) \log(P(v|u)). \end{aligned} \tag{8}$$

To reduce computational complexity, we alternately optimize the total mutual information with respect to world model and policy network. First, we fix the policy parameters θ and only optimize the parameters of world model ψ to maximize the total mutual information $I_{\theta,\psi}(\tau^{\text{img}}, \tau^{\text{real}})$. Since the first term $\mathcal{H}(\tau^{\text{real}})$ measures the entropy of real trajectories generated by policy π_θ on real MDP, it is not related to parameters of the world models ψ and we can remove this term. As for the second term $\sum_{u,v} P(u,v) \log(P(v|u))$, we consider the fact that our world model in conjunction with the policy network, can be regarded as a predictor for real trajectories and the second term serves as a log likelihood of a real trajectory of given imagined one. Thus, optimizing this term is equivalent to minimize the prediction error on training pairs of imagined and real trajectories (u, v). When the policy is fixed, $P(u, v)$ is tractable and we can directly approximate it by sampling the data from replay buffer $\mathcal{B}$ (i.e., a collection of experienced trajectories). Thus, the second term becomes $\sum_{u,v\sim\mathcal{B}} \log(P(v|u;\psi))$, which is equivalent to the conventional model prediction error $-\mathcal{L}_\psi^{\text{Model}}$. In summary, we can get the gradient,

$$\nabla_\psi I_{\theta,\psi}(\tau^{\text{img}}, \tau^{\text{real}}) = -\nabla_\psi \mathcal{L}_\psi^{\text{Model}}(\tau^{\text{img}}, \tau^{\text{real}}), \tag{9}$$

Second, we fix the model parameters ψ and only optimize the parameters of policy network θ to maximize the total mutual information $I_{\theta,\psi}(\tau^{\text{img}}, \tau^{\text{real}})$. The first term of mutual information becomes maximizing the entropy of the current policy. In some sense, this term encourages exploration and also learns a robust policy. We use a Gaussian distribution $\mathcal{N}(m_\theta(s_t), v_\theta(s_t))$ to model the stochastic policy π_θ, and thus can analytically compute its entropy on real data as $\mathbb{E}_{s_t\sim\tau^{\text{real}}} \frac{1}{2} \log 2\pi e v_\theta^2(s_t)$. Then we consider how to optimize the second term, $\sum_{u,v} P(u,v) \log(P(v|u))$. The joint distribution of real and imagined trajectories $P(u, v)$ is determined by the policy π_θ. When the updates of the world model are stopped, the log likelihood of a real trajectory of given imagined one $\log(P(v|u))$ is fixed and can be regarded as the weight for optimizing distribution $P(u, v)$ by policy. Thus, the essential objective of maximizing $\sum_{u,v} P(u,v) \log(P(v|u))$ with respect to policy parameters θ is to guide policy to the space with high confidence of model prediction (i.e., high log likelihood $\log(P(v|u))$). Specifically, we implement it by a confidence-aware policy optimization, which reweights the degree of learning by prediction confidence $\log(P(\tau^{\text{img\_roll}}|\tau^{\text{img}}))$ during the policy improvement process. The new objective of reweighted policy improvement is written as $\log(P(\tau^{\text{img\_roll}}|\tau^{\text{img}}))\mathcal{J}_\theta^{\text{SVG}}(\tau^{\text{img\_roll}})$. In addition, we normalize the confidence weight for each batch to make training stable. In summary, the gradient of policy optimization is rewritten as:

$$\begin{aligned} &\nabla_\theta \left(I_{\theta,\psi}(\tau^{\text{img}}, \tau^{\text{real}}) + \mathcal{J}_\theta^{\text{SVG}}(\tau^{\text{img}}) \right) \\ =&\nabla_\theta \left(\mathbb{E}_{s_t\sim\tau^{\text{real}}} \frac{1}{2} \log 2\pi e v_\theta^2(s_t) + \log(P(\tau^{\text{img\_roll}}|\tau^{\text{img}}))\mathcal{J}_\theta^{\text{SVG}}(\tau^{\text{img\_roll}}) \right). \end{aligned} \tag{10}$$

From Equation 9 and 10, we can see there are three terms, model error minimization, policy entropy maximization, and confidence-aware policy optimization, derivated by our total objective of optimizing mutual information between real and imaginary trajectories. We have the same model error loss as Dreamer, and thus the main difference from Dreamer is the policy entropy maximization and confidence-aware policy optimization. Intuitively, entropy maximization term aims at increasing the search space of SVG-based policy search like Dreamer and thus can explore more possibilities. Then the confidence-aware optimization term reweighs the search results by confidence, which contributes to improve the search quality and make sure the additional search results from large entropy are reliable enough. This approach has strong connections to distributional shift refinement in offline RL setting and may be beneficial to the community of batch RL [57]. In addition, considering that τ^{real}, τ^{img} and $\tau^{\text{img\_roll}}$ are trajectories under current policy, we use a first-in-first-out replay buffer with limited capacity to mimic a approximately on-policy data stream.

Algorithm 1 summarizes our entire algorithm of optimizing mutual information and policy.

Algorithm 1 BIRD Algorithm

Initialize buffer $\mathcal{B}$ with random agent.
Initialize parameters θ, ϕ, ψ randomly. Set hyper-parameters: imagination horizon H, learning step C, interacting step T, batch size B, batch length L.
while not converged **do**
  for $i = 1 \dots C$ **do**
    Draw B data sequences $\{(o_t, a_t, r_t)\}_t^{t+L}$ from $\mathcal{B}$.
    Compute latent states $s_t \sim p_\psi(s_t|s_{t-1}, a_{t-1}, o_t)$ and imaginary trajectories $\{(s_x, a_x)\}_{x=t}^{t+H}$
    For each s_x, predict rewards $p_\psi(r_x|s_x)$ and values $v_\phi(s_x)$ ▷ *Calculate imaginary returns*
    Update θ, ϕ, ψ using Equation 5 ▷ *Optimize policy and mutual information*
  end for
  Reset o_1 in real world.
  for $t = 1 \dots T$ **do**
    Compute latent state $s_t \sim p_\psi(s_t|s_{t-1}, a_{t-1}, o_t)$.
    Compute $a_t \sim \pi_\theta(a_t|s_t)$ using policy network and add exploration noise.
    Take action a_t and get r_t, o_{t+1} from real world. ▷ *Interact with real world*
  end for
  Add experience $\{(o_t, a_t, r_t)_{t=1}^T\}$ to $\mathcal{B}$.
end while

4.3 Policy Optimization with Entropy Maximization

In the context of model-free RL, maximum entropy deep RL [49, 58] contributes to learning robust policies with estimation errors, generating a question: if we simply add a maximization objective for policy entropy in the context of model-based RL with stochastic value gradients, can we also obtain policies from imaginations that generalize well to real environment? Thus, we design an ablation version of BIRD, Soft-BIRD, which just adds a entropy augmented objective to the return objective:

$$\pi_\theta^* = \arg\max_\theta \sum_t \mathbb{E}\left(r_t + \alpha\mathcal{H}(\pi(\cdot|s_t))\right), \tag{11}$$

where α is a hyper-parameter. We use a soft Bellman Equation for value function $v'_\phi(s_t)$ like SAC [49] and rewrite the objective of policy improvement $\mathcal{J}'^{\text{SVG}}_\theta$ as:

$$\begin{aligned} v'_\phi(s_t) &= \mathbb{E}\left(r_t + \alpha\mathcal{H}(\pi_\theta(\cdot|s_t)) + \gamma v'_\phi(s_{t+1})\right), \\ \mathcal{J}'^{\text{SVG}}_\theta(\tau^{\text{img}}) &= \mathbb{E}_{a_i\sim\pi_\theta, s_i\sim p_\psi(s_i|s_{i-1},a_{i-1})} \sum_{k=1}^{H} \lambda_k \left[\left(\sum_{i=t}^{h-1} \gamma^{i-t}(r_i + \alpha\mathcal{H}(\pi_\theta(\cdot|s_i)))\right) + \gamma^{h-t} v'_\phi(s_h)\right]. \end{aligned} \tag{12}$$

Compared to BIRD, soft-BIRD only maximizes the entropy of the policy instead of optimizing the mutual information between real and imaginary trajectories generated from the policy, which will provide further insights on the contribution of BIRD.

5 Experiments

We evaluate BIRD on DeepMind Control Suite (`https://github.com/deepmind/dm_control`) [30], a standard benchmark for continuous control. In Section 5.2, we compare BIRD with both model-free and model-based RL methods. For model-free baselines, we compare with D4PG [59], a distributed extension of DDPG [2], and A3C [56], the distributed actor-critic approach. We include the scores for D4PG with pixel inputs and A3C with state inputs, which are also used as baselines in Dreamer. For model-based baselines, we use PlaNet [12] and Dreamer [13], two state-of-the-art model-based RL. Some popular model-based RL papers [60, 61, 62, 63] are not inlcuded in our experiments since they use MPC for sampling-based planning and do not show effectiveness on RL tasks with image inputs. Compared to the MPC-based approaches that generate many rollouts to select the highest performing action sequence, our paper builds upon analytic value gradients that can directly propagate gradients through a differentiable world model and is more computationally efficient on domains that require learning from pixels. Our paper focuses on visual control tasks, and thus we only compare with state-of-the-art algorithms of these tasks (i.e., PlaNet and Dreamer).

In addition, we conduct an ablation experiment in Section 5.3 to illustrate the contribution of mutual information maximization. In Section 5.4, we further study cases and visualize BIRD's generalization to real-world information.

5.1 Experiment Setting

We mainly follow the experiment settings of Dreamer. Among all environments, observations are $64 \times 64 \times 3$ images, rewards are scaled to 0 to 1, and the dimensions of action space vary from 1 to 12 . Action repeat is fixed at 2 for all tasks. We implement Dreamer by its released codes (`https://github.com/google-research/dreamer`) and all hyper-parameters remain the same as reported. Since our model loss term in Equation 9 has the same form as Dreamer, we directly use the same model learning component as Dreamer that adopts multi-step prediction and removes latent overshooting used in PlaNet. We also use the same architecture for neural networks thus we have the same computational complexity as Dreamer. Specifically, CNN layers are employed to compress observations into latent state space and GRU [64] is used for learning latent dynamics. Policy network, reward network, and value network are all implemented with multi-layer perceptrons (MLP) and they respectively trained with Adam optimizer [65]. For all experiments, we select a discount factor of 0.99 and a mutual information coefficient of 1e-8. Buffersize is 100k. We train BIRD with a single Nvidia 2080Ti and a single CPU, and it takes 8 hours to run 1 million samples.

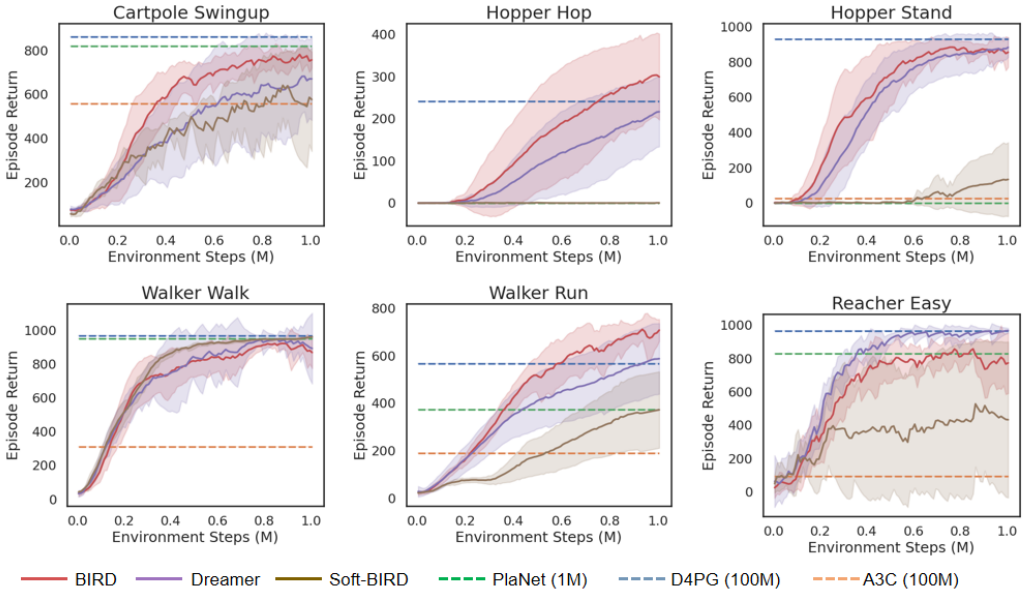

Figure 2: Results on DeepMind Control Suite. The shaded areas show the standard deviation across 10 seeds. BIRD achieves considerable performance in several challenging tasks and requires less samples than baselines.

5.2 Results on DeepMind Control Suite

Learning policy from raw visual observation has always been a challenging problem for RL algorithms. We significantly improve the state-of-the-art visual control approach on the visual control tasks from DeepMind Control Suite, which provides a promising avenue for model-based policy learning from pixels. Figure 5 shows the training curves on 6 tasks and additional results are placed in supplementary materials. Comparison results demonstrate that BIRD significantly outperforms baselines in terms of sample efficiency. We observe that BIRD can use half training samples to obtain the same score with PlaNet and Dreamer in *Hopper Stand* and *Hopper Hop*. Among all tasks, BIRD achieves comparable performance to D4PG and A3C, which are trained with 1,000 times more samples. In addition, BIRD achieves higher or similar convergence scores in all tasks than baselines. Here, we provide insights into the superiority of BIRD. As the mutual information between real and imaginary

trajectories increases, the behaviors that BIRD learns using the world model can be adapted to the real environment more appropriately and faster, while other model-based methods require a slower adaptation process. Besides, although world model usually tend to overfit poor policies in the early stage, BIRD will not be tempted by greedy policy optimization on the poor trajectories generated by such an imperfect model. Because the entropy maximization term in Equation 10 endows the agent a stronger exploration ability, and the confidence-aware policy optimization term encourages it re-estimate all the gathered trajectories and focus on optimizing high-confidence ones.

5.3 Ablation Study

In order to verify the outperformance of BIRD is not simply due to simply increasing the entropy of policy, we conduct an ablation study that compares BIRD with Soft-BIRD (4.3). Figure 5 shows the best performance of Soft-BIRD, but there is still a big gap from BIRD. As shown in *Walker Run* of Figure 5, we find that the score of Soft-BIRD first rises for a while, but eventually falls. The failure of Soft-BIRD suggests that policy improvement in model-based RL with analytic gradients is bottlenecked by the discrepancy of reality and imagination, thus only improving the entropy of policy will not help.

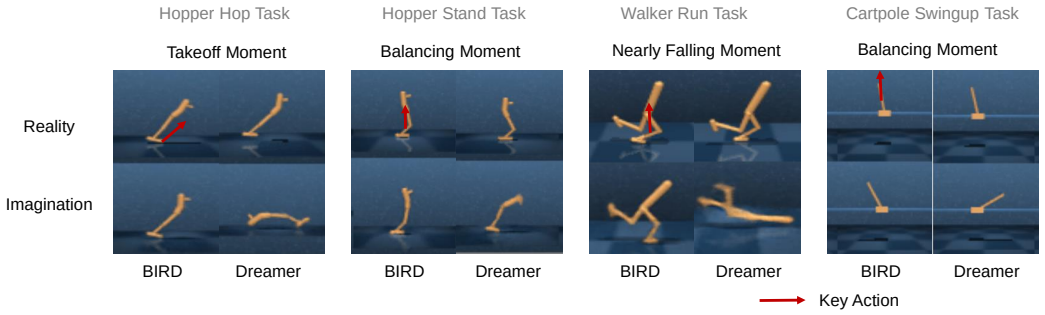

Figure 3: Prediction comparison between BIRD and Dreamer. BIRD has better predictions on key actions that will have long-term impacts, which implies that the policy generalizes to real environments well.

5.4 Case Study: Predictions on Key Actions

Our algorithm learns a world model with better generalization to real trajectories, especially on key actions which matters for long-horizon behavior learning. We visualize some predictions on key actions, such as the explosive force for standing up and jumping in *Hopper Stand* and *Hopper Hop*, stomping with front leg to prevent tumble in *Walker Run*, and throwing pole up to keep stable in *Cartpole Swingup*. As shown in Figure 3, BIRD makes more accurate predictions compared to Dreamer. For example, in *Hopper Hop*, Dreamer wrongly predicts the takeoff moment to fall down while BIRD has an accurate foresight that the agent will leap from the ground. Precise forecast of the key actions implicitly suggests that our imaginary trajectories generated by the learned policy indeed possess more real-world information.

6 Conclusion

Generalization from imagination to reality is a crucial yet challenging problem in the context of model-based RL. In this paper, we propose a novel model-based framework, called **BrIdging Reality** and **D**ream (**BIRD**), which not only performs differentiable planning on imaginary trajectories, but also encourages adaptive generalization to reality by optimizing mutual information between imaginary and real trajectories. Results on challenging visual control tasks demonstrate that our algorithm achieves state-of-the-art performance in terms of sample efficiency. Our ablation study further shows that the superiority is attributed to maximizing mutual information rather than simply increasing the entropy of the policy. In the future, we will explore directions to further improve the generalization of imaginations, such as generalizable representations and reusable skill discovery.

Broader Impact

Model-free RL requires a large amount of samples, thus limits its applications to real-world tasks. For example, the trial-and-error training process of a robot requires substantial manpower and financial resources, and certain harmful actions can greatly reduce the life of the robot. Building a world model and learning behaviors by imaginations provides a boarder prospect for real-world applications. This paper is situated in model-based RL and further improves sample efficiency over existing work, which will accelerate the development of real-world applications on automatic control, such as robotics and autonomous driving. In addition, this paper tackles a valuable problem about generalization, from imagination to reality, thus it is also of great interest to researchers in generalizable machine learning.

In the long run, this paper will improve the efficiency of factory operations, avoid artificial repetition of difficult or dangerous work, save costs, and reduce risks in the industrial and agricultural industry. For daily life, it will create a more intelligent lifestyle and improve the quality of life.

Our algorithm is a generic framework that does not leverages biases in data. We evaluated our model in a popular benchmark of visual control tasks. However, similar to a majority of deep learning approaches, our algorithm has a common disadvantage. The learned knowledge and policy is not friendly to humans and it is hard for us to know why the agent learns to act so well. Interpretability has always been a challenging open question and in the future we are interested in incorporating recent deep learning progresses on causal inference into RL.

Acknowledgments and Disclosure of Funding

This work is supported in part by Science and Technology Innovation 2030 – "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100904), and a grant from the Institute of Guo Qiang,Tsinghua University.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[2] 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. *International Conference on Learning Representations*, 2016.

[3] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] Pedro Tsividis, Thomas Pouncy, Jaqueline L. Xu, Joshua B. Tenenbaum, and Samuel J. Gershman. Human learning in atari. In *2017 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 27-29, 2017*. AAAI Press, 2017.

[6] Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. In *Advances in Neural Information Processing Systems*, pages 6118–6128, 2017.

[7] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

[8] David Ha and Jurgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018.

[9] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8234–8244, 2018.

[10] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *arXiv e-prints*, page arXiv:1911.08265, November 2019.

[11] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

[12] Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 2019.

[13] Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[14] Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990.

[15] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[16] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[17] Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, volume 87 of *Proceedings of Machine Learning Research*, pages 617–629. PMLR, 2018.

[18] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[19] Anil V Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[20] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

[21] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

[22] Michael Fairbank. Reinforcement learning by value gradients. *CoRR*, abs/0803.3539, 2008.

[23] Nicolas Heess, Gregory Wayne, David Silver, Timothy P. Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2944–2952, 2015.

[24] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[25] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[26] Sergey Levine and Vladlen Koltun. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1–9. JMLR.org, 2013.

[27] Arunkumar Byravan, Jost Tobias Springenberg, Abbas Abdolmaleki, Roland Hafner, Michael Neunert, Thomas Lampe, Noah Siegel, Nicolas Heess, and Martin Riedmiller. Imagined value gradients: Model-based policy optimization with transferable latent dynamics models. *arXiv preprint arXiv:1910.04142*, 2019.

[28] Sterling C. Johnson, Leslie C. Baxter, Lana S. Wilder, James G. Pipe, Joseph E. Heiserman, and George P. Prigatano. Neural correlates of self-reflection. *Brain*, (8):8, 2002.

[29] EG BORING. A history of introspection. *Psychological bulletin*, 50(3):169—189, May 1953.

[30] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

[31] Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 7559–7566. IEEE, 2018.

[32] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 1101–1112. PMLR, 2019.

[33] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pages 344–356. PMLR, 2017.

[34] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *CoRR*, abs/1907.02057, 2019.

[35] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1071–1079, 2014.

[36] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 156–163. IEEE, 2015.

[37] Tianhao Zhang, Gregory Kahn, Sergey Levine, and Pieter Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In Danica Kragic, Antonio Bicchi, and Alessandro De Luca, editors, *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 528–535. IEEE, 2016.

[38] Yevgen Chebotar, Mrinal Kalakrishnan, Ali Yahya, Adrian Li, Stefan Schaal, and Sergey Levine. Path integral guided policy search. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3381–3388. IEEE, 2017.

[39] Mikael Henaff, William F Whitney, and Yann LeCun. Model-based planning with discrete and continuous actions. *arXiv preprint arXiv:1705.07177*, 2017.

[40] Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks. *arXiv preprint arXiv:1804.00645*, 2018.

[41] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[42] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[43] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[44] Marylou Gabrié, Andre Manoel, Clément Luneau, Jean Barbier, Nicolas Macris, Florent Krzakala, and Lenka Zdeborová. Entropy and mutual information in models of deep neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1826–1836, 2018.

[45] Yingjun Pei and Xinwen Hou. Learning representations in reinforcement learning: An information bottleneck approach. *CoRR*, abs/1911.05695, 2019.

[46] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[47] Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A. Pires, Toby Pohlen, and Rémi Munos. Neural predictive belief representations. *CoRR*, abs/1811.06407, 2018.

[48] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 2017.

[49] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.

[50] Brendan O'Donoghue, Rémi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[51] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2775–2785, 2017.

[52] Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Hugo Larochelle, Matthew Botvinick, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[53] Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. Information-directed exploration for deep reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[54] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[55] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[56] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1928–1937. JMLR.org, 2016.

[57] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. 2020.

[58] Xiaoyu Tan, Chao Qu, Junwu Xiong, and James Zhang. S2{vg}: Soft stochastic value gradient method, 2020.

[59] Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy P. Lillicrap. Distributed distributional deterministic policy gradients. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[60] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

[61] Rinu Boney, Norman Di Palo, Mathias Berglund, Alexander Ilin, Juho Kannala, Antti Rasmus, and Harri Valpola. Regularizing trajectory optimization with denoising autoencoders. In *Advances in Neural Information Processing Systems*, pages 2859–2869, 2019.

[62] Rinu Boney, Juho Kannala, and Alexander Ilin. Regularizing model-based planning with energy-based models. In *Conference on Robot Learning*, pages 182–191. PMLR, 2020.

[63] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*, 2020.

[64] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[65] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

7 Experiment Details

We use the convolutional neural network and GRU [64] architecture for the dynamics model, with the deterministic unit size of 300 and stochastic unit size of 40. All other neural networks consist of three linear layers with ELU activations. Action network separately outputs a scaled tanh mean and a softplus standard deviation for the Normal distribution. The distribution will be transformed with a tanh function before sampling the action vector.

The learning rates for dynamics model, action network and value estimation are $6 \times 10^{-4}, 8 \times 10^{-5}, 8 \times 10^{-5}$. We use Adam [65] to optimize the networks and clip the gradients not to exceed 100. The mutual information regularizer α and the KL regularizer β are 1×10^{-8} and 1. γ and λ in value target are set as 0.99 and 0.95. The training data for dynamics model are in batch size of 50 and each of them is in length 50. The horizon steps H for imagination is 15.

When interacting with the environment, we sample a random variable from a normal distribution $\mathcal{N}(0, 0.3)$ as exploration noise. The maximum of episode length is 1000. The networks are updated for 100 steps once an episode is done. The buffer is prefilled with 5 episodes using random agent and the total buffersize is 100k. The action repeat is fixed to 2 for all tasks.

8 Additional Results

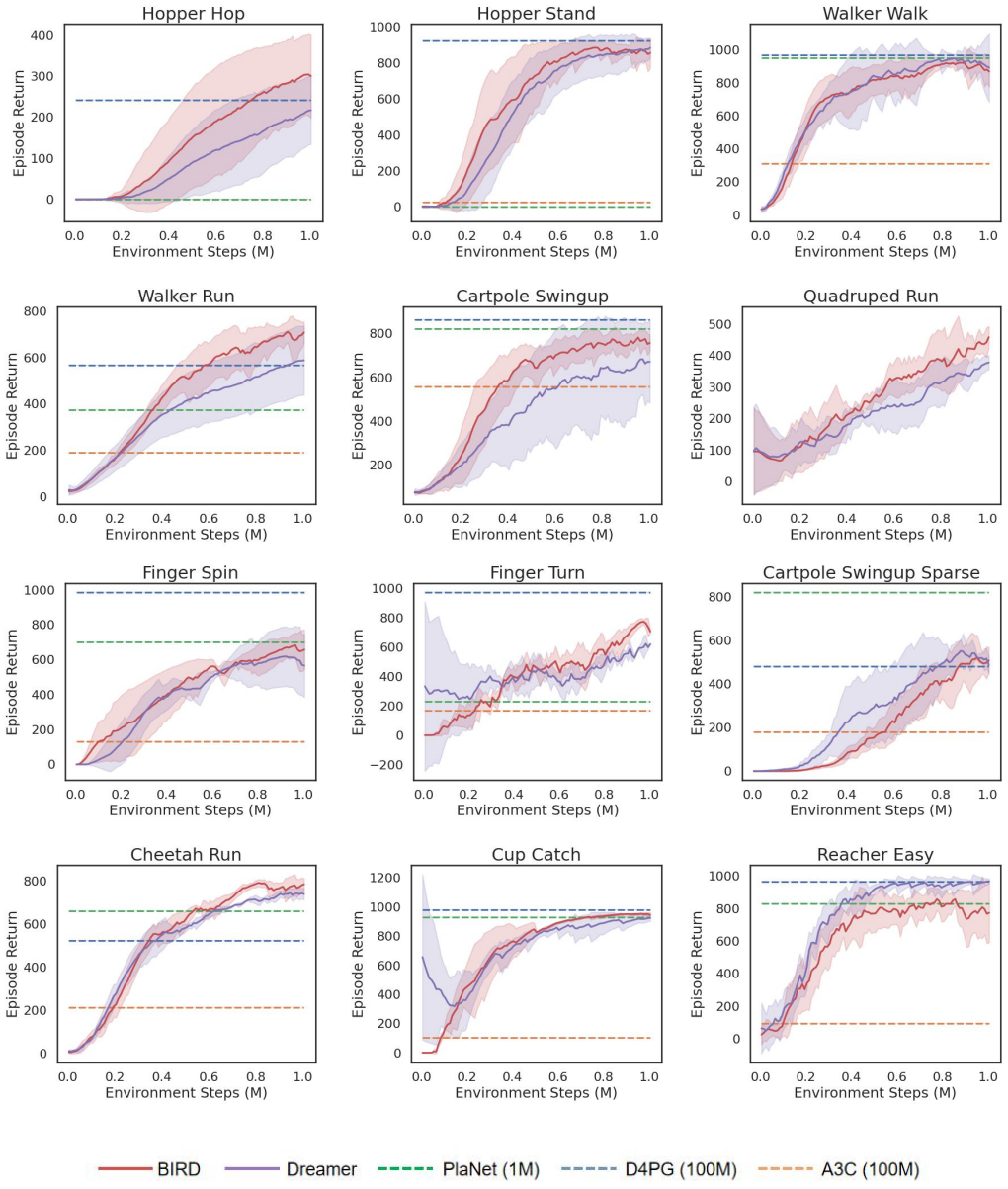

Figure 4: Results on DeepMind Control Suite. The shaded areas show the standard deviation across 3 seeds.

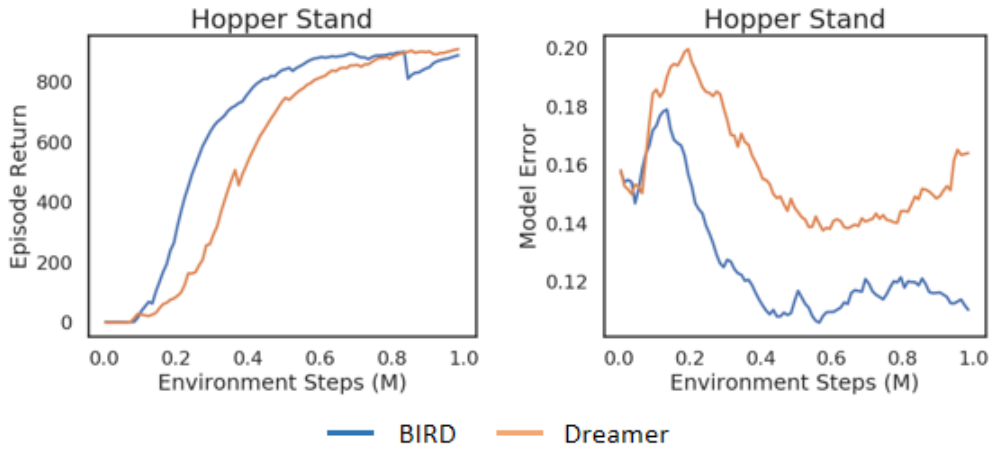

Figure 5: Comparison of model error. Since image reconstruction error will be dominated by image background and cannot reflect the prediction error on latent state, we calculate the model error as the discrepancy between latent states that predicted by model and encoded from posterior image observations. BIRD that significantly outperforms Dreamer in terms of returns has a much lower model error.