CLOSING THE TRAIN-TEST GAP IN WORLD MODELS FOR GRADIENT-BASED PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models paired with model predictive control (MPC) can be trained offline on large-scale datasets of expert trajectories and enable generalization to a wide range of tasks chosen at inference time. Compared to traditional MPC procedures, which rely either on slow search algorithms or on iteratively solving optimization problems exactly, gradient-based planning offers a computationally efficient alternative. However, the performance of gradient-based planning has thus far lagged behind that of other approaches. In this paper, we propose improved methods for training world models that enable efficient gradient-based planning. We begin with the observation that although a world model is trained on a next-state prediction objective, it is used at test-time to instead estimate a sequence of actions. The goal of our work is to close this train-test gap. To that end, we propose train-time data synthesis techniques that enable significantly improved gradient-based planning with existing world models. Moreover, we demonstrate an improvement over the search-based CEM method on an object manipulation task in 10% of the time budget.

1 INTRODUCTION

In robotic tasks, anticipating how the actions of an agent affect the state of its environment is fundamental for both prediction (Finn et al., 2016) and planning (Mohanan and Salgoankar, 2018; Kavraki et al., 2002). Classical approaches derive models of the environment evolution analytically from first principles, relying on prior knowledge of the environment, the agent, and any uncertainty (Goldstein et al., 1950; Siciliano et al., 2009; Spong et al., 2020). In contrast, learning-based methods extract such models directly from data, enabling them to capture complex dynamics and thus improve generalization and robustness to uncertainty (Sutton et al., 1998; Schrittwieser et al., 2020; LeCun, 2022).

World models (Ha and Schmidhuber, 2018), in particular, have emerged as a powerful paradigm. Given the current state and an action, the world model predicts the resulting next state. These models can be learned either from exact state information (e.g. Sutton, 1991) or directly from high-dimensional sensory inputs such as images (e.g. Hafner et al., 2023) . The latter is especially compelling as it enables perception, prediction, and control directly from raw images by leveraging pre-trained visual representations, and removes the need for measuring the precise environment states which is difficult in practice (Assran et al., 2023; Bardes et al., 2023).

Recently, world models have been shown to leverage their predictive capabilities for planning, enabling agents to solve a variety of tasks (Hafner et al., 2019a;b; Schrittwieser et al., 2020; Hafner et al., 2023; Zhou et al., 2025). A model of the dynamics is learned offline, while the planning task is defined at inference as a constrained optimization problem: given the current state, find a sequence of actions that comes as close as possible to a target state. Because the planning objective can be specified at test time, training the world model does not need to be task-specific; the same model can be reused across different tasks simply by modifying the planning objective. This inference-time optimization provides an effective alternative to reinforcement learning (RL) approaches (Sutton et al., 1998). Unlike model-free RL, which often suffers from poor sample-efficiency, or model-based RL (Hansen et al., 2023; Hafner et al., 2023), which typically requires training a separate policy for each new task, planning with world models can evaluate potential actions without interacting with

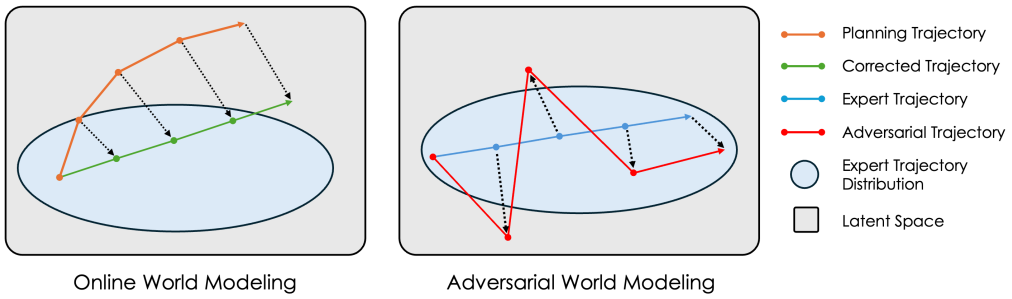

Figure 1: An overview of our two proposed methods. Online World Modeling finetunes a world model using the simulator dynamics function h to correct trajectories produced during planning that may exit the expert trajectory distribution. Adversarial World Modeling finetunes a world model on perturbations of expert trajectories such that they may exit the expert trajectory distribution and promote robustness in the world model during planning.

the environment and specifies the task only at inference time, enabling efficient generalization across multiple tasks.

Several model-based planning algorithms can be used with world models. Traditional methods, such as DDP (Mayne, 1966) and iLQR (Li and Todorov, 2004), rely on iteratively solving exact optimization problems derived from linear and quadratic approximations of the dynamics around a nominal trajectory. While highly effective in low-dimensional settings, these methods become impractical for large-scale world models, where solving the resulting optimization problem is computationally intractable. As an alternative, search-based methods such as the Cross Entropy Method (CEM) (Rubinstein and Kroese, 2004) and Model Predictive Path Integral control (MPPI) (Williams et al., 2017a) have been widely adopted as a gradient-free alternative and have proven effective in practice. However, they are computationally intensive as they require iteratively sampling candidate solutions and performing world model rollouts to evaluate each one, a procedure that scales poorly in high-dimensional spaces. Gradient-based methods (SV et al., 2023), by contrast, avoid the limitations of sampling by directly exploiting the differentiability of world models. These methods eliminate the costly rollouts required by search-based approaches, thus scaling more efficiently in high-dimensional spaces. However, gradient-based planning has seen little success to date.

Despite its advantages, gradient-based planning with world models still faces several challenges. By decoupling model learning from planning, a train-test gap naturally emerges: during training, the objective is typically next-state prediction, whereas at test time, planning involves solving an optimization problem over multiple consecutive actions. This presents several challenges. First, inference-time optimization may explore regions of the action space that the world model has never seen during training, potentially producing adversarial trajectories that achieve the desired task in the world model's representation space, but fail to achieve it when executed in the real world or in the simulator. Second, even with a flawless world model, where predictions exactly match real-world or simulator rollouts, finding a global minimum in representation spaces learned with neural networks is challenging due to non-smoothness and high non-convexity. Third, because gradients are propagated through multiple passes of the world model, errors can accumulate and compound during planning.

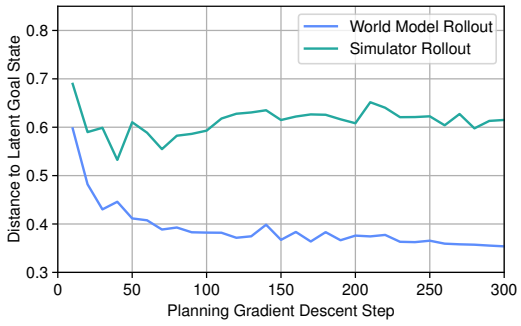

Figure 2: L_2 distance between the latent goal state and the predicted final latent state obtained from rolling out predicted actions in the world model and simulator, over the course of gradient-based planning on the PushT task.

For example, during gradient-based planning, we find that a trained latent world model is likely encountering latent states outside of its training distribution. In Figure 2, we show that a trained world model produces actions that perform substantially worse in the environment than the world model predicts they will. Although the world model is trained offline to emulate the simulator, gradient-based planning reduces the world model's ability to do so, plausibly because it pushes the world model into unexplored latent space regimes.

To address this challenge, we formulate two methods for expanding the region of familiar latent states by continuously adding new trajectories to the dataset and finetuning the world model on them, shown in Figure 1. Online World Modeling is an imitation learning based algorithm for adjusting the latent states predicted during planning using the simulator of the environment. Adversarial World Modeling is an adversarial training algorithm which finds latent states where the world model is most prone to error, to improve robustness at planning time. We demonstrate that world models finetuned on these methods significantly improve the quality of gradient-based planning and in some instances outperform the more expensive search-based method CEM.

2 Related Work

Learning world models from sensory data. Learning-based dynamics models have become central to control and decision making, offering a data-driven alternative to classical approaches that rely on first principles modeling (Goldstein et al., 1950; Schmidt and Lipson, 2009; Macchelli et al., 2009). Early work focused on modeling dynamics in low-dimensional state-space (Deisenroth and Rasmussen, 2011; Lenz et al., 2015; Henaff et al., 2017; Sharma et al., 2019), while more recent methods learn directly from high-dimensional sensory inputs such as images. One line of work trains models to predict future observations in pixel-space (Finn et al., 2016; Kaiser et al., 2019), demonstrating success in applications such as human motion prediction (Finn et al. (2016)), robotic manipulation (Finn and Levine, 2016; Agrawal et al., 2016; Zhang et al., 2019), and solving Atari games (Kaiser et al., 2019). However, pixel-level prediction is often computationally expensive due to the cost of reconstructing images. To address this, alternative approaches learn a compact latent representation where dynamics are modeled (Karl et al., 2016; Hafner et al., 2019b; Shi et al., 2022; Karypidis et al., 2024). These models are typically supervised either by decoding latent predictions to match ground truth observations (Edwards et al., 2018; Zhang et al., 2021; Bounou et al., 2021; Hu et al., 2022; Akan and Güney, 2022; Hafner et al., 2019b), or by using prediction objectives that operate directly in latent space, such as those in joint-embedding prediction architectures (JEPAs) (LeCun, 2022; Bardes et al., 2024; Drozdov et al., 2024; Guan et al., 2024; Zhou et al., 2025). Our method builds upon this latter category of world models, and specifically leverages the approach introduced in (Zhou et al., 2025).

Planning with world models. Planning with world models is challenging due to their inherent non-linearity and non-convexity. Search-based methods such as (CEM)(Rubinstein and Kroese, 2004) and MPPI (Williams et al., 2017a) are widely used in this context (Williams et al., 2017b; Nagabandi et al., 2019; Hafner et al., 2019b; Zhan et al., 2021; Zhou et al., 2025). These methods explore the action space effectively, helping to escape from local minima, but typically scale poorly in high dimensions because of their sampling nature. In contrast, gradient-based methods exploit the differentiability of the world model to optimize actions directly via backpropagation. This approach offers better scalability, but suffers from local minima, adversarial trajectories, and non-smooth objective landscapes (Bharadhwaj et al., 2020; Xu et al., 2022; Chen et al., 2022; Wang et al., 2023). To combine the strengths of both approaches, hybrid methods have been proposed. For example, (Bharadhwaj et al., 2020) interleave CEM and gradient descent steps during optimization, leveraging CEM for global exploration and gradient descent for local refinement. In this work, we focus on improving the planning capabilities of world models. Zhou et al. (Zhou et al., 2025) show that when using DINOv2 (Oquab et al., 2024) embeddings, gradient-based planning underperforms compared to CEM. We build on this approach, focusing on improving gradient-based planning performance.

Train-test gap in planning with world models A key challenge when planning with learned world models is the mismatch between training distributions and the trajectories generated during test-time planning Ajay et al. (2018); Ke et al. (2019); Zhu et al. (2023). In face, models are typically trained to minimize prediction or reconstruction error on trajectories from a dataset or a behavioral policy, but planning algorithms can generate at test-time out-of-distribution action sequences that drive the

model into poorly-trained regions, potentially leading to compounding errors or adversarial trajectories that exploit model inaccuracies (Schiewer et al., 2024; Jackson et al., 2024). Several strategies aim to address this train-test gap. Techniques like random-shooting can further help mitigate adversarial trajectories (Nagabandi et al., 2018). Alternatively, regularization-based methods, such as implicit policy training with gradient penalties, aim to improve model smoothness and planning stability (Florence et al., 2022). Closer to our approach, dataset-aggregation methods (Ross et al., 2011b) expand the training distribution by rolling out action trajectories found by the planning algorithm and adding them to the training set (Talvitie, 2014; Nagabandi et al., 2018). In a similar spirit to our approach, (Zhang et al., 2025) introduce an adversarial attack method to encourage diverse state visitation distribution in a model-based RL setting.

3 World models and Gradient-Based planning

We present two data aggregation methods for closing the train-test gap between the standard world modeling objective and gradient-based planning: Online World Modeling and Adversarial World Modeling. We provide a visual description of both methods in Figure 1.

3.1 Problem formulation

World models aim to learn the dynamics of an environment from raw observations. At test time, the world model is used for planning, serving as a dynamics constraint in an optimization problem.

Let $\mathcal{S} \subset \mathbb{R}^n$ denote the state space and $\mathcal{A} \subset \mathbb{R}^d$ the action space. The environment evolves following a typically unknown dynamics function h such that

$$h : \mathcal{S} \times \mathcal{A} \to \mathcal{S}, \quad s_{t+1} = h(s_t, a_t), \quad \text{for all } t, \tag{1}$$

where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ denote the state and action at time t, respectively. In practice, we only observe a sequence $[o_1, \ldots, o_T]$, where $o_t \in \mathcal{O} \subset \mathbb{R}^p$ is an observation of the underlying state s_t. Our goal is to learn a latent world model consisting of an embedding function $\Phi_\mu : \mathcal{O} \to \mathcal{Z}$, which maps observations to latent representations, and a transition function $f_\theta : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$, which predicts the next latent state given the current latent state and an action, such that

$$z_t = \Phi_\mu(o_t), \quad z_{t+1} = f_\theta(z_t, a_t), \quad \text{for all } t. \tag{2}$$

Encoder. Following Zhou et al. (2025), we use a pre-trained encoder as our embedding function Φ_μ. The encoder has a rich feature representation pretrained on a variety of visual domains, enabling the latent world model to be robust across a variety of tasks.

Transition model. We train the transition function of the environment using the following teacher-forcing objective:

$$\min_\theta \mathbb{E}_{(o_t, a_t, o_{t+1})} \| f_\theta(\Phi_\mu(o_t), a_t) - \Phi_\mu(o_{t+1}) \|_2^2. \tag{3}$$

The triplets (o_t, a_t, o_{t+1}) are sampled from a dataset of trajectories, and we are minimizing the distance between the true and predicted embeddings of the next state o_{t+1}.

Planning. The planning objective is defined at test-time. Given an initial state $z_1 \in \mathcal{Z}$ and a goal state $z_{\text{goal}} \in \mathcal{Z}$, the planning task is to find a sequence of actions $\{\hat{a}_t^*\}_{t=1}^H$ that drives the system to the goal. Formally, we solve

$$\{\hat{a}_t^*\}_{t=1}^H = \arg\min_{\{\hat{a}_t\}} \| \hat{z}_{H+1} - z_{\text{goal}} \|_2^2 \tag{4}$$

where the latent trajectory is generated recursively as

$$\hat{z}_2 = f_\theta(z_1, \hat{a}_1), \quad \hat{z}_{t+1} = f_\theta(\hat{z}_t, \hat{a}_t) \quad \text{for} \quad t > 1. \tag{5}$$

This recursive procedure is encapsulated with the function $\text{rollout}_f : \mathcal{Z} \times \mathcal{A}^H \to \mathcal{Z}^H$. Gradient-based planning (GBP) solves the planning task via minimizing the loss function $\| \hat{z}_{H+1} - z_{\text{goal}} \|_2^2$ with respect to the sequence of actions $\{\hat{a}_t\}$ via gradient descent. Crucially, since the world model is differentiable, $\nabla_{\{\hat{a}_t\}} \hat{z}_{H+1} = \nabla_{\{\hat{a}_t\}} \text{rollout}_f(z_1, \{\hat{a}_t\})_{H+1}$ is well-defined. We detail GBP in

Algorithm 1: Gradient-Based Planning (GBP) via Gradient Descent

Input: Start state z_1, goal state z_{goal}, world model f_θ, horizon H, optimization iterations I
Output: Optimal action sequence $\{\hat{a}_t\}_{t=1}^{H}$

Initialize predicted actions $\{\hat{a}_t\}_{t=1}^{H} \sim \mathcal{N}(0, I_H)$
for $i = 1, \ldots, I$ **do**
  $\hat{z}_{H+1} \leftarrow \text{rollout}_f(z_1, \{\hat{a}_t\})_{H+1}$
  $\mathcal{L}_{\text{goal}} \leftarrow \|\hat{z}_{H+1} - z_{\text{goal}}\|_2^2$
  $\{\hat{a}_t\} \leftarrow \{\hat{a}_t\} - \eta \cdot \nabla_{\{\hat{a}_t\}} \mathcal{L}_{\text{goal}}$
end
return $\{\hat{a}_t\}_{t=1}^{H}$

Algorithm 1. The goal state loss landscape is determined by the world model and thus the success of GBP is dependent on the world model accurately predicting future states regardless of the sequence of predicted actions.

Planning methods like GBP can struggle with long-horizon planning. Model Predictive Control (MPC), a procedure that predicts a horizon of H actions but only takes the first $K \leq H$ actions, can help alleviate this. Each MPC step, the predicted actions are rolled out in the environment simulator whereupon the resulting final state becomes the initial state for the next planning step.

3.2 Online World Modeling

As we illustrate in Figure 2, the actions achieved via GBM may be out-of-distribution for the world model. Indeed, the WM is typically trained on a static distribution of expert trajectories. On the other hand, GBP specifically finds actions that lead the WM to predict a desired target state. It is well-known that such optimization can produce adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014b), and generally will produce out-of-distribution actions for the world model. This effect will produce compounding errors over long-horizon predictions.

To address this issue, we propose Online World Modeling, where we attempt to directly correct such trajectories and finetune the world model on the actions produced via GBP. We illustrate this method in Figure 1.

Algorithm 2: Online World Modeling

Input: Pretrained world model f_θ, simulator dynamics function h, encoder Φ_μ, dataset of trajectories $\mathcal{T}$, iterations N, horizon H, planning iterations I
Output: Updated world model f_θ

Initialize new trajectory dataset $\mathcal{T}'$
for $i = 1, \ldots, N$ **do**
  Sample trajectory $\tau_i = (z_1, a_1, z_2, a_2, \ldots, a_H, z_{H+1}) \sim \mathcal{T}$
  $\{\hat{a}_t\}_{t=1}^{H} \leftarrow \text{GBP}(z_1, z_{H+1}, p_\theta, H, I)$
  $\{s'_t\}_{t=2}^{H+1} \leftarrow \text{rollout}_h(s_1, \{\hat{a}_t\})$
  $\{z'_t\}_{t=2}^{H+1} \leftarrow \{\Phi_\mu(s'_t)\}_{t=2}^{H+1}$
  $\tau'_i \leftarrow (z_1, \hat{a}_1, z'_2, \hat{a}_2, \ldots, \hat{a}_H, z'_{H+1})$
  $\mathcal{T}' \leftarrow \mathcal{T}' \cup \tau'_i$
  Train f_θ on next-state prediction using $\mathcal{T}'$
end
return f_θ

First, we conduct GBP using the first and last latent states of an expert trajectory τ. This way, we produce a sequence of predicted actions $\{\hat{a}_t\}_{t=1}^{H}$ that might send the world model to regimes in the latent state space outside of the training distribution. Then, we obtain a *corrected trajectory*: the actual states that would be achieved by taking the actions $\{\hat{a}_t\}_{t=1}^{H}$ in the environment. Formally, the corrected trajectory is obtained via a rollout using the true transition function h: $\text{rollout}_h(\{\hat{a}_t\}_{t=1}^{H}, z_1)$.

The corrected trajectory,

$$\tau' = (z_1, \hat{a}_1, z'_2, \hat{a}_2, \ldots, z'_{H+1}) \tag{6}$$

is added to the dataset that the world model trains with every time the dataset is updated. Repeatedly training on the corrected trajectories ensures that the world model's behavior is adequately adjusted. We provide more detail in Algorithm 2.

Online world modeling is reminiscent of the DAgger (Dataset Aggregation) method (Ross et al., 2011a), an online imitation learning method wherein a base policy network is iteratively trained on its own rollouts with the action predictions replaced by those from an expert policy. In a similar spirit, we invoke the ground-truth simulator as our expert world model that we intend to imitate.

3.3 Adversarial World Modeling

Rather than only updating the world model on trajectories encountered during planning, we propose an adversarial training method to train on latent states where the world model is anticipated to perform poorly, without conducting planning. These adversarial samples may lie outside the expert trajectory distribution that the world model was originally trained on, so our adversarial training algorithm promotes robustness to these regions of latent space during planning.

Adversarial training (Madry et al., 2018) promotes models to be robust to adversarial attacks. Specifically, an adversarial attack is the application of a perturbation δ that maximally degrades the prediction of a model f_θ. Thus, the robust optimization objective is

$$\min_\theta \sum_i \max_{\delta \in \Delta} \mathcal{L}(f_\theta(x_i + \delta, y_i)), \tag{7}$$

where $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$ because adversarial examples should be mostly indistinguishable from the original dataset. The constraint on the magnitude of δ supports the use of the Fast Gradient Sign Method (Goodfellow et al., 2014a) which approximately solves the inner maximization problem in 7 in one step using $\delta^* = \epsilon \text{sign}(\nabla_x \mathcal{L}(f_\theta(x_i, y_i))$. Wong et al. (2020) shows that when combined with a random initialization of the perturbation, this one-step approximation of the maximally adversarial perturbation is as effective as and significantly cheaper than an iterative projected gradient descent method. This enables us to generate adversarial samples over entire large-scale offline imitation learning datasets.

Algorithm 3: Adversarial World Modeling

Input: Pretrained world model f_θ, dataset of trajectories $\mathcal{T}$, action perturbation scaling λ_a, state perturbation scaling λ_z, iterations N, minibatch size B, horizon H

Output: Updated world model f_θ

Initialize new trajectory dataset $\mathcal{T}'$
for $i = 1, \ldots, N$ **do**
  Sample minibatch $\tau_i = \{(z_1^j, a_1^j, z_2^j, a_2^j, \ldots, a_H^j, z_{H+1}^j)\}_{j=1}^B \sim \mathcal{T}$
  $\epsilon_a = \sigma(\{a_1^j \ldots a_H^j\})$
  $\epsilon_z = \sigma(\{z_1^j \ldots z_{H+1}^j\})$
  $\delta_a \sim U[-\epsilon_a, \epsilon_a]$
  $\delta_z \sim U[-\epsilon_z, \epsilon_z]$
  for $t = 1, \ldots, H$ **do**
    $\nabla_{\delta_a}, \nabla_{\delta_z} \leftarrow \nabla_{\delta_a, \delta_z} \|p_\theta(z_t + \delta_z, a_t + \delta_a) - z_{t+1}\|_2^2$
    $\delta_a \leftarrow \text{clip}(\delta_a + \alpha_a \nabla_{\delta_a}, -\lambda_a \epsilon_a, \lambda_a \epsilon_a)$
    $\delta_s \leftarrow \text{clip}(\delta_s + \alpha_z \nabla_{\delta_z}, -\lambda_z \epsilon_z, \lambda_z \epsilon_z)$
    $a'_t \leftarrow a_t + \delta_a$
    $z'_t \leftarrow z_t + \delta_z$
  end
  $\mathcal{T}' \leftarrow \mathcal{T}' \cup \tau_i$
  Train f_θ on next-state prediction using $\mathcal{T}'$
end
return f_θ

In training on these adversarial trajectories, we encourage the world model to become more robust to unseen states during GBP. Let ϵ_a denote the radius of the perturbation to the actions $\{a_t\}$ and ϵ_z denote the radius of perturbation to the latent states $\{z_t\}$. For each state-action pair in a given minibatch, we approximate the maximally adversarial perturbation

$$\nabla_{\delta_a}, \nabla_{\delta_z} = \nabla_{\delta_a, \delta_z} \|p_\theta(z_t + \delta_z, a_t + \delta_a) - z_{t+1}\|_2^2$$
$$\delta_a = \delta_a + \alpha_a \nabla_{\delta_a}, \quad \delta_z = \delta_z + \alpha_z \nabla_{\delta_z}$$
$$\alpha_a = 1.25\epsilon_a, \quad \alpha_z = 1.25\epsilon_z$$

with step sizes α_a, α_z. We find that adaptively setting our perturbation radius ϵ to the standard deviation of the current minibatch and applying distinct scaling factors $\lambda_a = 0.05, \lambda_z = 0.02$ provides stability during training. Lastly, we clip our computed perturbation to within $[-\epsilon, \epsilon]$. We provide more detail in Algorithm 3 and illustrate the intuition in Figure 2.

Practically, Adversarial World Modeling has several advantages over Online World Modeling. Firstly, it does not require simulating new trajectories during training, which is challenging in many real-world applications. Moreover, adversarial training has been shown to make the model smoother, which can make the optimization problem in the planning phase easier to solve (Mejia et al., 2019).

4 EXPERIMENTS

We evaluate the performance of our methods using pretrained world models from DINO World Model (Zhou et al., 2025) on 3 tasks: PushT, Point-Maze, and Wall. These tasks are adopted from DINO-WM. In terms of success rate, we show improvements to GBP across all 3 tasks and even outperform the more expensive CEM on PushT. We also demonstrate that world models finetuned with our methods are better able to emulate the simulator during planning.

We evaluate our method on the task of driving a system from an initial configuration o_1 to a target configuration o_{goal}, both specified as observations in $\mathcal{O}$. We report planning results, both in Open-Loop and in MPC, in Table 1. In the open-loop setup, we run Algorithm 1 with the initial state $\Phi_\mu(o'_1)$ and evaluate the predicted actions. In the MPC setup, we run Algorithm 1 once for each MPC step (using $\Phi_\mu(o'_1)$ as the initial state for the first MPC step), rollout the predicted actions $\{\hat{a}_t\}$ in the environment simulator to reach latent state $\hat{z}_{H+1}$, and set $\hat{z}_1 = \hat{z}_{H+1}$ for the next MPC iteration.

We adopt the latent world model framework from DINO World Model (DINO-WM) (Zhou et al., 2025). The embedding function Φ_μ is implemented using the pre-trained DINOv2 encoder introduced in (Oquab et al., 2024) and remains frozen while finetuning the transition model f_θ. f_θ is implemented using the ViT architecture introduced in (Dosovitskiy et al., 2021). We use a VQVAE decoder (van den Oord et al., 2018) to visualize latent states. We train a function $g_\theta : \mathcal{Z} \times \mathcal{Z} \to \mathcal{A}^T, g_\theta(z_1, z_g) = \{\hat{a}_t\}$ to initialize a sequence of actions for gradient-based planning. In practice, g_θ is a convolutional encoder. We analyze the impact of including this initialization network g_θ in Table 1.

Optimizer. Motivated by the observation that Gradient-Based Planning via Gradient Descent typically performs poorly despite low world model loss, we hypothesize that the optimization landscape is rugged. To avoid local minima, we additionally experiment with the Adam optimizer Kingma and Ba (2014) with a learning rate of 0.3 in place of Gradient Descent.

4.1 RESULTS

Across the board, we outperform Gradient-Based Planning on models trained purely via next-state prediction. In the open-loop setting, we achieve +50% on Push-T, +20% on PointMaze, and +22% in Wall over Gradient Descent. Moreover, for the Push-T environment, we outperform CEM by 4% SR.

Gradient Descent vs. Adam. We find that Adam almost always yields higher planning performance over Gradient Descent, and we hypothesize that this is due to its ability to traverse a more complex optimization landscape.

		Push-T					PointMaze					Wall				
World Model	**Planner**	GD	GD†	Ad	Ad†	CEM	GD	GD†	Ad	Ad†	CEM	GD	GD†	Ad	Ad†	CEM
Teacher Forcing	Open-Loop	40	44	54	62	86	16	16	18	14	**80**	8	4	16	12	**74**
	MPC	62	60	76	84	—	46	40	52	54	—	6	6	48	32	—
+ *Online*	Open-Loop	34	56	52	66	—	18	8	20	28	—	18	10	30	18	—
	MPC	56	52	76	82	—	**52**	40	**68**	46	—	34	2	46	22	—
+ *Adversarial*	Open-Loop	66	**74**	82	**90**	—	12	22	30	36	—	18	18	24	20	—
	MPC	**76**	74	**92**	90	—	52	**44**	56	**56**	—	**58**	**24**	**68**	**48**	—

Table 1: **Gradient-Based Planning**: We evaluate the performance on 3 tasks of 3 world models in both open-loop and MPC frameworks. *Teacher Forcing*, *Teacher Forcing + Online*, and *Teacher Forcing + Adversarial* are trained following Sec. 3.1, Sec. 3.2 and Sec. 3.3 respectively. For each task, we apply Gradient Descent (GD) and Adam (Ad) while our initial action sequence is either sampled from a normal distribution or inferred using our initialization network g_θ (denoted via $\dagger$). For GD and Ad, we report open-loop success rates (SR) after 300 optimization steps, and MPC SR after 10 MPC iterations, each with 100 and 300 optimization steps respectively for Ad and GD. Finally, for the Wall task, we use AdamW as an optimizer as we empirically observe that action gradients are unstable and do not converge otherwise.

Random Initialization vs. Initialization Network. As the initialization network has been trained on the direct action planning task, it should provide a conducive initialization for GBP. Indeed, we observe that for the Push-T task, using this initialization provides a consistent boost. However, for the PointMaze and Wall tasks, we find that the initialization is often quite poor. For example, the decrease from 34 $\rightarrow$ 2 SR when applying the initialization network in the Wall environment may indicate that the initialization is a local minima that Gradient Descent is unable to escape. Furthermore, we hypothesize that since our initialization network is trained fully offline, it fails to generalize to unseen goal targets at test-time.

Online vs. Adversarial. While both Online World Modeling and Adversarial World modeling bootstrap new data to improve the robustness of our world model at GBP-time, the distributions they induce are quite different. Whereas Online World Modeling anticipates and covers the distribution seen at planning time, Adversarial World Modeling exploits the current loss landscape of the world model to encourage local smoothness near expert trajectories. For most settings, we find that Adversarial World Modeling outperforms Online World Modeling, with the exception of PointMaze. We hypothesize this is due to the PointMaze task benefitting from search. Motivated by this result, we hypothesize that these two objectives balance different concerns — the former compounding errors over world model rollouts (impacting the accuracy of predictions) and the latter in optimization (impacting the accuracy of actions between two states).

4.2 Simulator Deviation

In order to determine whether our methods are able to bridge the gap between the performance of actions from rolling out with the simulator compared to the world model, we measure their relative difference over the course of planning. Given a predicted action sequence $\{\hat{a}_t\}_{t=1}^H$ at iteration i of GBP, we compute a relative loss difference

$$\frac{\|\Phi_\mu(\text{rollout}_h(s_1, \{\hat{a}_t\}_{t=1}^H)) - z_{H+1}\|_2^2 - \|\text{rollout}_f(z_1, \{\hat{a}_t\}_{t=1}^H) - z_{H+1}\|_2^2}{\|\text{rollout}_f(z_1, \{\hat{a}_t\}_{t=1}^H) - z_{H+1}\|_2^2}. \tag{8}$$

When this metric is low, it indicates that the world model has an adherence to the simulator during planning. Results for this relative difference metric on the PushT task over the course of Open-Loop GBP are reported in Figure 3. We see that both Adversarial World Modeling and Online World Modeling perform successfully reduce the gap between world model and simulator. The greater adherence in the Online World Modeling case can be attributed to the direct correction of predicted states.

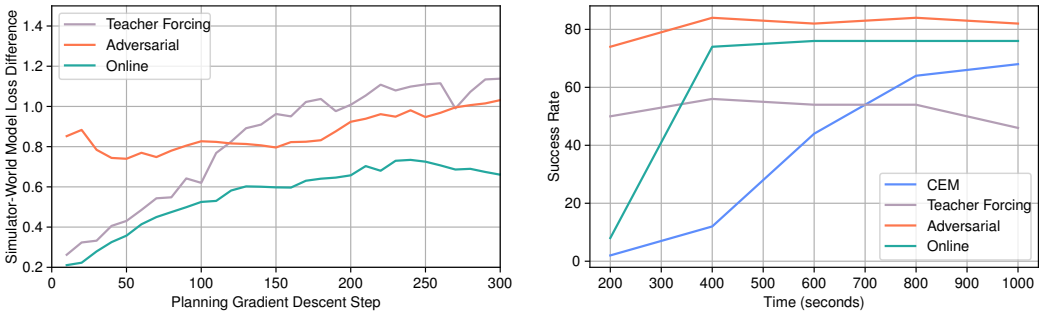

Figure 3: **Left:** The relative difference in latent goal state loss between rolling out in the simulator versus rolling out with the world model. This is with naive GBP on the PushT task. We highlight that both our methods reduced this deviation. **Right:** SR over wall clock time excluding simulator computations. GBP and our methods achieve significantly greater performance in a fraction of the time.

4.3 Test-Time Efficiency

When using a world model to conduct planning in real world environments, fast inference is crucial for actively interacting with an environment. In Figure 3, we show that within 200 seconds, our training methods coupled with test-time GBP outperform the CEM baseline after 1000 seconds. This speed allows GBP to be a more viable algorithm for real-time planning.

5 Conclusion

In this work, we present Online World Modeling and Adversarial World Modeling as techniques to close the train-test gap that emerges between training world models on next-state prediction and using an iterative planning process to take action. By demonstrating improved performance on GBP and occasionally surpassing CEM, we hope GBP can be more adopted for planning with world models. Future directions for this work are twofold. Firstly, we seek to extend our method to directly improve the quality of action gradients during planning. Secondly, evaluating the robustness of our methods to long-horizon planning will be crucial for practical usage.

Limitations. Our work assumes having access to offline datasets with representative state-action data, which may not be available for complex environments in which simulators are not provided. Likewise, our Online World Modeling algorithm relies on the presence of a computationally efficient environment simulator, which may be infeasible for real-world tasks.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauzá, L. Kaelbling, J. Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2018. URL https://api.semanticscholar.org/CorpusId:51954048.

Adil Kaan Akan and F. Güney. Stretchbev: Stretching future instance prediction spatially and temporally. In *European Conference on Computer Vision*, 2022. URL https://api.semanticscholar.org/CorpusId:247749011.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. 2023.

Adrien Bardes, Q. Garrido, Jean Ponce, Xinlei Chen, Michael G. Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *ArXiv*, abs/2404.08471, 2024. URL https://api.semanticscholar.org/CorpusId:269137489.

Homanga Bharadhwaj, Kevin Xie, and F. Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. *ArXiv*, abs/2004.08763, 2020. URL https://api.semanticscholar.org/CorpusId:215827996.

Oumayma Bounou, Jean Ponce, and Justin Carpentier. Online learning and control of complex dynamical systems from sensory input. *Advances in Neural Information Processing Systems*, 34: 27852–27864, 2021.

Siwei Chen, Yiqing Xu, Cunjun Yu, Linfeng Li, Xiao Ma, Zhongwen Xu, and David Hsu. Benchmarking deformable object manipulation with differentiable physics. *ArXiv*, abs/2210.13066, 2022. URL https://api.semanticscholar.org/CorpusId:257482484.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video representation learning with joint-embedding predictive architectures. *ArXiv*, abs/2412.10925, 2024. URL https://api.semanticscholar.org/CorpusId:274777110.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and C. Isbell. Imitating latent policies from observation. *ArXiv*, abs/1805.07914, 2018. URL https://api.semanticscholar.org/CorpusId:29156793.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793, 2016. URL https://api.semanticscholar.org/CorpusID:2780699.

Chelsea Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. *ArXiv*, abs/1605.07157, 2016. URL https://arxiv.org/pdf/1605.07157.pdf.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.

Herbert Goldstein, Charles P Poole, and John Safko. *Classical mechanics*, volume 2. Addison-wesley Reading, MA, 1950.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Yanchen Guan, Haicheng Liao, Zhenning Li, Guohui Zhang, and Chengzhong Xu. World models for autonomous driving: An initial survey. *ArXiv*, abs/2403.02622, 2024. URL https://api.semanticscholar.org/CorpusId:268249117.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

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

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, R. Cipolla, and J. Shotton. Model-based imitation learning for urban driving. *ArXiv*, abs/2210.07729, 2022. URL https://api.semanticscholar.org/CorpusId:252907712.

Matthew Jackson, Michael Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and J. Foerster. Policy-guided diffusion. *ArXiv*, abs/2404.06356, 2024. URL https://api.semanticscholar.org/CorpusId:269009692.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, K. Czechowski, D. Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, G. Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. *ArXiv*, abs/1903.00374, 2019. URL https://api.semanticscholar.org/CorpusID:67856232.

Maximilian Karl, Maximilian Sölch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *ArXiv*, abs/1605.06432, 2016. URL https://api.semanticscholar.org/CorpusID:14992224.

Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Dino-foresight: Looking into the future with dino. *arXiv preprint arXiv:2412.11673*, 2024.

Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 2002.

Nan Rosemary Ke, Amanpreet Singh, Ahmed Touati, Anirudh Goyal, Yoshua Bengio, Devi Parikh, and Dhruv Batra. Learning dynamics model in reinforcement learning by incorporating the long term future. *ArXiv*, abs/1903.01599, 2019. URL https://api.semanticscholar.org/CorpusId:67877018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Ian Lenz, Ross A Knepper, and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, volume 10, page 25. Rome, Italy, 2015.

Weiwei Li and Emanuel Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *First International Conference on Informatics in Control, Automation and Robotics*, volume 2, pages 222–229. SciTePress, 2004.

Alessandro Macchelli, Claudio Melchiorri, and Stefano Stramigioli. Port-based modeling and simulation of mechanical systems with rigid and flexible links. *IEEE transactions on robotics*, 25(5): 1016–1029, 2009.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2018.

David Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.

Felipe A Mejia, Paul Gamble, Zigfried Hampel-Arias, Michael Lomnitz, Nina Lopatina, Lucas Tindall, and Maria Alejandra Barrios. Robust or private? adversarial training makes models more vulnerable to privacy attacks. *arXiv preprint arXiv:1906.06449*, 2019.

MG Mohanan and Ambuja Salgoankar. A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems*, 100:171–185, 2018.

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)*, pages 7559–7566. IEEE, 2018.

Anusha Nagabandi, K. Konolige, S. Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, 2019. URL https://arxiv.org/pdf/1909.11652.pdf.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011a.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011b.

Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

Robin Schiewer, Anand Subramoney, and Laurenz Wiskott. Exploring the limits of hierarchical world models in reinforcement learning. *Scientific Reports*, 14, 2024. URL https://api.semanticscholar.org/CorpusId:270214472.

Michael D. Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324:81 – 85, 2009. URL https://doi.org/10.1126/science.1165893.

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

Archit Sharma, Shixiang Shane Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *ArXiv*, abs/1907.01657, 2019. URL https://api.semanticscholar.org/CorpusID:195791369.

Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *ArXiv*, abs/2205.02909, 2022. URL https://api.semanticscholar.org/CorpusID:248562698.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer, 2009.

Mark W Spong, Seth Hutchinson, and M Vidyasagar. Robot modeling and control. *John Wiley &amp*, 2020.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Jyothir SV, Siddhartha Jalagam, Yann LeCun, and Vlad Sobal. Gradient-based planning with world models. *arXiv preprint arXiv:2312.17227*, 2023.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Erik Talvitie. Model regularization for stable sample rollouts. In *Conference on Uncertainty in Artificial Intelligence*, 2014. URL https://dslpitt.org/uai/displayArticleDetails.jsp?article_id=2514&mmnu=1&proceeding_id=30&smnu=2.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

Tsun-Hsuan Wang, Pingchuan Ma, A. Spielberg, Zhou Xian, Hao Zhang, J. Tenenbaum, D. Rus, and Chuang Gan. Softzoo: A soft robot co-design benchmark for locomotion in diverse environments. *ArXiv*, abs/2303.09555, 2023. URL https://api.semanticscholar.org/CorpusId:257557557.

Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017a.

Grady Williams, Nolan Wagener, Brian Goldfain, P. Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic mpc for model-based reinforcement learning. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017b. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7989202.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

Jie Xu, Viktor Makoviychuk, Yashraj S. Narang, Fabio Ramos, W. Matusik, Animesh Garg, and M. Macklin. Accelerated policy learning with parallel differentiable simulation. *ArXiv*, abs/2204.07137, 2022. URL https://api.semanticscholar.org/CorpusId:248178090.

Xianyuan Zhan, Xiangyu Zhu, and Haoran Xu. Model-based offline planning with trajectory pruning. In *International Joint Conference on Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusId:234742314.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, pages 7444–7453. PMLR, 2019.

Wenbo Zhang, Karl Schmeckpeper, P. Chaudhari, and Kostas Daniilidis. Deformable linear object prediction using locally linear latent dynamics. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13503–13509, 2021. URL https://api.semanticscholar.org/CorpusId:232380092.

Zongyuan Zhang, Tian dong Duan, Zheng Lin, Dong Huang, Zihan Fang, Zekai Sun, Ling Xiong, Hongbin Liang, Heming Cui, and Yong Cui. State-aware perturbation optimization for robust deep reinforcement learning. *ArXiv*, abs/2503.20613, 2025. URL https://api.semanticscholar.org/CorpusId:277322758.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2025.

Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Yong Yu, and Weinan Zhang. Diffusion models for reinforcement learning: A survey. *ArXiv*, abs/2311.01223, 2023. URL https://api.semanticscholar.org/CorpusId:264935559.