PARALLEL STOCHASTIC GRADIENT-BASED PLANNING FOR WORLD MODELS

Michael Psenka[1,2,◦], **Michael Rabbat**[2], **Aditi Krishnapriyan**[1], **Yann LeCun**[2,3,*], **Amir Bar**[2,*]
[1]University of California, Berkeley [2]Meta FAIR [3]New York University
*Equal advising. ◦Work done at Meta.

ABSTRACT

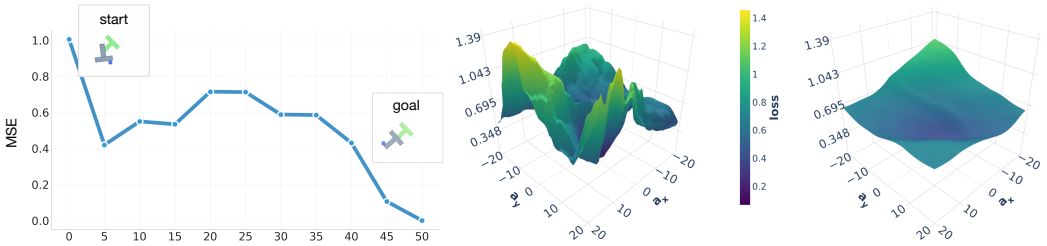
World models simulate environment dynamics from raw sensory inputs like video. However, using them for planning can be challenging due to the vast and unstructured search space. We propose a robust and highly parallelizable planner that leverages the differentiability of the learned world model for efficient optimization, solving long-horizon control tasks from visual input. Our method treats states as optimization variables ("virtual states") with soft dynamics constraints, enabling parallel computation and easier optimization. To facilitate exploration and avoid local optima, we introduce stochasticity into the states. To mitigate sensitive gradients through high-dimensional vision-based world models, we modify the gradient structure to descend towards valid plans while only requiring action-input gradients. Our planner, which we call **GRASP** (Gradient RelAxed Stochastic Planner), can be viewed as a stochastic version of a non-condensed or collocation-based optimal controller. We provide theoretical justification and experiments on video-based world models, where our resulting planner outperforms existing planning algorithms like the cross-entropy method (CEM) and vanilla gradient-based optimization (GD) on long-horizon experiments, both in success rate and time to convergence.

1 INTRODUCTION

Intelligent agents carry a small-scale model of external reality which allows them to simulate actions, reason about their consequences, and choose the ones that lead to the best outcome. Attempts to build such models date back to control theory, and in recent years researchers have made progress in building world models using deep neural networks trained directly from the raw sensory input (e.g., vision). For example, recent world models have shown success modeling computer games (Valevski et al., 2024), navigating real-world environments (Bar et al., 2025; Ball et al., 2025), and robot arm motion commands (Goswami et al., 2025). World models have numerous impactful applications from simulating complex medical procedures (Koju et al., 2025) to testing robots in visually realistic environments (Guo et al., 2025).

Current planning algorithms used with world models often rely on 0th-order optimization methods such as the Cross-Entropy Method (CEM Rubinstein & Kroese (2004)) or in general *shooting methods* which plan by iteratively rolling out trajectories and choosing actions from the optimal rollout (Bock & Plitt, 1984; Piovesan & Tanner, 2009). These approaches are simple and robust, but their performance degrades with longer planning horizons and higher action dimensionality (Bharadhwaj et al., 2020), motivating the use of gradient information when differentiable world models are available.

Gradient-based planners exploit the differentiability of learned world models to directly optimize action sequences, enabling more sample-efficient planning and finer-grained improvement than zero-order methods. However, there are two main challenges to this optimization approach: local minima (Jyothir et al., 2023) and instability due to differentiating through the full rollout, akin to back-propagation through time (Werbos, 2002). While prior methods have formulated the optimization for better conditioning (e.g., multiple shooting and direct collocation (Von Stryk, 1993)), these techniques are typically developed for known dynamical systems and do not scale as well to deep neural dynamics (often in latent spaces) with brittle or poorly calibrated Jacobians.



(a) Similarity Loss Across Time (b) Loss Landscape (GD) (c) Loss Landscape (ours)

Figure 1: *Difficulty of the planning problem*. Subfigure **(a)** shows the distance to the goal in L^2 norm throughout a successful trajectory. This illustrates the difficulty of planning optimization away from a minimizer: successful trajectories often have to first move away from the goal to successfully plan towards it later, resulting in greedy strategies failing. Subfigures **(b)-(c)** depict the loss landscape at convergence of standard rollout-based planners vs. our planner. The example given is in the Push-T environment at horizon length 50. The axes plotted over are with respect to two random, orthogonal, unit-norm directions in the full action space $\mathbb{R}^{50\times 2}$. Our planner loss is taken as in Eq. (7), and for GD the loss is taken as in Eq. (1).

In this work, we introduce a novel gradient-based planning method for learned world models that decouples temporal dynamics into parallel optimized states (rather than serial rollouts) while remaining robust at long horizons and in high-dimensional state spaces. Rather than planning exclusively through a deep, sequential rollout of the dynamics model, our approach optimizes over *lifted intermediate states* that are treated as independent optimization variables. Our approach makes two fundamental additions that help solve issues for gradient-based planning in higher dimensions: gradient sensitivity and local minima.

Firstly, a fundamental difficulty arises in the setting of vision-based world models. In high-dimensional learned state spaces, gradients with respect to state inputs can be brittle or adversarial, allowing the optimizer to exploit sensitive Jacobian structure rather than discovering physically meaningful transitions. To mitigate this issue, our planner deliberately stops gradients through the state inputs of the world model, while retaining gradients with respect to actions, which we find behave more reasonably. This alone would promote trajectories near the starting state, but with a dense one-step goal loss over the full trajectory, converged trajectories for these noisy iterations tend towards the goal.

Secondly, to address the remaining non-convexity of the lifted state approach, our planner incorporates Langevin-style stochastic updates on the lifted state variables, explicitly injecting noise during optimization to promote exploration of the state space and facilitate escape from unfavorable basins. This stochastic relaxation allows the planner to search over diverse intermediate trajectories while still favoring solutions that approximately satisfy the learned dynamics. Finally, we intermittently apply a small GD step to fine-tune stochastically optimized trajectories towards fully-optimized paths.

Together, these components yield a practical gradient-based planner for learned visual dynamics that remains stable at long horizons while avoiding the failure modes commonly encountered when back-propagating through deep world-model rollouts. We call our planner **GRASP** (Gradient RelAxed Stochastic Planner) to emphasize its primary components: using gradient information, relaxing the dynamics constraints, and stochastic optimization for exploration. In various settings, we achieve up to +10% success rate at less than half the compute time cost. We also provide a theoretical model for our planner to further illustrate its role. We demonstrate our planner on visual world models trained on problems in D4RL (Fu et al., 2020) and the DeepMind control suite (Tassa et al., 2018).

2 Problem Formulation

Our main object of interest is a learned world model $F_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ that predicts the next state given the current state and action. For visual domains, states are typically represented in a learned

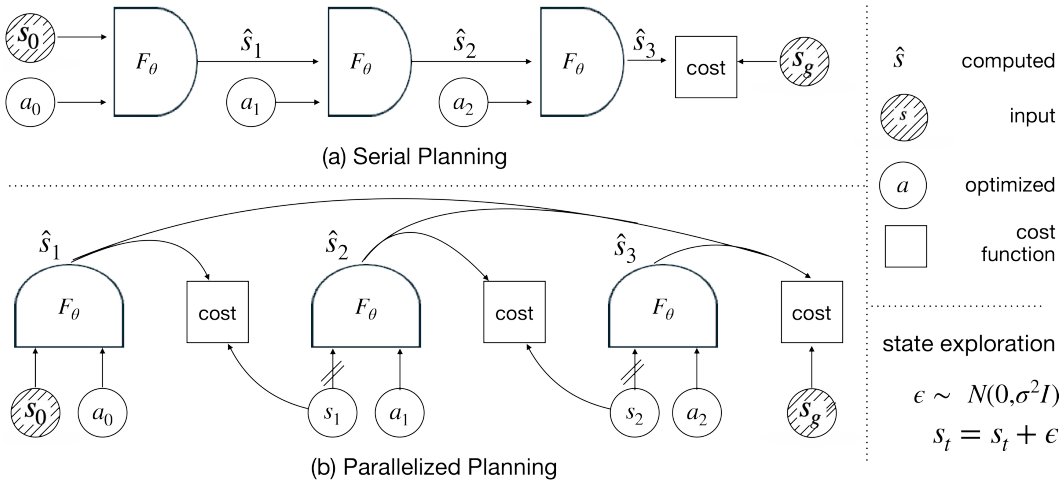


Figure 2: Graphical depiction of **(a)** a standard serial-based setup for optimization-based planning, where states are rolled out using the actions and the loss is evaluated on the goal state, **(b)** our setup, which parallelizes the world model evaluations by optimizing "virtual states" directly and only supervising pairwise dynamics satisfaction. The crossed lines for our method's depiction **(b)** denote stopped gradients and are detailed in Section 3.3, where dense goal losses keep the full planning graph connected while not requiring state gradients of the dynamics F_θ. For our planner, we find it helpful to alternate between **(a)** and **(b)** throughout the planning optimization.

latent space to handle high-dimensional observations. Here we assume $\mathcal{A} = \mathbb{R}^k$ is a continuous Euclidean action space.

We consider the problem of fixed-goal path planning: finding an action sequence $\mathbf{a} = (a_0, a_1, \ldots, a_{T-1})$ that, with respect to the dynamics of the world model F_θ and a given initial state $s_0 \in \mathcal{S}$, reaches a set goal state $g \in \mathcal{S}$:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \|\mathcal{F}_\theta^T(\mathbf{a}, s_0) - g\|_2^2. \tag{1}$$

where the map $\mathcal{F}_\theta^T$ is generated recursively through the update rule $s_{t+1} = F_\theta(s_t, a_t)$:

$$\mathcal{F}_\theta^T(\mathbf{a}, s_0) \coloneqq \underbrace{F_\theta(\ldots F_\theta(F_\theta(}_{T \text{ times}} s_0, a_0), a_1), \ldots a_{T-1}). \tag{2}$$

Optimizing Eq. (1) directly is challenging due to two main problems. First, it requires T applications of F_θ (see Eq. 2), which is computationally expensive and difficult to optimize due to the poor conditioning arising from repeated applications of F_θ (see Appendix A.1 for details). Second, it is susceptible to local minima and a jagged loss landscape—see Figure 1. Due to these reasons, existing planners are based on zero-order optimization algorithms like CEM and MPPI (Williams et al., 2016) which are highly stochastic and do not require gradient computation.

3 Decoupling Dynamics for Gradient-Based Planning

We consider planning with a world model F_θ and horizon T. Given an initial state $s_0 \in \mathcal{S}$ and goal state $g \in \mathcal{S}$, we optimize an action sequence $\mathbf{a} = (a_0, \ldots, a_{T-1})$ such that the rolled-out state $s_T(\mathbf{a})$ is as close to g as possible. A standard approach defines a trajectory by rolling out the model, but backpropagating through a deep composition of F_θ can be unstable and ill-conditioned. Following prior work on lifted planning (Tamimi & Li, 2009; Rybkin et al., 2021), we introduce auxiliary states $\mathbf{z} = (z_1, \ldots, z_T)$ and enforce dynamics consistency through a penalty function.

3.1 Parallelized planning

We first want to decouple the states from the explicit rollout outputs. Writing Eq. (1) in terms of each intermediate dynamics condition, we get the following:

$$\min_{\mathbf{s},\mathbf{a}} \mathcal{L}_{\text{dyn}}(\mathbf{s},\mathbf{a}) := \sum_{t=0}^{T-1} \left\| F_\theta(s_t, a_t) - s_{t+1} \right\|_2^2, \qquad \text{with } s_0 \text{ fixed}, \; s_T = g. \tag{3}$$

The minimization in Eq. (3) is equivalent to Eq. (1) in that they share global minimizers.

Immediately, this gives a great benefit in that *all world model evaluations are parallel*; there is no need to do serial rollouts like what is required in Eq. (1). There are however two main issues with optimizing this loss directly:

1. *Local minima.* When optimizing with respect to states, the states might be stuck in an unphysical region; for example, Figure 6 shows a case where states go straight through a barrier. To address this, we propose to use Langevin state updates which promote exploration (see Section 3.2).
2. *World model sensitivity for high-dimensional states.* When optimizing s directly over a higher dimensional space (e.g. vision-based), we observe that the Jacobian $J_s F_\theta(s, a)$ does not necessarily have any low-dimensional or convex structure; in practice, the world model can be easily steered toward outputting any desired output state, as depicted in Figure 3. We address this in Section 3.3 with a reshaping of the descent directions.

We now describe our approach to address these two fundamental problems with lifted-states approaches to planning.

3.2 Exploration via Langevin state updates

Consider the optimization induced by Eq. (3). A standard way to escape spurious basins is to replace deterministic gradient descent on $\mathbf{s}$ with overdamped Langevin dynamics (Gelfand & Mitter, 1991), whose Euler discretization takes the following form:

$$s_t^{k+1} \leftarrow s_t^k - \eta_s \nabla_{s_t} \mathcal{L}_{\text{dyn}}(\mathbf{s}^k, \mathbf{a}^k) + \sigma_{\text{state}} \xi_t^k, \qquad a_t^{k+1} \leftarrow a_t^k - \eta_a \nabla_{a_t} \mathcal{L}_{\text{dyn}}(\mathbf{s}^k, \mathbf{a}^k), \tag{4}$$

where $\xi_t^k \sim \mathcal{N}(0, I)$. Intuitively, the noise allows the iterates to "hop" between nearby basins of the lifted loss landscape, while keeping actions non-noised keeps the sampled trajectories towards reachable trajectories. See Appendix A.3 for a characterization of the sampled distribution.

3.3 Sensitivity to state gradients

In practice, F_θ is learned and can have brittle local geometry. When optimizing Eq. (3) by gradient descent in both $\mathbf{a}$ and $\mathbf{s}$, we observed that gradients with respect to the state inputs, $\nabla_s F_\theta(s, a)$, can be exploited: for any local goal-reaching objective of the form $\arg\min_s \|y - F_\theta(s, a)\|_2^2$, instead of the optimizer learning to find an s on-manifold such that applying the action a leads to end state y, the optimizer can find a nearby ambient $s + \delta$, $\|\delta\|_2 \ll 1$ such that the loss is practically minimized: $y \approx F_\theta(s + \delta, a)$, *regardless of the starting state* s. This is analogous to the adversarial robustness issue of image classifiers (Szegedy et al., 2013; Shamir et al., 2021): high-dimensional input spaces for trained neural networks can have high Lipschitz constants, hindering optimization performance.

(a) Expected behavior (b) Adversarial example

Figure 3: *Sensitivity of state gradient structure.* Examples of three states far away from the goal on the right (either in-distribution or out-of-distribution), such that taking a small step along the gradient $s' = s - \epsilon \nabla_s \mathcal{L}(s), \mathcal{L}(s) = \|F_\theta(s, a = \mathbf{0}) - g\|_2^2$, leads to a nearby state s' that solves the planning problem in a single step: $F_\theta(s', \mathbf{0}) = g$. Thus, optimizing states directly through the world model F_θ can be quite challenging.

Unfortunately, any loss function over $\mathbf{s}$ and $\mathbf{a}$ whose minimizers are feasible dynamics must depend on the state gradient $\nabla_s F$ in a meaningful way. We provide the informal theorem here, with formalization and proof in Appendix A.4.

Theorem 1 (informal). *A differentiable loss function over state/action trajectories $\mathcal{L} : \mathcal{S}^T \times \mathcal{A}^T \to \mathbb{R}$ given a world model $F_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ cannot satisfy both of the following at the same time:*

1. *Minimizers of $\mathcal{L}$ correspond to dynamically feasible trajectories: $F_\theta(s_t, a_t) = s_{t+1}$,*
2. *$\mathcal{L}$ is insensitive to the world model state gradient $\nabla_s F_\theta$.*

To address this adversarial sensitivity, we detach gradients through the *state inputs* of the world model, while still differentiating with respect to the actions. We denote by $\bar{s}_t$ a stop-gradient copy of s_t (i.e., $\bar{s}_t = s_t$ in value, but treated as constant during differentiation).

Grad-cut dynamics loss. We begin by applying a gradient stop to the state inputs in the dynamics loss:

$$\mathcal{L}^{\text{sg}}_{\text{dyn}}(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{T-1} \left\| F_\theta(\bar{s}_t, a_t) - s_{t+1} \right\|_2^2. \tag{5}$$

This objective is differentiable with respect to $\mathbf{a}$ and the *next* states s_{t+1}, but does not backpropagate through s_t via F_θ.

Require: Initial observation o_0, goal o_g, world model F_θ, horizon T, steps K, learning rates η_a, η_s.
Ensure: $\mathbf{a}^*$
$s_0 \leftarrow \text{encode}(o_0)$, $s_T \leftarrow \text{encode}(o_g)$
$\mathbf{a} \leftarrow \mathbf{a}^0$; $\mathbf{s} \leftarrow \text{init_states}(s_0, s_T)$
for $k = 0$ **to** $K - 1$ **do**
  Compute $\mathcal{L}$ as in Eq. (7)
  Joint step: $(\mathbf{a}, \mathbf{s}) \leftarrow (\mathbf{a}, \mathbf{s}) - (\eta_a \nabla_{\mathbf{a}} \mathcal{L}, \eta_s \nabla_{\mathbf{s}} \mathcal{L})$
  Stochastic state: $s_t \leftarrow s_t + \sigma_{\text{state}} \xi_t, \xi_t \sim \mathcal{N}(0, I)$ for $t = 1, \ldots, T-1$
  Sync (periodic): if $k \bmod K_{\text{sync}} = 0$, rollout from s_0; $s_{t+1} \leftarrow F_\theta(s_t, a_t)$ for $t = 0, \ldots, T-1$
  then take a GD step $\mathbf{a} \leftarrow \mathbf{a} - \eta_{\text{sync}} \nabla_{\mathbf{a}} \|s_T - g\|_2^2$
end for
return $\mathbf{a}^* \leftarrow \mathbf{a}$

Algorithm 1: GRASP planner; parallel stochastic gradient-based planning

Dense goal shaping on one-step predictions. While Eq. (5) improves robustness, it introduces a new degeneracy: paths gravitate towards the current rollout, regardless of proximity to the goal. To provide a task-aligned signal at every time step without state-input gradients, we add a goal loss on the one-step predictions:

$$\mathcal{L}^{\text{sg}}_{\text{goal}}(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{T-1} \left\| F_\theta(\bar{s}_t, a_t) - g \right\|_2^2. \tag{6}$$

This encourages each predicted next state to move toward the goal, supplying gradient information to every action a_t while maintaining the grad-cut on s_t. This is depicted visually in Figure 2, and theoretically in Appendix A.4. Crucially, due to the stop-gradient $\bar{s}_t$, gradients through $F_\theta(\bar{s}_t, a_t)$ flow only with respect to a_t (and not s_t), which prevents the optimizer from exploiting adversarial state-input directions. The final energy that is sampled from is then the following:

$$\mathcal{L}(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{T-1} \left\| F_\theta(\bar{s}_t, a_t) - s_{t+1} \right\|_2^2 + \gamma \sum_{t=0}^{T-1} \left\| F_\theta(\bar{s}_t, a_t) - g \right\|_2^2, \tag{7}$$

where $\gamma > 0$ is fixed.

Resulting noisy dynamics. The final resulting dynamics, after explicitly writing out $\nabla_a \mathcal{L}$, are as follows:

$$s_t^{k+1} \leftarrow s_t^k - 2\eta_s \Big(s_t - F_\theta(s_{t-1}, a_{t-1}) \Big) + \sigma_{\text{state}} \xi_t^k, \qquad a_t^{k+1} \leftarrow a_t^k - \eta_a \nabla_{a_t} \mathcal{L}(\mathbf{s}^k, \mathbf{a}^k), \tag{8}$$

where $\xi_t^k \sim \mathcal{N}(0, I)$. Importantly, while the action dynamics still follow a gradient flow, the states do not follow a true gradient vector field, and thus *the resulting dynamics are not Langevin.* What results are still noisy dynamics that bias towards valid goal-oriented trajectories, but whose efficiency will require an extra synchronization step as described in the following section.

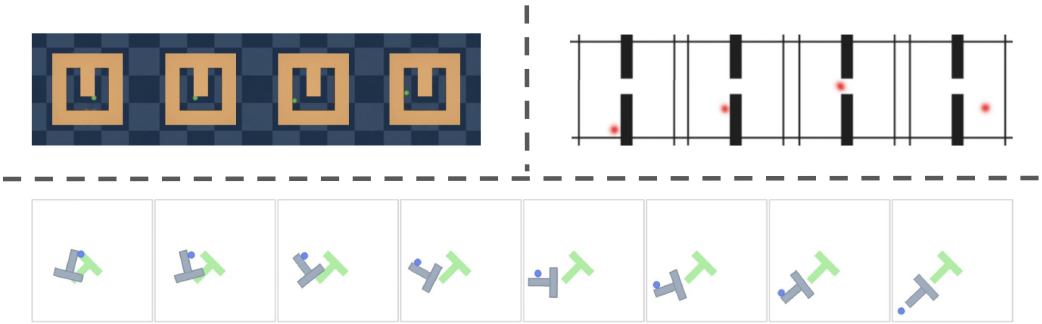


Figure 4: *Virtual states learned through planning.* All examples are instantiations of our planner at horizon 50 in the Point-Maze, Wall-Single, and Push-T environments. Directly optimized states find realistic, non-greedy paths towards the goal.

3.4 Full-rollout synchronization

The no-state-gradient updates are designed to be robust to brittle state-input Jacobians of the learned world model. However, the stochastic optimization in Eq. (8) still needs a method of strict descent towards true minima. In practice, we found it beneficial to periodically "sync" the plan by briefly running standard full-gradient planning on the original rollout objective.

Full-gradient rollout step. Every K_{sync} iterations, we perform J_{sync} steps of gradient descent on the original planning loss

$$\min_{\mathbf{a}} \|s_T(\mathbf{a}, s_0) - g\|_2^2, \tag{9}$$

where $s_T(\mathbf{a}, s_0)$ is computed by sequentially rolling out the world model

$$s_{t+1} = F_\theta(s_t, a_t), \qquad s_0 \text{ given.} \tag{10}$$

During this synchronization phase we update only the actions,

$$\mathbf{a} \leftarrow \mathbf{a} - \eta_{\text{sync}} \nabla_{\mathbf{a}} \|s_T(\mathbf{a}, s_0) - g\|_2^2, \tag{11}$$

using full backpropagation through the T-step rollout. By keeping these GD steps small relative to the stochastic dynamics of Eq. (8), we benefit from the smoothed loss landscape in Figure 1c for wider exploration, and the sharp but brittle landscape in Figure 1b for refinement.

4 Results

We evaluate our proposed planner GRASP across two complementary classes of environments designed to test (i) nonconvex long-horizon planning with obstacles and (ii) data-driven visual control under learned dynamics. Concretely, these experiments aim to answer three questions: (1) Can the proposed planner overcome the greedy local minima that often trap shooting methods? (2) Does the method remain robust as the planning horizon increases? (3) Does the proposed planner converge to plans faster than rollout-based planners?

We provide self-ablations in Table 2, demonstrating the value of various components of our planner: using the state gradient detaching, the GD sync steps, and the level of the noise.

Figure 4 visualizes planning iterations in several navigation environments, illustrating how trajectories initialized far from dynamically consistent rollouts converge to feasible plans that satisfy the learned dynamics.

4.1 Baselines

We compare against three commonly used planners. **CEM** optimizes action sequences by iteratively sampling candidate trajectories, selecting elites, and refitting a sampling distribution. **GD** directly optimizes the action sequence by backpropagating through the dynamics model. **LatCo** (Rybkin

	H=40 Succ. / Time	H=50 Succ. / Time	H=60 Succ. / Time	H=70 Succ. / Time	H=80 Succ. / Time
CEM	**61.4** / 35.3s	30.2 / 96.2s	7.2 / 83.1s	7.8 / 156.1s	2.8 / 132.2s
LatCo	15.0 / 598.0s	4.2 / 1114.7s	2.0 / 231.5s	0.0 / —	0.0 / —
GD	51.0 / 18.0s	37.6 / 76.3s	16.4 / 146.5s	12.0 / 103.1s	6.4 / 161.3s
GRASP (Ours)	59.0 / **8.5s**	**43.4** / **15.2s**	**26.2** / **49.1s**	**16.0** / **79.9s**	**10.4** / **58.9s**

Table 1: Open-loop planning results on long range Push-T. Reported are success rate (%) and median success time (seconds; successful trials only) across planning horizons. 500 trials per setting.

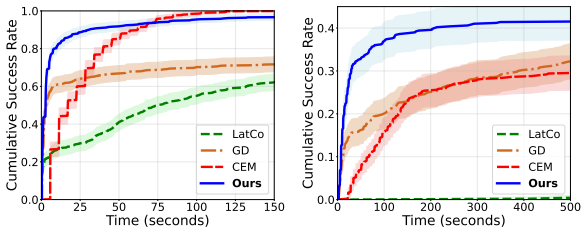
et al., 2021) optimizes in a lifted latent/state-space by jointly adjusting intermediate latent variables and actions. This setting is different than the original LatCo method, which was originally applied in a model-based RL environment, but it still provides an important baseline for performance if we were to purely focus on direct optimization of Eq. (3).

For all methods, we sweep over hyperparameters and report results using the best-performing setting for each environment and horizon. For our planner, we initialize the states $\{s_t\}_{t=0}^{T}$ as noised around the linear interpolation between s_0 and g: $s_t = \frac{t}{T}g + (1 - \frac{t}{T})s_0 + z$, $z \sim \mathcal{N}(0, \epsilon I)$, and actions initialized at zeros: $a_t = 0$.

4.2 Environments and Evaluation Protocol

We evaluate planning on three visual control environments with learned dynamics: *PointMaze*, *Wall-Single*, and *Push-T*. World models are trained using the DINO-WM framework (Zhou et al., 2024), following the original paper's setup, where the world model $F_\theta(s, a)$ takes 5 actions and predicts 5 steps ahead; that is, if $\dim(\mathcal{A}) = 2$, then F_θ takes actions as vectors of stacked actions $\mathbf{a} \in \mathbb{R}^{10}$. Likewise as in DINO-WM, start/goal pairs are taken from dataset trajectories.

All reported metrics measure task success under the learned world model. Success is defined as reaching the goal region within the planning horizon. All horizon lengths reported are environment horizons (so $H = 50$ corresponds to 10 DINO-WM evaluations).



(a) Point Maze ($H = 50$) (b) Push-T ($H = 50$)

Figure 5: **Success rate over time at a fixed horizon.** Success rate over fixed set of *open-loop* planning tasks for CEM, GD, LatCo (Rybkin et al., 2021), and our planner for a fixed horizon of 50. Curves summarize how quickly each planner makes progress under the learned world model setting when evaluated at a fixed planning horizon. Shaded regions are Wald 95% confidence intervals.

4.3 Long-term Planning and Horizon Scaling

We evaluate planners in the long-horizon regime, where our parallelized stochastic planner is more intended; greedy local minima and optimization instability become the dominant challenges.

GRASP remains reliable as the planning horizon increases: it solves more tasks and finishes a majority of its successful trials faster, showing stronger robustness to long horizons than the baselines as shown in Table 1. Beyond the median completion times, we provide further illustration of the solving speed of our planner in Figure 5, showing further that most of its plans converge at an earlier time. At the longer horizons for Push-T, where it is more important for the planner to explore non-greedy optima, GRASP finds more success than the baselines and is able to find the needed non-greedy trajectories, as visualized in Figure 7.

4.4 SHORT-TERM PLANNING

We also evaluate short-horizon planning, to demonstrate that our planner can match performance on shorter, easier tasks. Table 3 reports success rates across environments for horizons ranging from $H = 10$ to $H = 30$, while Table 4 reports median wall-clock planning times.

Across all environments and short horizons, the proposed planner achieves success rates comparable to the baselines. Alongside similar success rates, our proposed planner exhibits consistently low planning times. As shown in Table 4, it is among the fastest methods across all environments and horizons, often significantly faster than sampling-based approaches and competitive with gradient-based optimization, sacrificing some speed for a higher success rate. These results indicate that even in relatively short and easy planning regimes, our planner remains competitive with the baselines.

Overall, these results demonstrate that GRASP consistently matches strong baselines in short-term planning, while outperforming them in long-horizon settings by avoiding greedy failures and converging more quickly in wall-clock time.

5 RELATED WORK

World modeling has shown significant improvement in sample efficiency for model-based reinforcement learning (Hafner et al., 2025). By learning to predict future states given current states and actions, world models enable planning without access to an interactive environment (Ding et al., 2024). Recent work has focused on learning latent-space representations to handle high-dimensional observations Assran et al. (2025), with models now demonstrating the ability to scale and generalize across diverse environments (Bar et al., 2025). In this paper, we develop an efficient planning algorithm for action-conditioned video models.

Table 2: Ablation Studies over the GD sync steps, level of noise for Langevin dynamics, and whether we use our detached gradient approach with the goal-reaching objective or not. GD sync happens every 100 stochastic steps. Ablations done on the Push-T environment at horizon $H = 40$. Time reported is median over successful trials, which only beats our method when the accuracy is much smaller.

Setting	Accuracy (%)	Time (s)
Sync Steps		
No Sync	1.6	4.5
10	48.0	11.2
25 (ours)	**59.0**	8.5
50	58.0	9.8
Noise Level		
$\sigma = 0.0$	54.8	10.4
$\sigma = 0.5$ (ours)	**59.0**	8.5
$\sigma = 1.0$	50.6	8.4
Detach $\nabla_s F(s, a)$		
Stop (ours)	**59.0**	8.5
Flow	46.6	10.3

Sampling-based planning in world models traditionally relies on methods like the Cross-Entropy Method (CEM Rubinstein & Kroese (2004)) and random shooting. While these methods are robust and simple to implement, they suffer from serial evaluation bottlenecks and poor scaling with planning horizon length (Bharadhwaj et al., 2020). Recent work proposes performance improvements—such as faster CEM variants with action correlation and memory, parallelized sampling via diffeomorphic transforms, and massively parallel strategies—but fundamental limitations remain for very long horizons (Pinneri et al., 2021; Lai et al., 2022).

Gradient-based planning leverages the differentiability of neural world models to optimize action sequences directly (Jyothir et al., 2023). Early approaches applied backpropagation through time to optimize actions (Thrun et al., 1990), but face challenges with vanishing/exploding gradients and poor conditioning over long horizons. Hybrid strategies combining gradient descent with sampling-based methods—such as interleaving CEM with gradient updates—have shown promise. CEM-GD variants interleave backwards passes through the learned model with population-based search for improved convergence and scalability (Bharadhwaj et al., 2020; Huang et al., 2021). Recent work improves gradient-based planners by training world models to be adversarially robust to improve gradient-based planning (Parthasarathy et al., 2025); our method instead tries to improve gradient-based planning for more general world models, without any pretraining modifications.

State optimization and multiple shooting in optimal control. The idea of treating states as optimization variables separate from dynamics constraints has a rich history in classical optimal control. Non-condensed QP formulations in MPC (Jerez et al., 2011) decouple state and input optimization

	Push-T			PointMaze			WallSingle		
Planner	H=10	H=20	H=30	H=10	H=20	H=30	H=10	H=20	H=30
CEM	**100.0**	**96.0**	**76.0**	**100.0**	**100.0**	**100.0**	**98.0**	**100.0**	**100.0**
LatCo	67.4	55.2	35.4	**100.0**	99.8	99.0	91.4	81.6	62.8
GD	99.2	88.4	69.0	94.6	91.8	94.6	91.8	92.8	91.6
GRASP	**100.0**	89.2	75.2	**100.0**	**100.0**	99.8	95.2	91.8	95.0

Table 3: Short Term Planning. Success rate for Push-T, PointMaze, and WallSingle. 500 trials per setting. Our method has comparable success rate with consistently low completion time (Table 4).

	Push-T			PointMaze			WallSingle		
Planner	H=10	H=20	H=30	H=10	H=20	H=30	H=10	H=20	H=30
CEM	1.3	7.5	23.6	0.7	2.0	6.3	0.9	1.5	2.1
LatCo	39.9	437.2	1800.0	1.6	13.7	36.6	12.1	23.8	111.1
GD	**0.5**	2.4	25.3	**0.3**	**0.6**	**1.2**	**0.4**	**0.5**	**0.7**
GRASP	1.5	**2.1**	**9.1**	0.7	1.8	2.1	1.6	1.8	2.1

Table 4: Median completion times (seconds) across short-term experiments. 500 trials per setting. Our method has a consistently low completion time with a comparable success rate (Table 3).

for improved numerical properties. Multiple shooting methods (Tamimi & Li, 2009; Diedam & Sager, 2018) break long-horizon problems into shorter segments with continuity constraints. Direct collocation approaches (Bordalba et al., 2022; Nie & Kerrigan, 2025) optimize state and control trajectories simultaneously while enforcing dynamics through collocation constraints. These methods have primarily been applied to systems with known analytical dynamics. Trajectory optimization methods in robotics have developed parallel shooting techniques and GPU-accelerated planning algorithms (Guhathakurta et al., 2022), but most approaches still face fundamental limitations when applied to learned world models, particularly visual world models where dynamics are approximate and high-dimensional.

Noise and regularization in optimization. Stochastic optimization techniques and noise injection have long been recognized for their ability to improve optimization outcomes (Robbins & Monro, 1951), and can help regularize and explore complex loss landscapes (Welling & Teh, 2011; Xu et al., 2018; Bras, 2023; Foret et al., 2020). In the context of planning, noise is commonly used in sampling-based methods, but its systematic incorporation into gradient-based planning for learned models remains underexplored.

6 Limitations and Future Work

Although the proposed planner shows clear advantages in long-horizon settings, its benefits are more limited at short horizons. As demonstrated in our experiments, for small planning horizons the planner typically achieves success rates and completion times that are comparable to strong baselines such as CEM and gradient-based optimization, rather than strictly outperforming them. This suggests that the primary gains of the method arise in regimes where long-horizon reasoning and non-greedy planning are essential, rather than in short-horizon settings where simpler methods already perform well.

Hybrid planners (that combine iterations of a rollout-based planner like CEM and a gradient-based planner like GD) have been implemented to get the "best of both worlds" from the two approaches (Huang et al., 2021), and there are many ways to "hybridize" our planner as well. We leave exploration of such methods for future work.

Several components of the planner are designed to mitigate the unreliability of state gradients in learned world models. While effective, these modifications introduce additional structure and hyperparameters that would ideally be unnecessary. If state representations induced by the world model were smoother or more geometrically well-behaved in the state space, many of these stabilization mechanisms could be removed, potentially leading to further speed improvements. Promising

directions toward this goal include improved representation learning through adversarial training, diffusion-based world models, or other techniques that explicitly regularize the geometry of the learned state space.

7 Conclusion

World models provide a powerful framework for planning in complex environments, but existing approaches struggle with long horizons, high-dimensional actions, and serial computation. We propose GRASP, a new gradient-based planning algorithm with two key contributions: (a) a lifted planner that optimizes actions together with "virtual states" in a time-parallel manner, yielding more stable and scalable optimization while allowing direct control over exploration via stochastic state updates, and (b) an action-gradient-only planning variant for learned visual world models that avoids brittle state-input gradients while still exploiting differentiability with respect to actions. Experiments on visual world-model benchmarks show that our approach remains robust as horizons grow and finds non-greedy solutions at a faster rate than commonly used planners such as CEM or vanilla GD.

References

Uri M Ascher, Robert MM Mattheij, and Robert D Russell. *Numerical solution of boundary value problems for ordinary differential equations*. SIAM, 1995.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Learning for Dynamics and Control*, pp. 277–286. PMLR, 2020.

Hans Georg Bock and Karl-Josef Plitt. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608, 1984.

Ricard Bordalba, Tobias Schoels, Lluís Ros, Josep M Porta, and Moritz Diehl. Direct collocation methods for trajectory optimization in constrained robotic systems. *IEEE Transactions on Robotics*, 39(1):183–202, 2022.

Pierre Bras. Langevin algorithms for very deep neural networks with application to image classification. *Procedia Computer Science*, 222:303–310, 2023.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient

descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018, 2019.

H Diedam and Sebastian Sager. Global optimal control with the direct multiple shooting method. *Optimal Control Applications and Methods*, 39(2):449–470, 2018.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Saul B Gelfand and Sanjoy K Mitter. Recursive stochastic algorithms for global optimization in r^d. *SIAM Journal on Control and Optimization*, 29(5):999–1018, 1991.

Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models can leverage human videos for dexterous manipulation. *arXiv preprint arXiv:2512.13644*, 2025.

Dipanwita Guhathakurta, Fatemeh Rastgar, M Aditya Sharma, K Madhava Krishna, and Arun Kumar Singh. Fast joint multi-robot trajectory optimization by gpu accelerated batch solution of distributed sub-problems. *Frontiers in Robotics and AI*, 9:890385, 2022.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

Kevin Huang, Sahin Lale, Ugo Rosolia, Yuanyuan Shi, and Anima Anandkumar. Cem-gd: Cross-entropy method with gradient descent planner for model-based reinforcement learning. *arXiv preprint arXiv:2112.07746*, 2021.

Juan L Jerez, Eric C Kerrigan, and George A Constantinides. A condensed and sparse qp formulation for predictive control. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 5217–5222. IEEE, 2011.

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

Saurabh Koju, Saurav Bastola, Prashant Shrestha, Sanskar Amgain, Yash Raj Shrestha, Rudra PK Poudel, and Binod Bhattarai. Surgical vision world model. In *MICCAI Workshop on Data Engineering in Medical Imaging*, pp. 1–10. Springer, 2025.

Tin Lai, Weiming Zhi, Tucker Hermans, and Fabio Ramos. Parallelised diffeomorphic sampling-based motion planning. In *Conference on robot learning*, pp. 81–90. PMLR, 2022.

Yuanbo Nie and Eric C Kerrigan. Reliable solution to dynamic optimization problems using integrated residual regularized direct collocation. *arXiv preprint arXiv:2503.09123*, 2025.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pp. 1049–1065. PMLR, 2021.

Jorge L Piovesan and Herbert G Tanner. Randomized model predictive control for robot navigation. In *2009 IEEE International Conference on Robotics and Automation*, pp. 94–99. IEEE, 2009.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

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.

Oleh Rybkin, Chuning Zhu, Anusha Nagabandi, Kostas Daniilidis, Igor Mordatch, and Sergey Levine. Model-based reinforcement learning via latent-space collocation. In *International Conference on Machine Learning*, pp. 9190–9201. PMLR, 2021.

Adi Shamir, Odelia Melamed, and Oriel BenShmuel. The dimpled manifold model of adversarial examples in machine learning. *arXiv preprint arXiv:2106.10151*, 2021.

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.

Jasem Tamimi and Pu Li. Nonlinear model predictive control using multiple shooting combined with collocation on finite elements. *IFAC Proceedings Volumes*, 42(11):703–708, 2009.

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

Sebastian Thrun, Knut Möller, and Alexander Linden. Planning with an adaptive world model. *Advances in neural information processing systems*, 3, 1990.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

Oskar Von Stryk. Numerical solution of optimal control problems by direct collocation. In *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*, pp. 129–143. Springer, 1993.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 2002.

Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1433–1440. IEEE, 2016.

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of langevin dynamics based algorithms for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

A THEORY

In this section, we provide a theoretical analysis of the convergence properties of our planning approach compared to traditional shooting methods. We consider a simplified linear dynamics setting to derive formal convergence guarantees. We reproduce proof here for self-containedness; see e.g. Ascher et al. (1995) for other theoretical treatments of this problem.

A.1 CONVERGENCE OF VARIOUS METHODS IN THE CONVEX SETTING

Consider a linear dynamical system with dynamics:

$$s_{t+1} = As_t + Ba_t, \tag{12}$$

where $s_t \in \mathbb{R}^n$ is the state at time t, $a_t \in \mathbb{R}^m$ is the control input, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$.

Let $\mathcal{F}(s_0, \mathbf{a}) : \mathbb{R}^n \times \mathbb{R}^{mT} \to \mathbb{R}^n$ denote the rollout of the dynamics for T timesteps. Under linear dynamics, $\mathcal{F}$ takes the compact form:

$$\mathcal{F}(s_0, \mathbf{a}) = A^T s_0 + \sum_{t=0}^{T-1} A^{T-1-t} Ba_t. \tag{13}$$

We analyze the optimization landscape of two fundamental formulations for reaching a target state g from initial state s_{init}.

The *Shooting Method* minimizes the error at the final step relative to the control inputs $\mathbf{a}$, implicitly simulating the dynamics:

$$\min_{\mathbf{a}\in\mathbb{R}^{mT}} J_S(\mathbf{a}) = \|\mathcal{F}(s_{init}, \mathbf{a}) - g\|_2^2 \tag{14}$$

The *Lifted States Method* (or Multiple Shooting) treats both states and controls as optimization variables and minimizes the violation of the dynamics constraints (the physics defects):

$$\min_{\mathbf{a},\mathbf{s}} \quad J_L(\mathbf{a}, \mathbf{s}) = \sum_{t=0}^{T-1} \|As_t + Ba_t - s_{t+1}\|_2^2 \tag{15}$$

$$\text{subject to} \quad s_0 = s_{init}, \quad s_T = g \tag{16}$$

where $\mathbf{s} = (s_0, \dots, s_T)$.

Matrix Representation To analyze the convergence properties, we express both objectives in quadratic matrix form.

Shooting Method. Let $\mathbf{a} = (a_0; \dots; a_{T-1}) \in \mathbb{R}^{mT}$. The final state is linear in $\mathbf{a}$:

$$s_T = A^T s_{init} + \mathcal{C}_T \mathbf{a} \tag{17}$$

where $\mathcal{C}_T = [A^{T-1}B, A^{T-2}B, \dots, B] \in \mathbb{R}^{n\times mT}$ is the controllability matrix. The objective is:

$$J_S(\mathbf{a}) = \|\mathcal{C}_T \mathbf{a} - (g - A^T s_{init})\|_2^2 \tag{18}$$

The Hessian of this objective is $H_S = 2\mathcal{C}_T^\top \mathcal{C}_T$.

Lifted Method. We eliminate the fixed boundary variables s_0 and s_T and optimize over the free variables $\mathbf{z} = (s_1; \dots; s_{T-1}; a_0; \dots; a_{T-1})$. The dynamics residuals can be written as a linear system $M\mathbf{z} - b$. The dynamics equations for $t = 0, \dots, T-1$ correspond to rows in a large matrix M.

- For $t = 0$: $s_1 - Ba_0 = As_{init}$ (Const. term moved to RHS)
- For $0 < t < T-1$: $As_t + Ba_t - s_{t+1} = 0$
- For $t = T-1$: $-As_{T-1} - Ba_{T-1} = -g$ (s_T fixed to g)

The objective is $J_L(\mathbf{z}) = \|M\mathbf{z} - b\|_2^2$, and the Hessian is $H_L = 2M^\top M$. The matrix M has a sparse, block-banded structure.

Smoothness Analysis We compare the smoothness of the two optimization problems by comparing the Lipschitz constants of their gradients, $L = \lambda_{\max}(H)$.

Theorem 2 (Shooting: Exploding Smoothness)**.** *Let $A^\top$ have a real eigenvalue λ with $|\lambda| > 1$ and unit eigenvector v (a left eigenvector of A). Assume B aligns with this mode such that for some input direction w ($\|w\|_2 = 1$), the projection $|\langle v, Bw\rangle| = \mu > 0$.*

Then, the Lipschitz constant of the Shooting method gradient grows exponentially with T:

$$L_S = \lambda_{\max}(H_S) \geq 2\mu^2|\lambda|^{2(T-1)}. \tag{19}$$

Proof. The Lipschitz constant is the maximum eigenvalue of the Hessian $H_S = 2\mathcal{C}_T^\top \mathcal{C}_T$, which equals $2\|\mathcal{C}_T\|_2^2$. By definition, the spectral norm is the maximum gain over all possible inputs:

$$\|\mathcal{C}_T\|_2 = \sup_{\mathbf{a}\neq 0} \frac{\|\mathcal{C}_T\mathbf{a}\|_2}{\|\mathbf{a}\|_2}. \tag{20}$$

From the existence of a controllable non-contractive mode, we can construct a unit-norm $\mathbf{a}_{test} = (w; \mathbf{0}; \ldots; \mathbf{0})$ that evaluates to the form:

$$\mathcal{C}_T\mathbf{a}_{test} = A^{T-1}Bw, \tag{21}$$

and such that the following holds when projecting to the corresponding unstable eigenvector v:

$$\begin{aligned}
\|\mathcal{C}_T\mathbf{a}_{test}\|_2 &\geq |\langle v, A^{T-1}Bw\rangle|, && (22)\\
&= |\langle (A^\top)^{T-1}v, Bw\rangle|, && (23)\\
&= |\langle \lambda^{T-1}v, Bw\rangle|, && (24)\\
&= |\lambda|^{T-1}|\langle v, Bw\rangle|, && (25)\\
&= \mu|\lambda|^{T-1}. && (26)
\end{aligned}$$

Squaring this result gives the desired bound. □

Theorem 3 (Lifted: Stable Smoothness)**.** *The Lipschitz constant of the Lifted gradient is bounded by a constant independent of the horizon T. Specifically, one valid bound is:*

$$L_L = \lambda_{\max}(H_L) \leq 6\big(1 + \|A\|_2^2 + \|B\|_2^2\big). \tag{27}$$

Proof. The Hessian is $H_L = 2M^\top M$, so $\lambda_{\max}(H_L) = 2\|M\|_2^2$. It therefore suffices to upper bound $\|M\|_2$ by a constant that does not depend on T.

The matrix M is block-sparse: each block row corresponding to timestep t contains at most three non-zero blocks, namely an identity block (selecting s_{t+1}), a dynamics block (multiplying s_t), and an input block (multiplying a_t). Equivalently, the corresponding residual has the form

$$r_t \;=\; As_t + Ba_t - s_{t+1}, \tag{28}$$

with s_0 and s_T treated as fixed boundary values (so r_t is affine in the free variables).

Fix any optimization vector $\mathbf{z} = (s_1; \ldots; s_{T-1}; a_0; \ldots; a_{T-1})$ (stacking the free states and controls), and let $M\mathbf{z}$ denote the stacked residuals $(r_0, \ldots, r_{T-1})$ with constants removed. Using the inequality $\|x+y+z\|_2^2 \leq 3(\|x\|_2^2+\|y\|_2^2+\|z\|_2^2)$ and the operator norm bounds $\|As\|_2 \leq \|A\|_2\|s\|_2$, $\|Ba\|_2 \leq \|B\|_2\|a\|_2$, we obtain for each t:

$$\begin{aligned}
\|r_t\|_2^2 &= \|-s_{t+1} + As_t + Ba_t\|_2^2 && (29)\\
&\leq 3\Big(\|s_{t+1}\|_2^2 + \|A\|_2^2\|s_t\|_2^2 + \|B\|_2^2\|a_t\|_2^2\Big). && (30)
\end{aligned}$$

Summing over $t = 0, \ldots, T-1$ gives

$$\|M\mathbf{z}\|_2^2 \;=\; \sum_{t=0}^{T-1}\|r_t\|_2^2 \;\leq\; 3\sum_{t=0}^{T-1}\Big(\|s_{t+1}\|_2^2 + \|A\|_2^2\|s_t\|_2^2 + \|B\|_2^2\|a_t\|_2^2\Big). \tag{31}$$

Crucially, due to the banded structure, each free intermediate state $s_1, \ldots, s_{T-1}$ appears in at most two terms in the sum: once as s_{t+1} and once as s_t. Hence the state contributions can be bounded without any accumulation in T, yielding the following:

$$\|M\mathbf{z}\|_2^2 \leq 3\Big((1+\|A\|_2^2)\sum_{t=1}^{T-1}\|s_t\|_2^2 \;+\; \|B\|_2^2\sum_{t=0}^{T-1}\|a_t\|_2^2\Big), \tag{32}$$

$$\leq 3\big(1+\|A\|_2^2+\|B\|_2^2\big)\,\|\mathbf{z}\|_2^2. \tag{33}$$

Therefore, $\|M\|_2^2 = \sup_{\mathbf{z}\neq 0}\|M\mathbf{z}\|_2^2/\|\mathbf{z}\|_2^2 \leq 3(1+\|A\|_2^2+\|B\|_2^2)$, and thus

$$L_L \;=\; 2\|M\|_2^2 \;\leq\; 6\big(1+\|A\|_2^2+\|B\|_2^2\big), \tag{34}$$

which is independent of T. □

Interpretation. While we needed a slightly more restrictive assumption for the lower bound on the Shooting method's Hessian, such a condition is not unreasonable to find, as many realistic systems will not be universally stable within the controllability subspace. In these settings, the shooting method forces the optimization to traverse a loss landscape with curvature that varies exponentially (requiring exponentially small steps to remain stable), while the lifted states method "preconditions" the problem by creating variables for intermediate states, decoupling the long-term dependencies into local constraints. This results in a loss landscape with uniform smoothness $O(1)$ with respect to the planning horizon length T.

A.2 Gaussian noise regularization

The regularity from noisy gradient descent (or Langevin-based optimization) primarily stems from the smoothing of the Gaussian convolution:

Theorem 4 (Gaussian smoothing contracts gradients and yields scale control)**.** *Let $d \geq 1$, $\sigma > 0$, and let*

$$\phi_\sigma(\boldsymbol{\epsilon}) = (2\pi\sigma^2)^{-d/2}\exp\big(-\|\boldsymbol{\epsilon}\|^2/(2\sigma^2)\big) \tag{35}$$

be the density of $\mathcal{N}(0,\sigma^2\mathbf{I}_d)$. For $L \in \mathcal{C}^1(\mathbb{R}^d)$ define

$$L_\sigma(\mathbf{s}) = (\phi_\sigma * L)(\mathbf{s}) = \int_{\mathbb{R}^d}\phi_\sigma(\boldsymbol{\epsilon})\,L(\mathbf{s}-\boldsymbol{\epsilon})\,d\boldsymbol{\epsilon}. \tag{36}$$

The following statements hold of the resulting convolution:

1. *(Regularity never decreases.) For any $1 \leq p \leq \infty$, if the distributional gradient $\nabla_s L \in L^p(\mathbb{R}^d)$, then*
$$\|\nabla_s L_\sigma\|_{L^p} \;\leq\; \|\nabla_s L\|_{L^p}. \tag{37}$$
In particular, if L is Lipschitz, then $\mathrm{Lip}(L_\sigma) = \|\nabla_s L_\sigma\|_{L^\infty} \leq \|\nabla_s L\|_{L^\infty} = \mathrm{Lip}(L)$.

2. *(Explicit regularity control by variance.) For any $1 \leq p \leq \infty$, if $L \in L^p(\mathbb{R}^d)$, then*
$$\|\nabla_a L_\sigma\|_{L^p} \;\leq\; \frac{c_d}{\sigma}\,\|L\|_{L^p}, \qquad c_d := \mathbb{E}\|Z\| = \sqrt{2}\,\frac{\Gamma\big(\frac{d+1}{2}\big)}{\Gamma\big(\frac{d}{2}\big)},\;\; Z \sim \mathcal{N}(0,\mathbf{I}_d). \tag{38}$$
Applying $p=\infty$, we get $\mathrm{Lip}(L_\sigma) \leq \frac{c_d}{\sigma}\,\|L\|_{L^\infty}$.

Proof. An important property of convolution is that gradients commute to either argument:

$$\nabla_s L_\sigma = \nabla_s(\phi_\sigma * L), \tag{39}$$

$$= (\nabla\phi_\sigma) * L, \tag{40}$$

$$= \phi_\sigma * (\nabla_s L), \tag{41}$$

where $\nabla\phi_\sigma(\boldsymbol{\epsilon}) = -(\boldsymbol{\epsilon}/\sigma^2)\,\phi_\sigma(\boldsymbol{\epsilon})$.

For part 1, apply Young's convolution inequality with the fact that $\|\phi_\sigma\|_{L^1} = 1$. For any $g \in L^p$,

$$\|g * \phi_\sigma\|_{L^p} \le \|g\|_{L^p}. \tag{42}$$

Taking $g = \nabla_a L$ and using the commutation identity above gives

$$\|\nabla_s L_\sigma\|_{L^p} = \|\phi_\sigma * (\nabla_s L)\|_{L^p} \le \|\nabla_s L\|_{L^p}. \tag{43}$$

When $p = \infty$, $\|\nabla_s L\|_{L^\infty}$ is the Lipschitz constant of L, so $\mathrm{Lip}(L_\sigma) \le \mathrm{Lip}(L)$.

For part 2, again use the commutation identity together with Young's inequality in the form $\|h * k\|_{L^p} \le \|h\|_{L^1}\|k\|_{L^p}$. We obtain

$$\|\nabla_a L_\sigma\|_{L^p} = \|(\nabla\phi_\sigma) * L\|_{L^p} \le \|\nabla\phi_\sigma\|_{L^1}\, \|L\|_{L^p}. \tag{44}$$

It remains to compute $\|\nabla\phi_\sigma\|_{L^1}$. Since $\|\nabla\phi_\sigma(\boldsymbol{\epsilon}\|_2 = \|\boldsymbol{\epsilon}\|_2\phi_\sigma(\boldsymbol{\epsilon})/\sigma^2$ and $X \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d)$ has density ϕ_σ, we get

$$\begin{aligned}
\|\nabla\phi_\sigma\|_{L^1} &= \int_{\mathbb{R}^d} \frac{\|\boldsymbol{\epsilon}\|_2}{\sigma^2}\, \phi_\sigma(\boldsymbol{\epsilon})\, d\boldsymbol{\epsilon} && (45)\\
&= \frac{1}{\sigma^2}\, \mathbb{E}\|X\|_2 && (46)\\
&= \frac{1}{\sigma}\, \mathbb{E}\|Z\|_2 && (47)\\
&= \frac{c_d}{\sigma}, && (48)
\end{aligned}$$

where $X = \sigma Z$ with $Z \sim \mathcal{N}(0, \mathbf{I}_d)$. Substituting this into the previous display yields the claimed bound. The $p = \infty$ statement is the corresponding Lipschitz estimate. □

We then get regularity in expectation by noting that, after adding noise to a gradient step, this noise is then fed as input into the next step.

Corollary 1 (Noise induces smoothed gradients in expectation)**.** *Let $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d)$ and define the Gaussian-smoothed loss*

$$L_\sigma(\mathbf{s}) := \mathbb{E}_{\boldsymbol{\xi}}\big[L(\mathbf{s} + \boldsymbol{\xi})\big]. \tag{49}$$

Assume $L \in \mathcal{C}^1(\mathbb{R}^d)$ and that $\mathbb{E}_{\boldsymbol{\xi}}\|\nabla L(\mathbf{s} + \boldsymbol{\xi})\| < \infty$ (so differentiation may be interchanged with expectation). Then

$$\mathbb{E}_{\boldsymbol{\xi}}\big[\nabla L(\mathbf{s} + \boldsymbol{\xi})\big] = \nabla_{\mathbf{s}} L_\sigma(\mathbf{s}). \tag{50}$$

Moreover, if $L \in L^\infty(\mathbb{R}^d)$, then by Theorem 4 (Part 2),

$$\|\nabla_{\mathbf{s}} L_\sigma\|_{L^\infty} \le \frac{c_d}{\sigma}\, \|L\|_{L^\infty}, \qquad \textit{and hence} \qquad \Big\|\mathbb{E}_{\boldsymbol{\xi}}\big[\nabla L(\mathbf{s} + \boldsymbol{\xi})\big]\Big\| \le \frac{c_d}{\sigma}\, \|L\|_{L^\infty}. \tag{51}$$

This then provides motivation for the decaying noise schedule: as an annealing process from a smoother, less accurate gradient structure to a rigid but more accurate gradient structure. For more detailed theory on the regularity of noisy gradient descent, see e.g. Chaudhari et al. (2019) and the references therein.

A.3 Connection to Boltzmann sampling, state-only noising

The continuous-time Langevin dynamics corresponding to Eq. (4), if we were to also noise the actions a_t similarly, has a stationary distribution that concentrates on low-energy regions of $\mathcal{L}_{\mathrm{dyn}}(\mathbf{s}, \mathbf{a})$; in particular, under mild regularity conditions it admits the Gibbs (Boltzmann) density

$$p(\mathbf{s}, \mathbf{a}) \;\propto\; \exp\big(-\beta\, \mathcal{L}_{\mathrm{dyn}}(\mathbf{s}, \mathbf{a})\big), \tag{52}$$

for an inverse temperature $\beta > 0$ determined by the relative scaling between the drift and diffusion terms[1]. However, we are only noising the states, so the converged distribution is not Boltzmann. The new converged distribution, written loosely, collapses on solutions in action to the local dynamics problems:

$$p(\mathbf{s}, \mathbf{a}) \propto \delta(\mathbf{a} = \mathbf{a}^*(\mathbf{s})) \cdot \exp\big(-\beta\, \mathcal{L}_{\mathrm{dyn}}(\mathbf{s}, \mathbf{a})\big), \tag{53}$$

where $\mathbf{a}^*(\mathbf{s}) = \min_{\mathbf{a}} \mathcal{L}_{\mathrm{dyn}}(\mathbf{s}, \mathbf{a})$.

[1] Equivalently, one can view σ_{state} in Eq. (4) as setting an effective temperature: larger σ_{state} yields broader exploration, while smaller σ_{state} concentrates around local minima.

A.4 STATE-GRADIENT-FREE DYNAMICS

We now analyze a variant of the optimization method where we "cut" the gradient flow through the state evolution in the dynamics. This is akin to removing the adjoint (backward) pass through time, transforming the global trajectory optimization into a sequence of locally regularized problems.

First, we provide proof that there are no loss functions where (i) the minimizers correspond to dynamically feasible trajectories, that also (ii) has no dependency on the state gradients through the world model $F_\theta(s, a)$. We formalize this in the following theorem:

Theorem 5 (Nonexistence of exact dynamics-enforcing losses with Jacobian-free state gradients)**.** *Let $\mathcal{S} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{R}^m$ each be open and connected sets, and fix $s_0 \in \mathcal{S}$ and $g \in \mathcal{S}$. Consider horizon $T = 2$ with decision variables $(s_1, a_0, a_1) \in \mathcal{S} \times \mathcal{A} \times \mathcal{A}$ and boundary $s_2 = g$ fixed. Let $L_F : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ be decomposable into the following form:*

$$L_F(s_1, a_0, a_1) = \Phi(s_1, a_0, a_1, y_0, y_1), \tag{54}$$

$$y_0 = F(s_0, a_0), \tag{55}$$

$$y_1 = F(s_1, a_1), \tag{56}$$

where $\Phi : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \times \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is C^1. Let

$$\mathcal{M}(F) = \{(s_1, a_0, a_1) : s_1 = F(s_0, a_0) \text{ and } g = F(s_1, a_1)\} \tag{57}$$

denote the set of trajectories that satisfy the dynamics exactly at both steps (with the fixed boundary conditions). There does not exist such a Φ for which the following two properties hold simultaneously for every C^1 model F:

1. *Minimizers correspond to feasible dynamics:* $\arg\min_{(s_1,a_0,a_1)} L_F(s_1, a_0, a_1) = \mathcal{M}(F)$,
2. *Independence of loss to the dynamics' state gradient: for all $G : \mathcal{S} \times \mathcal{A} \to \mathcal{S}, G \in C^1$, $\Big(F(s_0, a_0) = G(s_0, a_0), F(s_1, a_1) = G(s_1, a_1)\Big) \Rightarrow \nabla_{s_1} L_F(s_1, a_0, a_1) = \nabla_{s_1} L_G(s_1, a_0, a_1)$.*

Proof. We argue by contradiction.

Fix any point (s_1, a_0, a_1) and write $y_0 = F(s_0, a_0)$ and $y_1 = F(s_1, a_1)$. By the chain rule,

$$\nabla_{s_1} L_F(s_1, a_0, a_1) = \frac{\partial \Phi}{\partial s_1}(s_1, a_0, a_1, y_0, y_1) + \big(\nabla_s F(s_1, a_1)\big)^\top \frac{\partial \Phi}{\partial y_1}(s_1, a_0, a_1, y_0, y_1). \tag{58}$$

We claim that $\partial\Phi/\partial y_1$ must vanish identically. To see this, fix arbitrary arguments $(s_1, a_0, a_1, y_0, y_1), (s_0, a_0) \neq (s_1, a_1)$ in the domain and construct two C^1 models F and G such that $F(s_0, a_0) = G(s_0, a_0) = y_0$ and $F(s_1, a_1) = G(s_1, a_1) = y_1$, but whose state Jacobians at (s_1, a_1) are prescribed arbitrarily and differ:

$$\nabla_s F(s_1, a_1) = J_F, \qquad \nabla_s G(s_1, a_1) = J_G. \tag{59}$$

To construct, choose a small ball B around (s_1, a_1) contained in $\mathcal{S} \times \mathcal{A}$, construct a smooth bump function ψ that equals 1 on a smaller concentric ball and 0 outside B (possible since $\mathcal{S}, \mathcal{A}$ open), and define

$$F(s, a) = H(s, a) + \psi(s, a)\Big((y_1 - H(s_1, a_1)) + (J_F - \nabla_s H(s_1, a_1))(s - s_1)\Big), \tag{60}$$

for an arbitrary C^1 base map H. Then $F(s_1, a_1) = y_1$ and $\nabla_s F(s_1, a_1) = J_F$. Defining G analogously with J_G gives the desired pair; values at (s_0, a_0) can be kept fixed by choosing disjoint supports or applying the same local surgery at (s_0, a_0).

By Jacobian-invariance, $\nabla_{s_1} L_F = \nabla_{s_1} L_G$ at (s_1, a_0, a_1). Subtracting the two chain-rule expressions cancels $\partial\Phi/\partial s_1$ and yields

$$(J_F - J_G)^\top \frac{\partial \Phi}{\partial y_1}(s_1, a_0, a_1, y_0, y_1) = 0. \tag{61}$$

Since $J_F - J_G$ can be any matrix in $\mathbb{R}^{n\times n}$, it follows that

$$\frac{\partial \Phi}{\partial y_1}(s_1, a_0, a_1, y_0, y_1) = 0. \tag{62}$$

Because the arguments were arbitrary, and since $\mathcal{S}$ is connected, we conclude $\partial\Phi/\partial y_1 \equiv 0$, meaning the loss is independent of y_1. Therefore there exists a C^1 function $\widetilde{\Phi}$ such that for every F,

$$L_F(s_1, a_0, a_1) = \widetilde{\Phi}(s_1, a_0, a_1, F(s_0, a_0)). \tag{63}$$

In particular, if two models F and G satisfy $F(s_0, a) = G(s_0, a)$ for all $a \in \mathcal{A}$, then $L_F \equiv L_G$ as functions of (s_1, a_0, a_1), and hence

$$\arg\min L_F = \arg\min L_G. \tag{64}$$

We now construct such a pair F, G but with different feasible sets, contradicting the exact-minimizers assumption. Pick two distinct actions $u, v \in \mathcal{A}$, two distinct states $s_A, s_B \in \mathcal{S}$, and an action $a^\star \in \mathcal{A}$. Using bump-function surgery as above, construct C^1 models F and G such that

$$F(s_0, a) = G(s_0, a) \quad \text{for all } a \in \mathcal{A}, \tag{65}$$

and

$$F(s_0, u) = G(s_0, u) = s_A, \tag{66}$$
$$F(s_0, v) = G(s_0, v) = s_B, \tag{67}$$

but with swapped second-step goal reachability:

$$F(s_A, a^\star) = g, \qquad F(s_B, a^\star) \neq g, \tag{68}$$
$$G(s_A, a^\star) \neq g, \qquad G(s_B, a^\star) = g. \tag{69}$$

Then $(s_A, u, a^\star) \in \mathcal{M}(F)$ but $(s_A, u, a^\star) \notin \mathcal{M}(G)$, and $(s_B, v, a^\star) \in \mathcal{M}(G)$ but $(s_B, v, a^\star) \notin \mathcal{M}(F)$, so $\mathcal{M}(F) \neq \mathcal{M}(G)$. On the other hand, since $F(s_0, \cdot) = G(s_0, \cdot)$ we have $\arg\min L_F = \arg\min L_G$. By assumption (i),

$$\arg\min L_F = \mathcal{M}(F) \quad \text{and} \quad \arg\min L_G = \mathcal{M}(G), \tag{70}$$

implying $\mathcal{M}(F) = \mathcal{M}(G)$, a contradiction. Therefore, no such Φ can exist. □

We introduce the stop-gradient operator $\mathrm{sg}(\cdot)$, where $\mathrm{sg}(x) = x$ during the forward evaluation, but $\nabla \mathrm{sg}(x) = 0$ during the backward pass. The modified objective incorporating the stop-gradient mechanism is:

$$\begin{aligned}\mathcal{L}_{sg}(\mathbf{s}, \mathbf{a}) = &\sum_{t=0}^{T-1} \|s_{t+1} - A\mathrm{sg}(s_t) - Ba_t\|_2^2 \\ &+ \sum_{t=0}^{T-1} \beta_t \|g - A\mathrm{sg}(s_t) - Ba_t\|_2^2\end{aligned} \tag{71}$$

where $\beta_t \geq 0$ are the goal loss coefficients. Note that the target g is effectively applied as a penalty on the state at each step to guide the local optimization.

Theorem 6 (Linear convergence to a unique fixed point)**.** *Consider the gradient descent iteration*

$$(\mathbf{s}^{k+1}, \mathbf{a}^{k+1}) = (\mathbf{s}^k, \mathbf{a}^k) - \eta \nabla \mathcal{L}_{sg}(\mathbf{s}^k, \mathbf{a}^k)$$

where s_0 is fixed and gradients are computed with the stop-gradient convention (i.e., treating $sg(s_t)$ as constant during differentiation). Assume the linear dynamics setting and that

$$\beta_t \in [\beta_{\min}, \beta_{\max}] \quad \textit{for all } t, \qquad \beta_{\min} > 0,$$

and that B has full column rank (equivalently, $\sigma_{\min}(B) > 0$). Then there exists a stepsize $\eta \in (0, \bar{\eta})$, where $\bar{\eta}$ depends only on $\beta_{\min}, \beta_{\max}, \|B\|_2, \sigma_{\min}(B)$ (and not on T), such that the induced update operator $\mathcal{T}$ on $\mathbf{z} = (\mathbf{s}, \mathbf{a})$ has a unique fixed point $\mathbf{z}^\star$ and the iterates converge linearly:

$$\|\mathbf{z}^k - \mathbf{z}^\star\| \leq Cq^k \|\mathbf{z}^0 - \mathbf{z}^\star\|, \qquad \textit{for some } q \in (0, 1) \textit{ and } C > 0.$$

Proof. Write the objective (re-indexing the goal term to match the action index) as

$$\mathcal{L}_{sg}(\mathbf{s},\mathbf{a}) = \sum_{t=0}^{T-1} \|s_{t+1} - A\,\mathrm{sg}(s_t) - Ba_t\|_2^2 \;+\; \sum_{t=0}^{T-1} \beta_t \|g - A\,\mathrm{sg}(s_t) - Ba_t\|_2^2. \tag{72}$$

Define the residuals

$$r_t := s_{t+1} - As_t - Ba_t, \qquad e_t := g - As_t - Ba_t. \tag{73}$$

Under the stop-gradient convention, $\mathrm{sg}(s_t)$ is treated as constant during differentiation, so s_t does not receive gradient contributions through the $A\,\mathrm{sg}(s_t)$ terms. It follows that the only state-gradient at time t comes from the appearance of s_t as the *next* state in the previous residual, namely

$$\nabla_{s_t}\mathcal{L}_{sg} = 2r_{t-1}, \qquad t = 1,\dots,T, \tag{74}$$

with the understanding that $r_{-1} = 0$ if s_0 is fixed. Likewise, the action-gradient at time t is

$$\nabla_{a_t}\mathcal{L}_{sg} = -2B^\top r_t - 2\beta_t B^\top e_t, \qquad t = 0,\dots,T-1. \tag{75}$$

Therefore, gradient descent with stepsize $\eta > 0$ yields the explicit update rules

$$s_t^{k+1} = s_t^k - 2\eta\big(s_t^k - As_{t-1}^k - Ba_{t-1}^k\big), \qquad t = 1,\dots,T, \tag{76}$$

and

$$a_t^{k+1} = a_t^k + 2\eta B^\top\Big(\big(s_{t+1}^k - As_t^k - Ba_t^k\big) + \beta_t\big(g - As_t^k - Ba_t^k\big)\Big), \qquad t = 0,\dots,T-1. \tag{77}$$

Stack the variables in the time-ordered vector $\mathbf{z} := (s_1, a_0, s_2, a_1, \dots, s_T, a_{T-1})$. The updates above define an affine map $\mathbf{z}^{k+1} = \mathcal{T}(\mathbf{z}^k) = J\mathbf{z}^k + c$ whose Jacobian J is block lower-triangular with respect to this ordering: indeed, $(s_{t+1}^{k+1}, a_t^{k+1})$ depends only on $(s_t^k, s_{t+1}^k, a_t^k)$ (and on the fixed constants g and s_0) and is independent of any future variables $(s_{t+2}^k, a_{t+1}^k, \dots)$. Consequently, the eigenvalues of J are exactly the union of the eigenvalues of its diagonal blocks.

To characterize a diagonal block, fix $t \in \{0,\dots,T-1\}$ and consider the pair $y_t := (s_{t+1}, a_t)$. Conditioned on s_t (which appears only as a constant inside $\mathrm{sg}(s_t)$ for differentiation), the update $(s_{t+1}^{k+1}, a_t^{k+1})$ is precisely one gradient step on the quadratic function

$$\phi_t(s_{t+1}, a_t; s_t) = \|s_{t+1} - As_t - Ba_t\|_2^2 + \beta_t\|g - As_t - Ba_t\|_2^2, \tag{78}$$

so the corresponding diagonal block equals $I - \eta H_t$ where $H_t = \nabla_{y_t}^2\phi_t$ is the constant Hessian with respect to (s_{t+1}, a_t):

$$H_t = 2\begin{bmatrix} I & -B \\ -B^\top & (1+\beta_t)B^\top B \end{bmatrix}. \tag{79}$$

Assume $\beta_t \in [\beta_{\min}, \beta_{\max}]$ with $\beta_{\min} > 0$ and that B has full column rank, so $B^\top B \succ 0$. The Schur complement of the I block is

$$(1+\beta_t)B^\top B - B^\top I^{-1} B = \beta_t B^\top B \succ 0, \tag{80}$$

hence $H_t \succ 0$ for every t, with eigenvalues uniformly bounded away from 0 and ∞ as t varies:

$$0 < \mu I \preceq H_t \preceq LI < \infty, \tag{81}$$

for constants μ, L depending only on $\beta_{\min}, \beta_{\max}, \|B\|_2$, and $\sigma_{\min}(B)$. Choosing any stepsize η such that $0 < \eta < 2/L$, we obtain for every t that all eigenvalues of $I - \eta H_t$ lie strictly inside the unit disk, and in particular:

$$\rho(I - \eta H_t) \le \max\{|1 - \eta\mu|,\ |1 - \eta L|\} =: q < 1, \tag{82}$$

where q is independent of t and T. Since J is block lower-triangular and its diagonal blocks are exactly $(I - \eta H_t)$ (up to a fixed permutation corresponding to the stacking order), we conclude

$$\rho(J) = \max_t \rho(I - \eta H_t) \le q_0 < 1. \tag{83}$$

Fix any q such that $q_0 < q < 1$. By applying Gelfand's formula, for any square matrix M and any $q > \rho(M)$, there exists an induced norm $\|\cdot\|_\dagger$ such that $\|M\|_\dagger \le q$. Applying this with $M = J$ yields a norm $\|\cdot\|_\dagger$ satisfying

$$\|J\|_\dagger \le q < 1. \tag{84}$$

Hence, for all $k \geq 0$,

$$\|\mathbf{z}^{k+1} - \mathbf{z}^\star\|_\dagger = \|J(\mathbf{z}^k - \mathbf{z}^\star)\|_\dagger \leq \|J\|_\dagger \, \|\mathbf{z}^k - \mathbf{z}^\star\|_\dagger \leq q \, \|\mathbf{z}^k - \mathbf{z}^\star\|_\dagger, \tag{85}$$

and therefore

$$\|\mathbf{z}^k - \mathbf{z}^\star\|_\dagger \leq q^k \|\mathbf{z}^0 - \mathbf{z}^\star\|_\dagger. \tag{86}$$

By norm equivalence in finite dimensions, there exists $C > 0$ such that

$$\|\mathbf{z}^k - \mathbf{z}^\star\|_2 \leq C q^k \|\mathbf{z}^0 - \mathbf{z}^\star\|_2. \tag{87}$$

□

Notes on the stopgrad optimization. The optimization indeed converges to a fixed point, but one can show that these stable points in the linear convex case are merely the greedy rollouts towards the goal. Two things make the optimization in our setting nontrivial: the nonconvexity of the world model F_θ, and the stochastic noise on the states s_t. We now present some characterization on the distribution of trajectories that our planner tends towards.

Let $F_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ be a differentiable world model and define the stop-gradient one-step prediction

$$\mu_t \;:=\; F_\theta(\bar{s}_t, a_t), \qquad \bar{s}_t = \mathrm{stopgrad}(s_t).$$

Consider the stopgrad lifted objective (cf. Eq. Eq. (7))

$$\mathcal{L}(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{T-1} \|\mu_t - s_{t+1}\|_2^2 \;+\; \gamma \sum_{t=0}^{T-1} \|\mu_t - g\|_2^2, \tag{88}$$

and the (no-sync) optimization updates

$$s_{t+1}^{k+1} = s_{t+1}^k - 2\eta_s\big(s_{t+1}^k - \mu_t^k\big) + \sigma\, \xi_{t+1}^k, \qquad \xi_{t+1}^k \sim \mathcal{N}(0, I), \tag{89}$$

$$a_t^{k+1} = a_t^k - \eta_a \nabla_{a_t} \mathcal{L}(\mathbf{s}^k, \mathbf{a}^k). \tag{90}$$

Throughout, assume $0 < \eta_s < 1$ (for stability of the state contraction).

Theorem 7 (Gaussian tube around one-step predictions)**.** *Fix $\{\bar{s}_t^k\}_{t=0}^{T-1}$ and $\{a_t^k\}_{t=0}^{T-1}$ at iteration k, and let $\mu_t^k = F_\theta(\bar{s}_t^k, a_t^k)$. Then the state update Eq. (89) satisfies the conditional mean recursion*

$$\mathbb{E}\big[s_{t+1}^{k+1} \mid \mu_t^k\big] = (1 - 2\eta_s)\, \mathbb{E}\big[s_{t+1}^k \mid \mu_t^k\big] + 2\eta_s\, \mu_t^k. \tag{91}$$

Moreover, if $\mu_t^k \equiv \mu_t$ is held fixed, then s_{t+1}^k converges in distribution to a Gaussian "tube" around μ_t:

$$s_{t+1}^\infty \;\sim\; \mathcal{N}\big(\mu_t,\, \Sigma_{\mathrm{tube}}\big), \qquad \Sigma_{\mathrm{tube}} = \frac{\sigma^2}{1 - (1 - 2\eta_s)^2}\, I = \frac{\sigma^2}{4\eta_s(1 - \eta_s)}\, I. \tag{92}$$

Analogously, in continuous optimization-time τ, the limiting SDE

$$ds_{t+1}(\tau) = -\lambda\big(s_{t+1}(\tau) - \mu_t\big)d\tau + \sigma\, dW_{t+1}(\tau), \quad \lambda > 0,$$

has stationary law $\mathcal{N}(\mu_t, \frac{\sigma^2}{2\lambda} I)$.

Proof. Rewrite Eq. (89) as an affine Gaussian recursion:

$$s_{t+1}^{k+1} = (1 - 2\eta_s) s_{t+1}^k + 2\eta_s \mu_t^k + \sigma \xi_{t+1}^k.$$

Taking conditional expectation yields Eq. (91). If $\mu_t^k \equiv \mu_t$ is fixed, the centered process $u^k := s_{t+1}^k - \mu_t$ satisfies $u^{k+1} = (1 - 2\eta_s)u^k + \sigma\xi^k$, i.e. an AR(1) process with contraction factor $|1 - 2\eta_s| < 1$. Its unique stationary covariance Σ_{tube} solves the discrete Lyapunov equation $\Sigma_{\mathrm{tube}} = (1 - 2\eta_s)^2 \Sigma_{\mathrm{tube}} + \sigma^2 I$, giving Eq. (92). The continuous-time statement is standard, given that μ_t is fixed. □

Theorem 8 (Goal shaping induces goal-directed drift of tube center)**.** *Define* $\mu_t^k = F_\theta(\bar{s}_t^k, a_t^k)$ *and let*

$$J_t^k \;:=\; \nabla_{a_t} F_\theta(\bar{s}_t^k, a_t^k) \in \mathbb{R}^{d_s \times d_a}, \qquad P_t^k \;:=\; J_t^k (J_t^k)^\top \succeq 0.$$

Assume a first-order linearization in the action step:

$$\mu_t^{k+1} \approx \mu_t^k + J_t^k\,(a_t^{k+1} - a_t^k). \tag{93}$$

Then the action update Eq. (90) induced by Eq. (88) yields the tube-center evolution

$$\mu_t^{k+1} \;\approx\; \underbrace{\big(I - \alpha\gamma P_t^k\big)\mu_t^k + \alpha\gamma P_t^k g}_{\textit{goal-directed averaging (drift)}} \;+\; \underbrace{\alpha\, P_t^k\, \varepsilon_{t+1}^k}_{\textit{exploration forcing}}, \qquad \alpha := 2\eta_a, \tag{94}$$

where ε_{t+1}^k *denotes the tube residual* $s_{t+1}^k = \mu_t^k + \varepsilon_{t+1}^k$*. In particular, if* $\mathbb{E}[\varepsilon_{t+1}^k \mid \mu_t^k] = 0$*, then*

$$\mathbb{E}[\mu_t^{k+1} \mid \mu_t^k, s_t^k, a_t^k] \;\approx\; \big(I - \alpha\gamma P_t^k\big)\mu_t^k + \alpha\gamma P_t^k g, \tag{95}$$

so in controllable directions the mean prediction μ_t *moves toward* g *as an averaging step. If* $0 < \alpha\gamma\lambda_{\max}(P_t^k) < 1$*, this is a contractive averaging step toward* g *on* $\mathrm{Range}(P_t^k)$*.*

Proof. From Eq. (88), only terms at time t depend on a_t via μ_t. Using $\nabla_{a_t}\mu_t = J_t$, the gradient is

$$\nabla_{a_t}\mathcal{L} = 2(J_t)^\top(\mu_t - s_{t+1}) + 2\gamma(J_t)^\top(\mu_t - g). \tag{96}$$

Thus the action step is

$$a_t^{k+1} - a_t^k = -\eta_a \nabla_{a_t}\mathcal{L} = -2\eta_a (J_t^k)^\top(\mu_t^k - s_{t+1}^k) - 2\eta_a\gamma(J_t^k)^\top(\mu_t^k - g).$$

Apply the linearization Eq. (93):

$$\begin{aligned}\mu_t^{k+1} &\approx \mu_t^k - 2\eta_a J_t^k (J_t^k)^\top(\mu_t^k - s_{t+1}^k) - 2\eta_a\gamma J_t^k (J_t^k)^\top(\mu_t^k - g)\\ &= \mu_t^k - 2\eta_a P_t^k(\mu_t^k - (\mu_t^k + \varepsilon_{t+1}^k)) - 2\eta_a\gamma P_t^k(\mu_t^k - g),\end{aligned}$$

which simplifies to Eq. (94) after substituting $\alpha = 2\eta_a$. Taking conditional expectation and using $\mathbb{E}[\varepsilon_{t+1}^k \mid \mu_t^k] = 0$ yields Eq. (95). □

Unlike the stopgrad lifted-state dynamics (Theorem 7), the rollout distribution is not contracted toward the current rollout, but instead noise accumulates throughout nonlinear iterations of the world model. The stochastic rollout distribution need not concentrate in a local tube around the current deterministic rollout trajectory; it can drift and spread away as the horizon T grows.

Theorem 9 (Mean evolution and non-tube behavior of noisy rollouts)**.** *Consider a rollout-based stochastic trajectory generated in model-time:*

$$s_{t+1} = F_\theta(s_t, a_t) + \sigma_{\text{env}}\zeta_t, \qquad \zeta_t \sim \mathcal{N}(0, I), \qquad s_0 \textit{ fixed}, \tag{97}$$

and a dense goal objective along the rollout (e.g. $\sum_{t=0}^{T-1}\|s_{t+1} - g\|^2$*). Let* $m_t := \mathbb{E}[s_t]$ *and* $\Sigma_t := \mathrm{Cov}(s_t)$*.*

The rollout mean obeys the exact identity

$$m_{t+1} = \mathbb{E}[F_\theta(s_t, a_t)]\,. \tag{98}$$

In particular, if F_θ *is affine in* s *(i.e.* $F_\theta(s, a) = As + Ba + c$*), then*

$$m_{t+1} = F_\theta(m_t, a_t). \tag{99}$$

For general nonlinear F_θ*, a second-order moment expansion yields*

$$m_{t+1} \approx F_\theta(m_t, a_t) + \frac{1}{2}\left(\mathrm{Hess}_s F_\theta(m_t, a_t)\right) : \Sigma_t, \tag{100}$$

so the mean generally does not *follow the deterministic rollout* $F_\theta(m_t, a_t)$*.*

A first-order linearization gives the approximate covariance propagation

$$\Sigma_{t+1} \approx G_t\,\Sigma_t\,G_t^\top + \sigma_{\text{env}}^2 I, \qquad G_t := \nabla_s F_\theta(\mu_t, a_t). \tag{101}$$

Proof. Taking conditional expectation of Eq. (97) gives $\mathbb{E}[s_{t+1} \mid s_t] = F_\theta(s_t, a_t)$, and then total expectation implies Eq. (98). For affine F_θ, expectation commutes with F_θ, yielding Eq. (99). For nonlinear F_θ, expand $F_\theta(s_t, a_t)$ around μ_t by Taylor's theorem; the first-order term vanishes in expectation and the second-order term produces Eq. (100). The covariance recursion Eq. (101) follows by linearizing $F_\theta(s_t, a_t) \approx F_\theta(m_t, a_t) + G_t(s_t - \mu_t)$ and computing $\mathrm{Cov}(\cdot)$, adding the independent noise variance $\sigma_{\text{env}}^2 I$. The final "non-tube" claim follows because there is no optimization-time contraction that repeatedly pulls s_{t+1} back toward a moving center (as in Eq. (91)); instead the forward propagation Eq. (101) typically increases spread and the mean can deviate from the deterministic path by Eq. (100). □

Theorem 7 shows that noisy lifted state updates form an noisy "tube" around the one-step predictions $m_t = F_\theta(\bar{s}_t, a_t)$, keeping exploration local and dynamically consistent in optimization-time. Theorem 8 then shows that dense one-step goal shaping moves the tube center toward the goal by a preconditioned averaging step, while the dynamics residual contributes approximately zero-mean stochastic forcing that enables exploration without horizon-coupled backpropagation. In contrast, Theorem 9 shows that noisy rollouts evolve by forward propagation of randomness: the mean follows $m_{t+1} = \mathbb{E}[F_\theta(s_t, a_t)]$ (not generally the deterministic rollout), and the distribution can drift/spread rather than concentrate in a tube around the current plan.

B Ablations

To rigorously evaluate the contribution of each individual component within our proposed framework, we conducted an ablation study on the Push-T environment with a horizon of 40 steps. In this experiment, we systematically removed specific modules or mechanisms from the full method while keeping all other hyperparameters constant. This analysis allows us to isolate the impact of important components such as: the noised-states optimization, the GD sync steps, and whether our state-gradient-approach is needed

The results, summarized in Table 2, highlight the critical role of each design choice in achieving robust performance. We observe that the **Full Method** achieves the highest success rate, validating the synergy between the proposed components.

The stopgrad is an important component, without it the state optimization feels "sticky" and is difficult to tune. Noise also remains an important component to allow the planner to explore around local minima.

We also ablate different, simpler options of a noise and gradient-based planner, building off a normal rollout-based GD planner. We ablate with two kinds of noise:

1. *Action noise*, where actions are sampled around Gaussians of a fixed variance σ_a before rollout evaluation and gradient computation.
2. *State noise*, where noise is added during the rollout directly to the states:

$$s_{t+1} = F(s_t, a_t) + z, \tag{102}$$

$$z \sim \mathcal{N}(0, \sigma_s I). \tag{103}$$

These ablations are provided in Table 8.

C Hyperparameters

The following are hyperparameters used for both the world model and our planner. We provide the model architecture and training hyperparameters used for the DINO-WM model on the PushT environment in Tables 5 and 6. These are the same as in the original paper (Zhou et al., 2024). Table 7 lists the hyperparameters used for our planner.

D Additional experiments

Component	**Specification**
Image size	224×224
Frameskip (action chunking)	5
History length (T_h, number of predictions)	3
Prediction length (T_p)	1
Encoder (frozen)	
Backbone	DINOv2 (`dinov2_vits14`)
Feature key	`x_norm_patchtokens`
Action / Proprio Embedding	
Action dim	10 (2×5)
Proprio dim	10 (2×5)
Predictor (ViT)	
Depth	6
Heads	16
MLP dim	2048
Dropout	0.1
Pooling	mean
Decoder (VQ-VAE)	
Channels	384
Codebook size (n_{embed})	2048
Residual blocks	4
Residual channels	128

Table 5: DINO-WM architecture for PushT.

Hyperparameter	**Value**
Epochs	500
Batch size	32
Encoder learning rate	1×10^{-6}
Predictor learning rate	5×10^{-4}
Decoder learning rate	3×10^{-4}
Action encoder learning rate	5×10^{-4}
Train encoder	False
Train predictor	True
Train decoder	True

Table 6: Training configuration for DINO-WM on PushT.

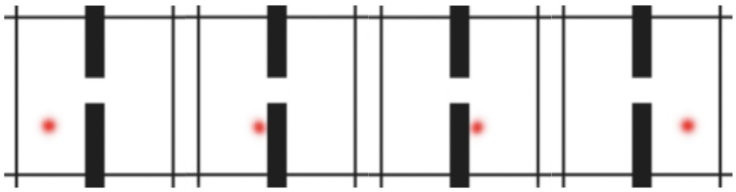


Figure 6: Example of a bad local minima when relaxing the dynamics constraint into a penalty function. The dynamics are nonzero for the middle transition, but both the states and actions are at a local minima; there's no direction they can move locally to reduce the dynamics loss further.

Hyperparameter	Value
Optimization	
Optimizer	Adam
State learning rate (η_s)	10^{-1}
Action learning rate (η_a)	10^{-3}
Goal loss weight (γ)	1
State Exploration (Langevin-style noise)	
State noise scale (σ_{state})	0.5
Periodic GD Refinement	
Use GD sync	True
GD interval	100
GD optimization steps	25
GD learning rate	0.1
Initialization	
State init method	cylinder
Initialization noise level	0.95
Action init distribution	Gaussian
Action init mean	0
Action init std	0.01

Table 7: Hyperparameters for GRASP. Cylinder initialization denotes initalizing states as a linear interpolation plus Gaussian noise, orthogonalized to the linear interpolation.

Table 8: Stochastic GD Baselines Ablation Study, showing alternatives from our method for a stochastic gradient-based planner. For rollouts of the world model $F_\theta(s, a)$, values of σ_a correspond to variance of noise added to actions throughout optimization, and values of σ_s correspond to noise added to states directly through the rollout; that is, for rollouts of the world model, $s_{t+1} = F_\theta(s_t, a_i) + \xi, \xi \sim \mathcal{N}(0, I)$. None beat our method's performance, shown in Table 2. Reported time is average time over all experiments, with reported intervals as 95% CI.

Setting	Accuracy (%)	Time (s)
Default Mode (objective.mode=default)		
$\sigma_s = 0.0$	53.0 $\pm$ 4.4	864 $\pm$ 77
$\sigma_s = 0.5$	50.6 $\pm$ 4.4	913.97 $\pm$ 78
$\sigma_s = 1.0$	37.2 $\pm$ 4.3	1149 $\pm$ 75
$\sigma_a = 0.0$	47.6 $\pm$ 4.4	1056 $\pm$ 73
$\sigma_a = 0.5$	53.0 $\pm$ 4.4	864 $\pm$ 77
$\sigma_a = 1.0$	33.4 $\pm$ 4.2	1208 $\pm$ 73
$\sigma_s = 0.5, \sigma_a = 0.5$	53.8 $\pm$ 4.4	854 $\pm$ 78
Mode All (objective.mode=all)		
$\sigma_s = 0.0$	54.0 $\pm$ 4.4	865 $\pm$ 77
$\sigma_s = 0.5$	55.8 $\pm$ 4.4	838 $\pm$ 77
$\sigma_s = 1.0$	41.0 $\pm$ 4.4	1111 $\pm$ 75
$\sigma_a = 0.0$	46.6 $\pm$ 4.4	1067 $\pm$ 73
$\sigma_a = 0.5$	52.8 $\pm$ 4.4	873 $\pm$ 77
$\sigma_a = 1.0$	52.4 $\pm$ 4.4	878 $\pm$ 77
$\sigma_s = 0.5, \sigma_a = 0.5$	55.8 $\pm$ 4.4	838 $\pm$ 77

Figure 7: Example of converged plans for our planner on the Push-T environment at horizon 80, with goals depicted to the right of the dashed line. The parallel optimization is able to converge more consistently at longer horizons, and find the required non-greedy paths to the goal.