Improving Model-Based Reinforcement Learning by Converging to Flatter Minima

Shrinivas Ramasubramanian, **Benjamin Freed**, **Alexandre Capone**, **Jeff Schneider**

Robotics Institute, Carnegie Mellon University; Pittsburgh, PA 15213
{shrinivr, bfreed, acapone2, jeff.schneider}@cs.cmu.edu

Abstract

Model-based reinforcement learning (MBRL) hinges on a learned dynamics model whose errors can compound along imagined rollouts. We study how encouraging *flatness* in the model's training loss affects downstream control, and show that steering optimization toward flatter minima yields a better policy. Concretely, we integrate *Sharpness-Aware Minimization* (SAM) into world-model training as a drop-in objective, leaving the planner and policy components unchanged. On the theory side, we derive PAC-Bayesian bounds that link first-order sharpness to the value-estimation gap and the performance gap between model-optimal and true-optimal policies, implying that flatter minima tighten both. Empirically, SAM reduces measured sharpness and value-prediction error and improves returns across HumanoidBench, Atari-100k, and high-DoF DeepMind Control tasks. Augmenting existing MBRL algorithms with SAM increases mean return, with especially large gains in settings with high dimensional state–action spaces. We further observe positive transfer across algorithms and input modalities, including a transformer-based world-model. These results position flat-minima training as a simple, general mechanism for more robust MBRL without architectural changes. [1]

1 Introduction

Model-based reinforcement learning (MBRL) involves learning the environment dynamics by training an explicit dynamics world-model. With this model, an agent can simulate trajectories and cut real environment interaction dramatically [7, 17–19, 44, 50], enabling strong sample efficiency compared to model-free methods [5, 15, 38, 42, 43]. Yet these gains hinge on model fidelity: prediction errors can compound across imagined rollouts [24, 49], especially under distribution shift as the policy improves or when the data-collection policy differs from the control policy [8, 9, 13, 22, 27, 29–31, 46]. In practice, latent imagination models may hallucinate on long horizons, degrading downstream control [24]. A parallel literature in supervised learning links *flat* minima of the training loss to better generalization and robustness [12, 14, 25, 26, 36, 48, 55], with benefits under label imbalance, label noise, and distribution shift [3, 41, 45, 53, 54]. Sharpness-Aware Minimization (SAM) achieves this by biasing optimization toward flatter minima. SAM-like updates have improved policy-gradient and value-based agents [33, 34], but despite its centrality, the world-model in MBRL has received little attention through the lens of flat minima.

This paper studies flat-minima training for world-models in MBRL. We integrate SAM into the environment dynamics model objective as a drop-in change (no architectural or planning modifications), and examine how reducing sharpness of the model's loss relates to rollout reliability and control. Theoretically, we connect first-order sharpness to both value-estimation error and the

[1] Code: https://github.com/autonlab/MBRL-flat-minima.git

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

sub-optimality of the model-optimal policy via PAC-Bayesian bounds. Empirically, we compare standard return curves and value-prediction error versus Monte-Carlo estimates, probe sensitivity to the SAM radius ρ, and report compute overhead. Our contributions can be summarised as follows:

- We propose a drop-in SAM for world-model training; planners/policies unchanged.
- Our theoretical contribution includes a PAC-Bayesian bound that links loss landscape sharpness to value estimation error and performance gaps.
- Through hessian spectra, we show baseline minima is sharper. Our experiments provide evidence that encouraging convergence to flatter minima also reduces sharpness and value-prediction error.
- We demonstrate the **+89.1%** mean return on HumanoidBench; **+27.6%** gains on Atari-100k on a transformer world-model and **+20.6%** on high-DoF DMC environments.
- We observe that encouraging convergence to flatter minimum to different components of MBRL has different effects. It helps dynamics model the most and can aid policy when applied to policy loss, while it harms value/reward prediction heads.

Taken together, our results position flat-minima training as a *general, drop-in* mechanism for more robust MBRL: it tightens theory-grounded error terms, measurably flattens the learned model's landscape, and yields stronger policies across algorithms, observation modalities, and challenging high-DoF control tasks.

2 Related Works

A large body of work links the geometry of the loss landscape to generalization, with flatter minima correlating with improved out-of-distribution performance [12, 20, 25]. Information-theoretic and PAC-Bayesian perspectives explain this connection by relating flatness to shorter description length and broader posteriors that are robust to perturbations [21, 37]. Comparative studies identify flatness as a strong empirical predictor of generalization relative to gradient and spectral metrics [25], and recent bounds formalize how optimizing for flatness tightens generalization guarantees [1, 11, 51]. Sharpness-Aware Minimization (SAM) biases training toward flatter regions via an inner maximization over local perturbations followed by descent at the adversarial point [12]. Variants improve robustness [47, 52], incorporate curvature/Fisher information [28], analyze convergence in nonconvex settings [39], and address scale/efficiency (ASAM, Efficient SAM) [10, 32]. Empirically, batch-size and architecture studies further support the flatness–generalization link [26, 35].

MBRL attains high sample-efficiency but is sensitive to model errors that compound across long rollouts and under distribution shift as policies improve or when the data-collection and control policies differ [8, 9, 13, 16, 22, 23, 27, 29–31, 46]. Remedies include conservative/uncertainty-aware objectives [27], robust training [40], ensembles [8], and smoothness constraints (e.g., Lipschitz control) [2]. Nonetheless, latent imagination models can hallucinate on long horizons [16], motivating training schemes that reduce sensitivity to parameter perturbations and shift. Applying sharpness-aware training within RL has primarily targeted policy parameters, where flatter solutions improve plasticity and robustness under perturbations and shifts [33, 34]. These results suggest that flatness benefits translate beyond supervised learning. However, in MBRL the world-model has been comparatively underexplored through the lens of flat minima.

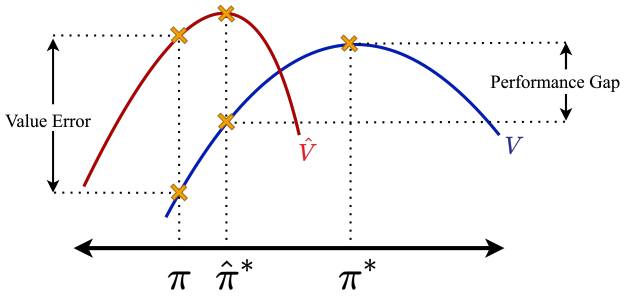

Figure 1: Illustrating value error and performance gap due to model error in estimating true returns (V) vs estimation ($\hat{V}$).

Prior sharpness-aware RL mainly targets policy/value losses; we instead optimize the world-model loss. We introduce a drop-in SAM objective for dynamics training (no planner/policy changes) and derive PAC-Bayesian bounds linking *first-order sharpness* of the model-error landscape to value- and performance-gap terms. Flat-minima training thus offers a simple, architecture-agnostic mechanism complementary to robustness/uncertainty methods [2, 8, 16].

3 Preliminaries

This section introduces fundamental concepts of MBRL, defines sharpness in loss landscapes, and discusses the general benefits of converging to flatter minima.

3.1 Model-Based Reinforcement Learning (MBRL)

In a Markov Decision Process (MDP), an agent interacts with an environment by taking an action a_t in a state s_t, receiving a reward r_t, and transitioning to a new state s_{t+1} according to unknown dynamics $P(s_{t+1}|s_t, a_t)$ and reward function $r(s_t, a_t)$. The agent's goal is to learn a policy $\pi(a_t|s_t)$ that maximizes the expected discounted sum of future rewards, $V^\pi(s_0) = \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^t r_t|s_0]$, where $\gamma \in [0, 1)$ is the discount factor. In MBRL, we learn an explicit model of the dynamics $\hat{P}_\theta(s_{t+1}|s_t, a_t)$ and/or reward function $\hat{r}_\theta(s_t, a_t)$, collectively denoted as the world-model $\hat{M}_\theta$, parameterized by θ. This model is typically trained by minimizing a loss function $\mathcal{L}_\mathcal{S}(\theta)$ on a dataset $\mathcal{S}$ of collected transitions (s_t, a_t, r_t, s_{t+1}). For example, $\mathcal{L}_\mathcal{S}(\theta)$ could be the mean squared error (MSE) between predicted and actual next states:

$$\mathcal{L}_\mathcal{S}(\theta) = \frac{1}{|\mathcal{S}|} \sum_{(s,a,s')\in\mathcal{S}} \|\hat{s}'_\theta(s, a) - s'\|^2,$$

where $\hat{s}'_\theta(s, a)$ is the model's prediction for the next state. Once learned, $\hat{M}_\theta$ can be used for planning (e.g., Model Predictive Control), policy optimization via simulated experience, or to compute value functions $\hat{V}^\pi(s_0)$ based on model roll-outs. A critical challenge in MBRL is ensuring the learned model $\hat{M}_\theta$ generalizes well, especially to state-action pairs encountered as the policy improves or during planning, which may differ from the training data distribution [9, 31]. Poor generalization can lead to compounding errors in multi-step predictions, degrading the quality of planned actions and learned policies [23]. The robustness of the model to such distributional shifts and to inherent uncertainties is paramount for effective MBRL. We later define d^π as the discounted state-action occupancy measure for a policy π.

3.2 Flat Minima, Generalization and Sharpness-Aware Minimization

In this subsection we briefly introduce the notions of flat and sharp minima for world-model training, define the first-order sharpness ($R_\rho^{(1)}$), and explain why flatter regions tend to generalize better through both an intuitive robustness perspective and a PAC-Bayesian view that ties sharpness to complexity terms in generalization bounds. We then present SAM and its practical surrogate objective as a way to bias optimization toward flatter minima. This setup prepares the ground for our later results that relate model sharpness to value-prediction error and to the sub-optimality of the policy induced by the learned model.

3.2.1 Flat Minima and Generalization

Let $\mathcal{L}_\mathcal{S}(\theta)$ be the empirical loss of a dynamics model with parameters $\theta \in \mathbb{R}^d$ on a training dataset $\mathcal{S}$. A parameter vector $\theta^\star$ is said to lie in a *flat minimum* if the loss function $\mathcal{L}_\mathcal{S}(\theta)$ is locally insensitive to small perturbations around $\theta^\star$. Conversely, a *sharp minimum* exhibits a rapid increase in loss even for small perturbations. More formally, for a small radius $\rho > 0$, a flat minimum satisfies:

$$\max_{\|\epsilon\|_2\leq\rho} \mathcal{L}_\mathcal{S}(\theta^\star + \epsilon) \;\approx\; \mathcal{L}_\mathcal{S}(\theta^\star).$$

In this paper, we primarily consider **First-order sharpness** ($R_\rho^{(1)}(\theta)$), which quantifies the maximum squared norm of the gradient in a ρ-neighborhood around θ:

$$R_\rho^{(1)}(\theta) := \max_{\|\epsilon\|_2\leq\rho} \|\nabla_\theta \mathcal{L}_\mathcal{S}(\theta + \epsilon)\|_2^2. \tag{1}$$

A small value of $R_\rho^{(1)}(\theta)$ indicates that the gradients remain small within the neighborhood, characteristic of a flatter region. Crucially, extensive empirical studies have shown that sharp minima correlate with poor model generalization and vice versa [12, 25, 26]. Intuitively, this can be explained with a robustness argument: a sharp minimum in the loss landscape implies potentially large changes in the

loss for small errors in parameter estimation or shifts in data. Flat minima in the model loss landscape are associated with better generalization. A model residing in a flat minimum is less sensitive to small variations in its parameters, which typically translates to more robust predictions. This robustness helps mitigate the compounding of errors during multi-step predictions and improves resilience to shifts in the state-action visitation distribution as the policy evolves. Smoother Jacobians of the model function with respect to its inputs and parameters, often found in flatter regions, contribute to this stability [33].

The benefits of flat minima can also be understood from a PAC-Bayesian viewpoint [37]. If we consider a distribution $q(\theta')$ centered around the learned parameters θ (e.g., θ is the mean or mode of q), flatter minima correspond to broader posterior distributions $q(\theta')$ that still maintain low loss. Standard PAC-Bayesian generalization bounds often include a complexity term related to the KL-divergence between the posterior $q(\theta')$ and a prior $p(\theta')$. For certain choices of prior and posterior, this KL-divergence term can be related to sharpness measures like $R_\rho^{(1)}$ [1, 11]. Minimizing sharpness can thus lead to tighter PAC-Bayesian bounds on the true risk (e.g., multi-step prediction error) of the learned model, providing a theoretical underpinning for the empirical observation that flatter minima generalize better.

3.2.2 Sharpness-Aware Minimization

Having established the importance of flat minima for generalization, we now introduce Sharpness-Aware Minimization (SAM) as a technique to achieve such minima for dynamics models in MBRL. We then present our theoretical results linking the sharpness of the learned model to MBRL performance. Sharpness-Aware Minimization (SAM) [12] is an optimization procedure designed to steer the learning process towards flatter minima. Instead of minimizing the original model loss $\mathcal{L}_\mathcal{S}(\theta)$ (e.g., MSE for next-state predictions), SAM aims to minimize a modified sharpness-regularized loss, $L_\mathcal{S}^{\mathrm{SAM}}(\theta)$, which considers the worst-case loss value in a local neighborhood:

$$\min_\theta \underbrace{\max_{\|\epsilon\|_2 \le \rho} \mathcal{L}_\mathcal{S}(\theta+\epsilon)}_{=: L_\mathcal{S}^{\mathrm{SAM}}(\theta)} + \lambda\|\theta\|_2^2, \tag{2}$$

where λ is a weight-decay coefficient and ρ is the radius of the L_2-ball. The inner maximization problem seeks an adversarial perturbation ϵ that maximizes the loss. This is typically approximated by a single gradient ascent step:

$$\max_{\|\epsilon\|_2 \le \rho} \mathcal{L}_\mathcal{S}(\theta+\epsilon) \approx \mathcal{L}_\mathcal{S}\left(\theta + \rho \frac{\nabla_\theta \mathcal{L}_\mathcal{S}(\theta)}{\|\nabla_\theta \mathcal{L}_\mathcal{S}(\theta)\|_2}\right). \tag{3}$$

The parameters θ are then updated using the gradient of this surrogate loss. A single SAM step thus involves two forward-backward passes: one to compute $\nabla_\theta \mathcal{L}_\mathcal{S}(\theta)$ (to find the ascent direction) and another to compute the gradient of the loss at the perturbed point for the parameter update. By minimizing $L_\mathcal{S}^{\mathrm{SAM}}(\theta)$, SAM effectively penalizes sharp regions where $\mathcal{L}_\mathcal{S}(\theta+\epsilon)$ can be much larger than $\mathcal{L}_\mathcal{S}(\theta)$, thereby encouraging convergence to flatter minima where $R_\rho^{(1)}(\theta)$ (as defined in Eq. 1) is smaller.

4 Sharpness-Aware Minimization for Dynamics Models

We now connect the concept of flat minima, as promoted by SAM, to the performance of MBRL agents. Our theoretical motivation demonstrates that reducing the sharpness of the learned dynamics model can lead to tighter bounds on policy performance. Recall that $\hat{M}_\theta$ is the learned dynamics model, $V^\pi(s_0)$ is the true expected return, and $\hat{V}^\pi(s_0)$ is the model-estimated return. We define $\hat{L}(\theta;\pi)$ as the empirical one-step model error, representing the average discrepancy (e.g., in terms of state prediction and reward prediction) between the model $\hat{M}_\theta$ and the true environment when evaluated on data generated under policy π. This $\hat{L}(\theta;\pi)$ measures the model's performance on the true dynamics distribution induced by π, which might differ from the training loss $\mathcal{L}_\mathcal{S}(\theta)$ defined over a fixed dataset $\mathcal{S}$. The theorems below relate this policy-dependent model error and its sharpness to value estimation and policy performance. Let M_{loss} be an upper bound on the per-sample one-step

model error, n be the number of samples used to estimate $\hat{L}(\theta;\pi)$, d the number of model parameters, ρ the perturbation radius for sharpness, and δ the confidence parameter. $\Omega(d,n,\rho,\delta)$ represents a model complexity or capacity term.

Theorem 1 (Return-estimation gap)**.** *For any policy π and discount $\gamma \in [0,1)$. Let $\hat{L}(\theta;\pi)$ be the empirical model error under π computed from n i.i.d. samples, and assume the per-sample loss is bounded by M_{loss} and the first-order sharpness of minimum $R_\rho^{(1)}(\theta;\pi)$, for any initial state s_0 we have, with probability at least $1-\delta$,*

$$\left|V^\pi(s_0) - \hat{V}^\pi(s_0)\right| \;\le\; \frac{\gamma}{(1-\gamma)^2}\left[\hat{L}(\theta;\pi) + R_\rho^{(1)}(\theta;\pi) + \sqrt{M_{\text{loss}}/n} + \Omega(d,n,\rho,\delta)\right].$$

The bound on the value-estimation gap, $|V^\pi - \hat{V}^\pi|$, decomposes into three interpretable terms. First, $\hat{L}(\theta;\pi)$ is the empirical one-step prediction error incurred by the learned dynamics $\hat{M}_\theta$ under policy π; naturally, smaller prediction error tightens the bound. Second, $R_\rho^{(1)}(\theta;\pi)$ measures the first-order sharpness of the true model error landscape with respect to θ under policy π: a low value indicates that the model's error $L(\theta;\pi)$ is stable to small perturbations of the parameters θ, an essential property because θ itself is only an estimate. Finally, the sampling- and capacity-related terms $\sqrt{M_{loss}/n}$ and $\Omega(d,n,\rho,\delta)$ quantify residual uncertainty arising from finite data and model complexity. The term $\frac{\gamma}{(1-\gamma)^2}$ highlights how these one-step error and sharpness concerns can be amplified over the effective horizon of the task. Crucially, if the model error landscape is flat (small $R_\rho^{(1)}(\theta;\pi)$), the value estimates $\hat{V}^\pi$ are more reliable.

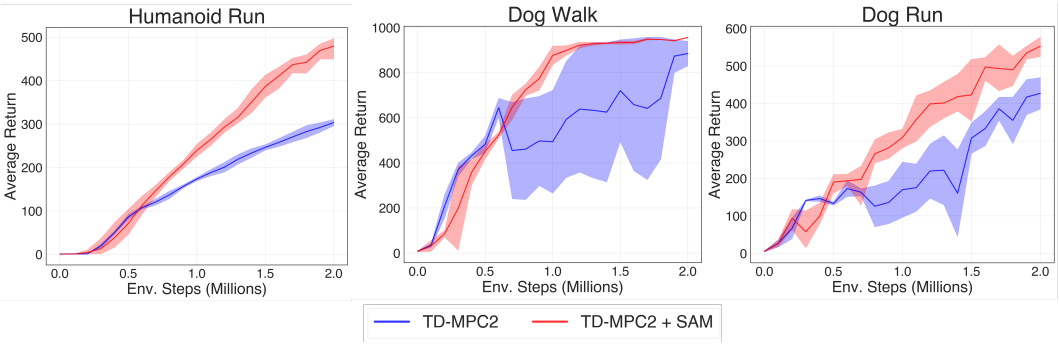

Figure 2: Comparison of applying SAM to TD-MPC2 in dynamics model for high-Dof env. in DMC. We observe a statistically significant improvement in policy in comparing TD-MPC2 + SAM (ours) against TD-MPC2 (baseline). All experiments were run upto 2M env. steps and 4 seeds

Theorem 2 (Performance Gap)**.** *Let π^* be the optimal policy in the true environment and $\hat{\pi}^*$ be the policy that is optimal according to the learned model $\hat{M}_\theta$. If $C = \max_{(s,a)} \frac{d^{\pi^*}(s,a)}{d^{\hat{\pi}^*}(s,a)}$ is the concentrability coefficient measuring the distribution mismatch between π^* and $\hat{\pi}^*$:*

$$\left|V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0)\right| \le \frac{\gamma(1+C)}{(1-\gamma)^2}\left[\hat{L}(\theta;\hat{\pi}^*) + R_\rho^{(1)}(\theta;\hat{\pi}^*) + \sqrt{\tfrac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta)\right].$$

This theorem addresses a pivotal question in MBRL: *How closely does the policy $\hat{\pi}^*$, derived from our learned model $\hat{M}_\theta$, perform compared to the true optimal policy π^*?* The performance gap, $|V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0)|$, is bounded by terms analogous to Thm. 1, specifically the model's one-step error $\hat{L}(\theta;\hat{\pi}^*)$ and the sharpness of this error landscape $R_\rho^{(1)}(\theta;\hat{\pi}^*)$. A key distinction is that these terms are now evaluated under the model-optimal policy $\hat{\pi}^*$, reflecting the model's accuracy and robustness in the regions it deems most promising for high returns. A low sharpness $R_\rho^{(1)}(\theta;\hat{\pi}^*)$ is especially important, indicating that the model's favorable assessment of $\hat{\pi}^*$ is stable and less likely to be an artifact of exploiting inaccuracies in a sharp area of its learned dynamics. The bound also incorporates the concentrability coefficient C, which acknowledges that significant distributional differences between π^* and $\hat{\pi}^*$ can widen the performance gap. A small $R_\rho^{(1)}(\theta;\hat{\pi}^*)$ suggests that the

model's assessment of $\hat{\pi}^*$ is robust, making it more likely that $\hat{\pi}^*$ is genuinely a good policy in the true environment, not just an artifact of exploiting model inaccuracies in a sharp region of the model error landscape.

Implications for SAM in MBRL. The dynamics model in MBRL is typically trained by minimizing a surrogate loss function, $\mathcal{L}_{\mathcal{S}}(\theta)$ (e.g., mean squared error of next-state predictions on a replay buffer $\mathcal{S}$). Standard optimizers aim to reduce $\mathcal{L}_{\mathcal{S}}(\theta)$. SAM modifies this process to find parameters θ that not only yield low $\mathcal{L}_{\mathcal{S}}(\theta)$ but also reside in a flat region of this training loss landscape (i.e., low $R_\rho^{(1)}(\theta)$ for $\mathcal{L}_{\mathcal{S}}$, as defined in Eq. 1). Thm. 1 and 2 feature $\hat{L}(\theta;\pi)$ (the empirical true model error under policy π) and its sharpness $R_\rho^{(1)}(\theta;\pi)$. The crucial connection is that a model trained with SAM to achieve a flat minimum for its training loss $\mathcal{L}_{\mathcal{S}}$ is expected to generalize better. Better generalization implies that the model's actual predictive accuracy on states visited by π (related to $\hat{L}(\theta;\pi)$) is more robust to variations in input and parameters. This robustness, in turn, should translate to a flatter landscape for the true model error, i.e., a smaller $R_\rho^{(1)}(\theta;\pi)$ as it appears in the theorems. Therefore, after achieving a low empirical model error $\hat{L}(\theta;\pi)$ (or $\hat{L}(\theta;\hat{\pi}^*)$), the sharpness term $R_\rho^{(1)}$ becomes a critical factor governing the reliability of value estimates and the quality of the derived policy. By promoting convergence to flatter minima of the training loss $\mathcal{L}_{\mathcal{S}}$, SAM aims to reduce the corresponding sharpness $R_\rho^{(1)}(\theta;\pi)$ in the true model error landscape. This, according to our theoretical motivation, tightens the bounds on both value estimation error and the suboptimality of the learned policy, providing a principled reason for expecting improved performance from SAM-trained dynamics models in MBRL.

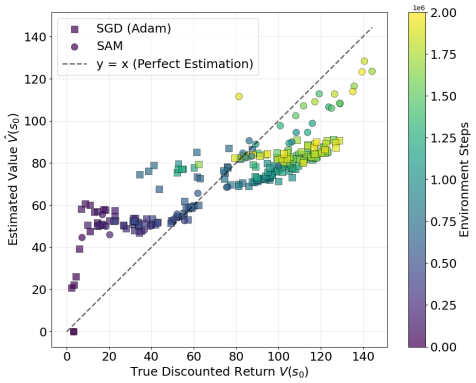

Figure 3: Scatter plot of estimating true returns V vs estimated returns $\hat{V}$. $\hat{V}$ is closer to V when we use SAM as our base optimizer especially during the later env. steps.

5 Experiments

We evaluate whether training the world-model with SAM improves downstream control by encouraging convergence to flatter minima. The theoretical results in Sec. 4 motivate the expectation that flatter minima should reduce value-estimation error and narrow the gap between model-optimal and true-optimal policies. To test this claim without altering planner or policy architectures, we integrate SAM solely into the dynamics loss of established MBRL agents and assess performance across state-based high-degree-of-freedom (DoF) control (DMC App. Tab. 10 and HumanoidBench Fig. 4) and pixel-based discrete control (Atari-100k) Tab. 1. Our evaluation reports returns and sharpness Tab. 8 via top Hessian eigenvalues and value-prediction $\hat{V}^\pi$ against Monte-Carlo returns V^π Fig. 3, and records compute overhead. Baseline hyper-parameters follow the original papers; SAM introduces a single radius parameter ρ App. Tab. 4.

5.1 TD-MPC2 on DeepMind Control (DMC)

We build on TD-MPC2 [19] and isolate the effect of sharpness-aware training on the world-model by applying SAM only to the dynamics loss while leaving the planner and policy unchanged. We evaluate on high-DoF DeepMind Control tasks: `humanoid_run`, `humanoid_walk`, `dog_run`, `dog_walk`, and `dog_trot` using the official TD-MPC2 replay and optimization schedules for **2M environment steps** per task. Results are averaged over **four seeds** and reported as mean $\pm$ SEM. To verify that SAM actually finds flatter minima, we approximate the leading Hessian eigenvalues of the dynamics loss via power iteration similar to [12] on replay mini-batches. To connect geometry to control quality, we measure value-prediction error as the absolute gap between Monte-Carlo returns collected in the environment and model-estimated returns. Unless otherwise noted, all hyper-parameters follow the TD-MPC2 defaults, and the SAM radius ρ is selected via a small coarse grid in App. Tab. 5.

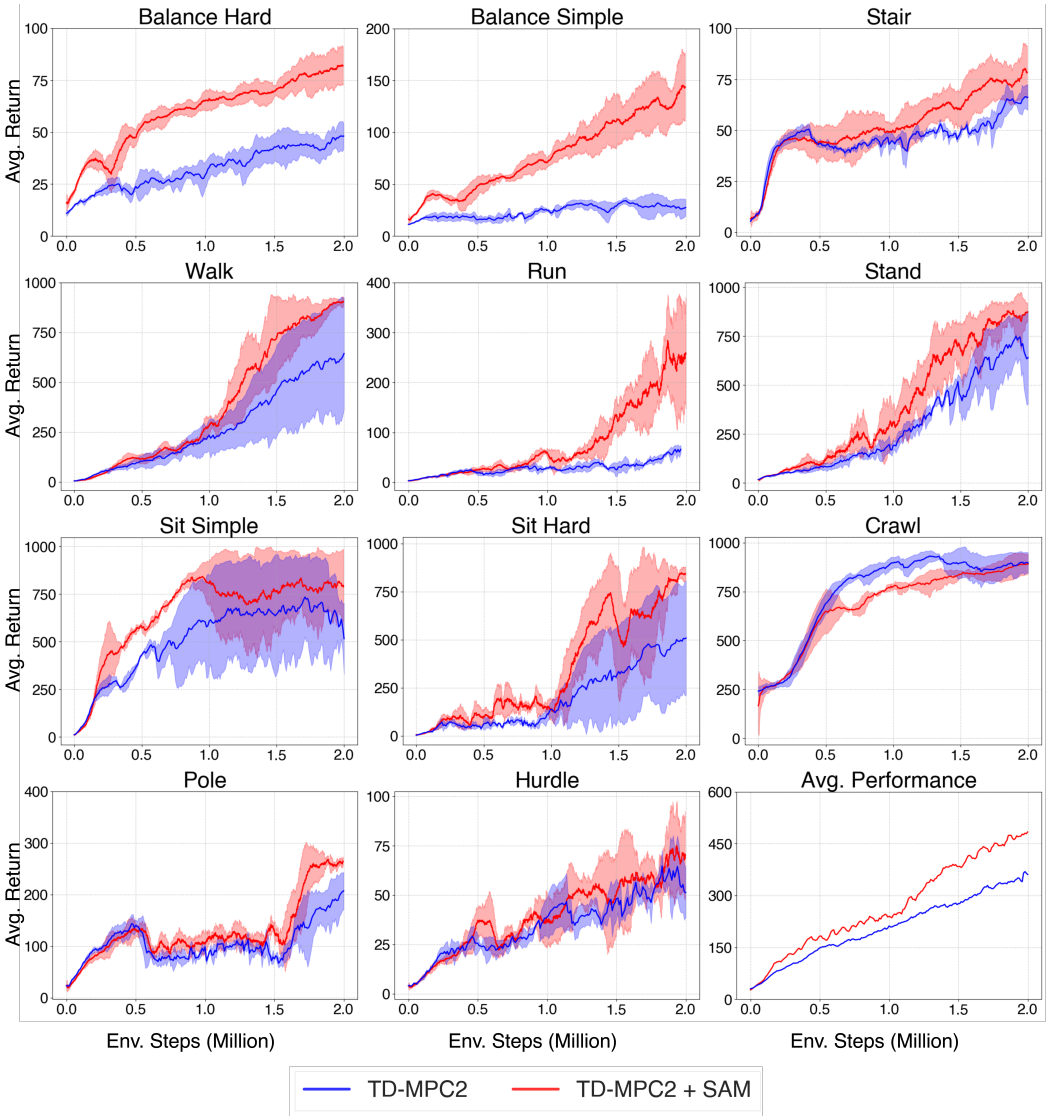

Figure 4: Learning curves for TD-MPC2 (baseline) and TD-MPC2+SAM on 11 HumanoidBench tasks, averaged over 4 seeds (shaded area is SEM). TD-MPC2+SAM demonstrates improved performance across a majority of tasks.

5.2 TD-MPC2 on HumanoidBench

In **HumanoidBench**, we integrate SAM into TD-MPC2 by modifying only the dynamics component of the world-model, replacing its training loss with a SAM surrogate while keeping the online planner, policy architecture, replay buffer, and optimization schedule exactly as in the original algorithm. We evaluate this SAM-augmented variant on the eleven-task locomotion suite for 2M environment steps per task, sweeping the SAM radius ρ once per suite, and running four random seeds; we report mean $\pm$ SEM and test for improvement using one-tailed paired t-tests across seeds (null: no gain; alternative: SAM $>$ baseline).Humanoid control couples high dimensionality and under-actuation (25 DoF with 19 actuators for the body; with two hands the action space is 61 and the state is 151 dimensions), intermittent non-smooth contacts with sharp stability margins, and long horizons that bind balance recovery to forward progress (e.g., hurdles, stairs, maze) or demand precise upper-body coordination without degrading gait (e.g., reach, balance). We report results on 11 different tasks in the suite that range in varying degree of hardness. By holding planning and policy fixed and touching only the dynamics-model objective, the study cleanly attributes any gains in sample efficiency, stability, or robustness to SAM's effect on the learned model, making HumanoidBench a stringent and informative testbed for SAM-enhanced model-based control.

5.3 TWISTER on Atari 100k

In addition to continuous-control domains, we test whether the benefits of SAM on world-model learning persist under pixel observations and discrete action spaces by applying the same modification to TWISTER, a Dreamer-style agent that replaces the recurrent world-model with a transformer while retaining the standard actor–critic heads. This experiment complements our HumanoidBench study in two important ways. We consider 19/25 Atari-100k envs. due to compute constraints. First, Atari-100k poses a stringent sample-efficiency test from raw RGB with a fixed 100k-step budget, making it a natural counterpoint to high-DoF proprioceptive control; any improvement there suggests that SAM's effect is not tied to a particular modality or planner but to the robustness of the learned world dynamics. Second, the original TWISTER paper argues that transformer world-models better capture long-range temporal structure than recurrent variants, yet such high-capacity sequence models are also susceptible to sharp minima during maximum-likelihood training [4]; inserting SAM precisely at the world-model update probes whether flattening the loss landscape yields more stable value targets and rollouts for the unchanged policy/value heads. Concretely, we evaluate TWISTER and TWISTER+SAM on twenty Atari-100k games for 100k environment steps, average over five seeds, and keep all hyperparameters identical except for the SAM radius ρ (swept on a small grid).

5.4 Results

We evaluate SAM as a drop-in change to the world-model loss in two settings TD-MPC2 for high-DoF proprioceptive control and TWISTER for pixel-based Atari and observe consistent gains. On HumanoidBench (11 tasks, 2M env. steps, 4 seeds) (Fig. 4, TD-MPC2+SAM improves mean return by **+89.1%** over the TD-MPC2 baseline, with especially pronounced and often earlier-emerging gains on locomotion-heavy tasks (e.g., `walk`, `run`, `stand`). Several tasks with small margins (`stair`, `sit_simple`) show trends favoring SAM but with mixed statistical significance; per-task t-tests are reported later (App. Tab. 3). Importantly, gains often appear *early* in training, consistent with SAM stabilizing world-model rollouts used by MPC. On high-DoF DMC (Fig. 2, App. Tab. 10), SAM similarly improves performance across `humanoid_run`, `humanoid_walk`, `dog_run`, `dog_walk`, and `dog_trot` at 2M steps (mean $\pm$ SEM over 4 seeds), yielding an average per-task relative gain of **+20.6%**, with the largest boost on `humanoid_run` (**+60.8%**; see Tab. 10. To probe modality and action-space generality, we apply the same modification to TWISTER [6] on Atari-100k (19 games, 100k steps, 5 seeds) as in the case for TD-MPC2 on HumanoidBench and high DoF-DMC envs. and find broad improvements with the SAM-augmented agent, while keeping all settings identical (Tab. 1); excluding `Freeway` where both agents score 0, TWISTER+SAM achieves a **+27.6%** improvement in human-normalized score, with large wins in absolute performance on `Battle Zone` (**+87.6%**), `Frostbite` (**+314.9%**), `Gopher` (**+97.3%**), and `Road Runner` (**+58.3%**). These results indicate that steering world-model training toward flatter minima reliably improves downstream control without changing the planner or policy, and that the effect transfers across algorithms, observation modalities, and action spaces.

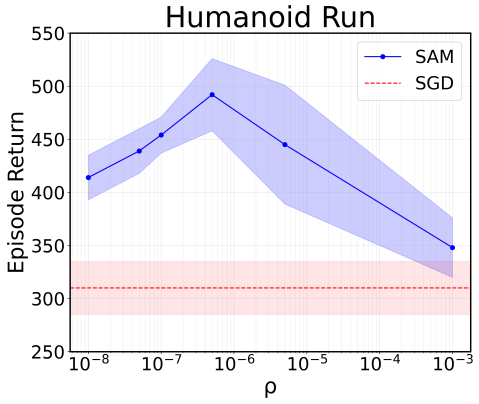

Figure 5: Returns on `humanoid_run` vs ρ for TD-MPC2 + SAM

5.5 Ablation and Sensitivity

Radius ablation and choice. Sweeping the SAM radius ρ yields a characteristic unimodal response (Fig. 5). Very small ρ under-regularizes and recovers the baseline. Very large ρ over-perturbs parameters and can destabilize training. To minimize tuning we fix a single ρ per setting based on the sweep: ρ=1.0×10^{-3} for Atari-100k, ρ=2.5×10^{-5} for high-DoF DMC, and ρ=0.00125 for HumanoidBench with a targeted exception of ρ=0.005 on `sit_simple` and `sit_hard`. These suites prefer different magnitudes, which we attribute to observation modality and model capacity. Adaptive variants such as ASAM [32] can further reduce tuning effort.

Table 1: Performance comparison across 20 Atari games. TWISTER and its SAM-augmented variant are benchmarked against prior model-based methods (TWIM, IRIS, Dreamer v3) and baseline agents (Random, Human, Simple). TWISTER w/ SAM consistently improves performance in challenging environments, demonstrating the effectiveness of sharpness-aware optimization in MBRL

Game	Random	Human	Simple	TWIM	IRIS	Dreamer v3	TWISTER	TWISTER w/ SAM
Alien	228	7128	617	675	420	959	823	947
Amidar	6	1720	74	122	143	139	172	172
Assault	222	742	527	683	1524	706	777	1102
Asterix	210	8503	1128	1116	854	932	1132	1030
Bank Heist	14	753	34	467	53	649	673	886
Battle Zone	2360	37188	4031	5068	13074	12250	5452	10230
Boxing	0	12	8	78	70	78	80	86
Demon Attack	152	1971	208	350	2034	303	286	293
Freeway	0	30	17	24	31	0	0	0
Frostbite	65	4335	237	1476	259	909	388	1610
Gopher	258	2412	597	1675	2236	3730	2078	4099
Hero	1027	30826	2657	7254	7037	11161	9836	12320
James Bond	29	303	100	362	463	445	354	426
Kangaroo	52	3035	51	1240	838	4098	1349	1555
Ms Pacman	307	6952	1480	1588	999	1327	2319	2409
Pong	-21	15	13	19	15	18	20	20
Road Runner	12	7845	5641	9107	9615	15565	9811	15532
Seaquest	68	42055	683	774	661	618	434	426
Up N Down	533	11693	3350	15982	3546	7600	4761	6857

Effect sizes and generality. Across suites, the same drop-in change produces consistent gains while holding the planner and policy fixed. On HumanoidBench, TD-MPC2+SAM improves mean return by **+89.1%**. On high-DoF DMC, we observe an average per-task relative gain of **+20.6%** (or **+12.5%** when aggregating returns across tasks) with the largest boost on `humanoid_run`. On Atari-100k, TWISTER+SAM achieves a **+27.6%** mean human-normalised performance improvement. These effects are consistent with flatter world-model minima producing more stable rollouts.

Value–prediction error. To test whether flatter minima translate into better planning targets, we compare model-estimated returns $\hat{V}^\pi(s_0)$ with Monte-Carlo returns $V^\pi(s_0)$ under matched initial states, frozen policies, and horizons, and report $\mathbb{E}[\,|V^\pi(s_0) - \hat{V}^\pi(s_0)|\,]$ per task and seed (Fig. 3). TD-MPC2+SAM consistently reduces this value–estimation gap on HumanoidBench and high-DoF DMC, with the largest decreases on locomotion-heavy tasks where compounding bias is most pronounced. The reductions appear mid-training and persist to 2M steps, which aligns with our theory in Sec. 4 that lowering first-order sharpness tightens the value-gap bound.

Where to apply SAM in TD-MPC2. We ablate the attachment point of SAM within TD-MPC2 (App. Tab. 7). Applying SAM *only* to the *dynamics* loss yields the largest and most reliable gains and lowers value error. Applying SAM to reward or value heads is detrimental in our setting because adversarial perturbations corrupt critic targets and planning costs. Policy-side SAM can help on some tasks, but its effect is secondary. These findings provide further evidence supporting the approach to regularize the model that mediates rollouts, not the scalar targets they produce.

Statistical significance. Per-task improvements on HumanoidBench are assessed with one-tailed paired t-tests across seeds (alternative: SAM $>$ baseline). Appendix Table 3 reports significant gains on multiple tasks, including `balance_hard`/`balance_simple` (p=0.002), `run` (p=0.032), and `pole` (p=0.024). The remaining tasks trend positive but do not reach α=0.05 with n=4 and higher variance.

Compute considerations. Vanilla SAM adds one inner ascent and one outer descent per update. In our implementation this is roughly a $1.7\times$ wall-clock multiplier at fixed batch sizes with modest memory overhead, and it does not change environment interaction. If wall-clock is constrained, practitioners can apply SAM intermittently (every k model updates), anneal the inner step frequency late in training, or cache shared activations to amortize the second backward pass.

6 Limitations

Our evaluation focuses on simulation (HumanoidBench, DMC, Atari-100k) and on two agents (TD-MPC2 and TWISTER), suggesting but not proving broader generality; future work will test

real robots, stronger dynamics shift, and additional MBRL methods. Results use four seeds for HumanoidBench/DMC and five for Atari-100k with one-tailed paired (t)-tests (App. Table 3); some tasks do not reach conventional significance at ($\alpha = 0.05$). We keep TD-MPC2's default, relatively short MPC horizon; longer horizons remain to be explored. The theory treats mini-batches as approximately independent; extending to standard mixing assumptions could strengthen guarantees. Our robustness diagnostics (leading Hessian eigenvalues and value–prediction error) are informative but cover a subset of tasks (App. Tab. 8, Fig. 3). Performance is somewhat sensitive to the SAM radius (ρ); we select one value per suite with two task-specific exceptions, and vanilla SAM increases training wall-clock by about ($1.7\times$) at fixed batch sizes (App. G); adaptive or intermittent variants may reduce tuning and cost. To isolate the effect of flatness we keep baseline hyperparameters as published and vary only (ρ). For Atari we use TWISTER, so absolute scores may differ from prior reports; we emphasize the relative improvements from adding SAM.

7 Conclusion

We studied sharpness-aware training for world-models in MBRL and showed that a simple, drop-in SAM objective on the dynamics loss improves planning without changing the planner or policy. Our analysis links first-order sharpness to both value-estimation error and the suboptimality of the model-optimal policy. In particular, Thm. 2 implies that flatter minima reduce model bias and tighten policy-performance guarantees. Empirically, these insights translate into stronger control. TD-MPC2+SAM lifts HumanoidBench mean return by **+89.1%**. We observe consistent gains on high-DoF DMC and broad improvements on Atari-100k with TWISTER while holding architectures and hyperparameters fixed apart from the SAM radius ρ. Ablations indicate that targeting the *dynamics* loss is the key lever. The added cost is modest in practice. Vanilla SAM introduces one inner ascent and one outer descent per update, which we measure as roughly a $1.7\times$ training-time multiplier at fixed batch sizes, with no change in environment steps. In terms of practice, ρ is the main knob and mild tuning suffices. SAM is complementary to data augmentation, explicit smoothness control, and ensemble-based uncertainty and can be layered with these tools. Taken together, our theory and experiments support a clear message. Encouraging the world-model to converge to flatter minima is a principled and practical way to reduce model bias and improve downstream policies in model-based RL.

8 Future Work

We see three immediate directions. (i) Develop *MBRL-specific* sharpness-aware training—objectives and schedules tailored to dynamics learning and planning. (ii) Validate the approach on *real robots*, studying reliability under latency, safety limits, and non-stationary dynamics. (iii) *Compare* against methods that may implicitly enforce flat minima (e.g., ensembles, noise/augmentation, Lipschitz constraints), measuring both sharpness and control performance.

Acknowledgments

Shrinivas is supported by a research grant from Stack AV. The authors would like to thank Prof. Nathan McNeese for providing access to their lab during the paper draft and and Rhea Basappa for providing feedback to an earlier draft of this paper. We would like to thank Pittsburgh Supercomputer Center for providing access to their HPC resources.

References

[1] M. Andriushchenko and N. Flammarion. Towards understanding sharpness-aware minimization. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 639–668. PMLR, 17–23 Jul 2022.

[2] K. Asadi, D. Misra, and M. L. Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

[3] C. Baek, J. Z. Kolter, and A. Raghunathan. Why is SAM robust to label noise? In *The Twelfth International Conference on Learning Representations*, 2024.

[4] D. Bahri, H. Mobahi, and Y. Tay. Sharpness-aware minimization improves language model generalization. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7360–7371, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap. Distributional policy gradients. In *International Conference on Learning Representations*. OpenReview.net, 2018.

[6] M. Burchi and R. Timofte. Learning transformer-based world models with contrastive predictive coding. In *The Thirteenth International Conference on Learning Representations*, 2025.

[7] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[8] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 4754–4765, 2018.

[9] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel. Model-based reinforcement learning via meta-policy optimization. In A. Billard, A. Dragan, J. Peters, and J. Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 617–629. PMLR, 29–31 Oct 2018.

[10] J. Du, H. Yan, J. Feng, J. T. Zhou, L. Zhen, R. S. M. Goh, and V. Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022.

[11] G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In G. Elidan, K. Kersting, and A. Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017.

[12] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

[13] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062. PMLR, 09–15 Jun 2019.

[14] P. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018.

[16] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

[17] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[18] D. Hafner, J. Pasukonis, J. Ba, and T. P. Lillicrap. Mastering diverse domains through world models. *CoRR*, abs/2301.04104, 2023.

[19] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

[20] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, Jan. 1997.

[21] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. In A. Globerson and R. Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press, 2018.

[22] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 12498—-12509, 2019.

[23] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[24] N. Jiang. A note on loss functions and error compounding in model-based reinforcement learning, 2024.

[25] Y. Jiang*, B. Neyshabur*, H. Mobahi, D. Krishnan, and S. Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

[26] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.

[27] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21810–21823, 2020.

[28] M. Kim, D. Li, S. X. Hu, and T. Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11148–11161. PMLR, 17–23 Jul 2022.

[29] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11761–11771, 2019.

[30] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1179–1191, 2020.

[31] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations (ICLR)*, 2018.

[32] J. Kwon, J. Kim, H. Park, and I. K. Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. *arXiv preprint arXiv:2102.11600*, 2021.

[33] H. Lee, H. Cho, H. Kim, D. Gwak, J. Kim, J. Choo, S.-Y. Yun, and C. Yun. Plastic: Improving input and label plasticity for sample efficient reinforcement learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

[34] H. K. Lee and S. W. Yoon. Flat reward in policy parameter space implies robust reinforcement learning. In *Proceedings of the 13th International Conference on Learning Representations (ICLR 2025)*, 2025.

[35] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[36] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[37] D. A. McAllester. Pac-bayesian model averaging. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, page 164–170, New York, NY, USA, 1999. Association for Computing Machinery.

[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, NY, USA, 20–22 Jun 2016. PMLR.

[39] D. Oikonomou and N. Loizou. Sharpness-aware minimization: General analysis and improved rates. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial reinforcement learning. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2817–2826. PMLR, 06–11 Aug 2017.

[41] H. Rangwani, S. K. Aithal, M. Mishra, and V. B. Radhakrishnan. Escaping saddle points for effective generalization on class-imbalanced data. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[42] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[44] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 10 2017.

[45] J. M. Springer, V. Nagarajan, and A. Raghunathan. Sharpness-aware minimization enhances feature quality via balanced learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] M. Sung, S. H. Karumanchi, A. Gahlawat, and N. Hovakimyan. Robust model based reinforcement learning using $\mathcal{L}_1$ adaptive control. In *The Twelfth International Conference on Learning Representations*, 2024.

[47] Z. Wei, J. Zhu, and Y. Zhang. Sharpness-aware minimization alone can improve adversarial robustness. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023.

[48] D. Wu, S.-T. Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[49] C. Xiao, Y. Wu, C. Ma, D. Schuurmans, and M. Müller. Learning to combat compounding-error in model-based reinforcement learning, 2020.

[50] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25476–25488. Curran Associates, Inc., 2021.

[51] X. Zhang, R. Xu, H. Yu, H. Zou, and P. Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *CVPR*, pages 20247–20257, 2023.

[52] Y. Zhang, H. He, J. Zhu, H. Chen, Y. Wang, and Z. Wei. On the duality between sharpness-aware minimization and adversarial training. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 59024–59041. PMLR, 21–27 Jul 2024.

[53] Y. Zhou, Y. Qu, X. Xu, and H. Shen. Imbsam: A closer look at sharpness-aware minimization in class-imbalanced recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11345–11355, October 2023.

[54] Z. Zhou, L. Li, P. Zhao, P.-A. Heng, and W. Gong. Class-conditional sharpness-aware minimization for deep long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3499–3509, June 2023.

[55] Y. Zou, K. Kawaguchi, Y. Liu, J. Liu, M.-L. Lee, and W. Hsu. Towards robust out-of-distribution generalization bounds via sharpness. In *International Conference on Learning Representations (ICLR)*, 2024.

A Notation Table

Table 2: Summary of Notation

Symbol	Description
$\mathcal{S}$	Training dataset of transitions (fixed replay buffer)
s, s_0, s_t	State, initial state, state at time t
a, a_t	Action, action at time t
θ	Parameters of the world-model
d	Dimensionality of model parameters θ
$\mathcal{L}_{\mathcal{S}}(\theta)$	Empirical loss of the model on dataset $\mathcal{S}$ with parameters θ
$L(\theta; \pi)$	True (population) one-step model error under policy π
$\hat{L}(\theta; \pi)$	Empirical one-step model error under policy π (on samples from d^π)
ρ	Radius of the neighborhood for sharpness calculation / SAM perturbation
ϵ	Perturbation vector for model parameters
$R_\rho^{(1)}(\theta)$	First-order sharpness of training loss $\mathcal{L}_{\mathcal{S}}(\theta)$
$R_\rho^{(1)}(\theta; \pi)$	First-order sharpness of true model error $L(\theta; \pi)$ (theory context)
$L_{\mathcal{S}}^{\mathrm{SAM}}(\theta)$	Sharpness-Aware Minimization objective function
λ	Weight decay coefficient (for SAM or general regularization)
M	True MDP (Markov Decision Process)
$\hat{M}_\theta$	Learned model of the MDP, parameterized by θ
$P(\cdot\|s, a)$	True transition probability function
$\hat{P}_\theta(\cdot\|s, a)$	Learned transition probability function
$r(s, a)$	True reward function
$\hat{r}_\theta(s, a)$	Learned reward function (if part of the model)
π	A policy $\pi : \mathcal{S} \to \mathcal{A}$
π^*	Optimal policy in the true environment M
$\hat{\pi}^*$	Optimal policy in the learned model $\hat{M}_\theta$
$V^\pi(s_0)$	True expected discounted return of policy π from s_0
$\hat{V}^\pi(s_0)$	Expected discounted return of policy π estimated by model $\hat{M}_\theta$
γ	Discount factor
$d^\pi(s, a)$	Discounted state-action occupancy measure for policy π
n	Number of samples (e.g., to estimate $\hat{L}(\theta; \pi)$ or size of $\mathcal{S}$)
δ	Confidence parameter (e.g., $1 - \delta$ probability)
C	Concentrability coefficient: $\max_{(s,a)} \frac{d^{\pi^*}(s,a)}{d^{\hat{\pi}^*}(s,a)}$
M_{loss} (in bounds)	Upper bound on the per-sample one-step model error value
$\Omega(d, n, \rho, \delta)$	Model complexity/capacity term in generalization bounds

B Proofs of Theoretical Results

In this section, we provide the proofs for Thm. 1 and Thm. 2 presented in Sec. 4. The proofs leverage standard results from learning theory and model-based reinforcement learning.

B.1 Preliminaries for Proofs

We first recall two key results that will be used in the proofs.

Simulation Lemma. For any policy π, discount factor γ, and a world-model $\hat{M}_\theta$ whose one-step population model error under π is $L(\theta; \pi) = \mathbb{E}_{(s,a)\sim d^\pi}[\|P(\cdot|s, a) - \hat{P}_\theta(\cdot|s, a)\|_1]$, the difference between the true value $V^\pi(s_0)$ and the model-estimated value $\hat{V}^\pi(s_0)$ is bounded (assuming rewards $r_t \in [0, R_{max}]$):

$$|V^\pi(s_0) - \hat{V}^\pi(s_0)| \leq \frac{\gamma R_{max}}{(1-\gamma)^2} L(\theta; \pi). \tag{4}$$

If rewards are part of the model error (e.g., bounded prediction error ϵ_r per step), the term $L(\theta;\pi)$ in the bound effectively incorporates both transition and reward errors. For simplicity and consistency with the main text where model error $\hat{L}$ directly contributes to the value gap scaled by $\gamma/(1-\gamma)^2$, we'll use a form of the simulation lemma where We define $L(\theta;\pi) := \mathbb{E}_{(s,a)\sim d^\pi}[\text{err}(s,a)]$ leads to:

$$|V^\pi(s_0) - \hat{V}^\pi(s_0)| \leq \frac{\gamma}{(1-\gamma)^2}\mathbb{E}_{(s,a)\sim d^\pi}[\text{err}(s,a)]. \tag{5}$$

We will denote $\mathbb{E}_{(s,a)\sim d^\pi}[\text{err}(s,a)]$ as $L(\theta;\pi)$, representing the true one-step prediction error of the model under policy π.

Bound on Population Model Error. Given an empirical model error $\hat{L}(\theta;\pi)$ calculated on a training set of size n, its first-order sharpness $R_\rho^{(1)}(\theta;\pi)$ (of the true model error landscape $L(\theta;\pi)$), a maximum per-sample loss M_{loss}, and model parameters $\theta \in \mathbb{R}^d$, the true population model error $L(\theta;\pi)$ can be bounded. Specifically, with probability at least $1-\delta$:

$$L(\theta;\pi) \lesssim \hat{L}(\theta;\pi) + R_\rho^{(1)}(\theta;\pi) + \sqrt{\frac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta), \tag{6}$$

where $\Omega(d,n,\rho,\delta)$ is a model complexity term. The $\lesssim$ indicates an approximation or bound that may hide constants. This type of bound often arises from PAC-Bayesian analysis or uniform convergence arguments applied to sharpness-aware contexts.

B.2 Proof of Thm. 1 (Return-Estimation Gap)

Theorem 3 (Return-Estimation Gap Restated, cf. Thm. 1)**.** *For any policy π and discount factor $\gamma \in (0,1)$, with probability at least $1-\delta$ over n i.i.d. training samples for the model error estimation:*

$$\left|V^\pi(s_0) - \hat{V}^\pi(s_0)\right| \;\leq\; \frac{\gamma}{(1-\gamma)^2}\left[\hat{L}(\theta;\pi) + R_\rho^{(1)}(\theta;\pi) + \sqrt{\tfrac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta)\right].$$

Proof. We start with the Simulation Lemma (Eq. 5):

$$|V^\pi(s_0) - \hat{V}^\pi(s_0)| \leq \frac{\gamma}{(1-\gamma)^2}L(\theta;\pi).$$

Now, we substitute the upper bound for the population model error $L(\theta;\pi)$ from Equation 6. This bound holds with probability at least $1-\delta$:

$$L(\theta;\pi) \lesssim \hat{L}(\theta;\pi) + R_\rho^{(1)}(\theta;\pi) + \sqrt{\frac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta).$$

Plugging this into the simulation lemma bound, we get:

$$\left|V^\pi(s_0) - \hat{V}^\pi(s_0)\right| \;\leq\; \frac{\gamma}{(1-\gamma)^2}\left[\hat{L}(\theta;\pi) + R_\rho^{(1)}(\theta;\pi) + \sqrt{\tfrac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta)\right].$$

This completes the proof. The term $\hat{L}(\theta;\pi)$ is the empirical one-step model error under policy π, and $R_\rho^{(1)}(\theta;\pi)$ is the first-order sharpness of the true model error landscape $L(\theta;\pi)$. □

B.3 Proof of Theorem 2 (Performance Gap)

Theorem 4 (Performance Gap Restated, cf. Theorem 2)**.** *Let π^* be the optimal policy in the true environment and $\hat{\pi}^*$ be the policy that is optimal according to the learned model $\hat{M}_\theta$. If $C = \max_{(s,a)} \frac{d^{\pi^*}(s,a)}{d^{\hat{\pi}^*}(s,a)}$ is the concentrability coefficient measuring the distribution mismatch between π^* and $\hat{\pi}^*$:*

$$\left|V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0)\right| \leq \frac{\gamma(1+C)}{(1-\gamma)^2}\left[\hat{L}(\theta;\hat{\pi}^*) + R_\rho^{(1)}(\theta;\hat{\pi}^*) + \sqrt{\tfrac{M_{loss}}{n}} + \Omega(d,n,\rho,\delta)\right].$$

Proof. Since π^* is the optimal policy in the true environment M, we have $V^{\pi^*}(s_0) \geq V^{\hat{\pi}^*}(s_0)$. Therefore, the performance gap is:

$$|V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0)| = V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0). \tag{7}$$

We can decompose this difference as:

$$V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0) = [V^{\pi^*}(s_0) - \hat{V}^{\pi^*}(s_0)] + [\hat{V}^{\pi^*}(s_0) - \hat{V}^{\hat{\pi}^*}(s_0)] + [\hat{V}^{\hat{\pi}^*}(s_0) - V^{\hat{\pi}^*}(s_0)]. \tag{8}$$

Applying the Simulation Lemma (Equation 5) to the first and third terms:

$$|V^{\pi^*}(s_0) - \hat{V}^{\pi^*}(s_0)| \leq \frac{\gamma}{(1-\gamma)^2} L(\theta; \pi^*) \tag{9}$$

$$|\hat{V}^{\hat{\pi}^*}(s_0) - V^{\hat{\pi}^*}(s_0)| \leq \frac{\gamma}{(1-\gamma)^2} L(\theta; \hat{\pi}^*) \tag{10}$$

Since $\hat{\pi}^*$ is the optimal policy in the learned model $\hat{M}_\theta$, we have $\hat{V}^{\hat{\pi}^*}(s_0) \geq \hat{V}^{\pi^*}(s_0)$. Therefore, the middle term in Equation 8 is non-positive:

$$\hat{V}^{\pi^*}(s_0) - \hat{V}^{\hat{\pi}^*}(s_0) \leq 0. \tag{11}$$

Substituting these into Equation 8 and using the bounds from 9 and 10:

$$\begin{aligned} V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0) &\leq \frac{\gamma}{(1-\gamma)^2} L(\theta; \pi^*) + 0 + \frac{\gamma}{(1-\gamma)^2} L(\theta; \hat{\pi}^*) \\ &= \frac{\gamma}{(1-\gamma)^2} [L(\theta; \pi^*) + L(\theta; \hat{\pi}^*)]. \end{aligned} \tag{12}$$

Using the definition of the concentrability coefficient $C = \max_{(s,a)} \frac{d^{\pi^*}(s,a)}{d^{\hat{\pi}^*}(s,a)}$, we can bound the model error under π^* in terms of the model error under $\hat{\pi}^*$. Assuming the model error is an expectation over the state-action distribution, $L(\theta; \pi) = \mathbb{E}_{(s,a)\sim d^\pi}[\text{err}(s, a; \theta)]$:

$$L(\theta; \pi^*) = \sum_{s,a} d^{\pi^*}(s,a) \text{err}(s,a;\theta) \leq \sum_{s,a} C \cdot d^{\hat{\pi}^*}(s,a) \text{err}(s,a;\theta) = C \cdot L(\theta; \hat{\pi}^*).$$

Substituting this into Equation 12:

$$\begin{aligned} V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0) &\leq \frac{\gamma}{(1-\gamma)^2} [C \cdot L(\theta; \hat{\pi}^*) + L(\theta; \hat{\pi}^*)] \\ &= \frac{\gamma(1+C)}{(1-\gamma)^2} L(\theta; \hat{\pi}^*). \end{aligned} \tag{13}$$

Finally, we substitute the upper bound for the population model error $L(\theta; \hat{\pi}^*)$ from Equation 6, which holds with probability at least $1 - \delta$:

$$L(\theta; \hat{\pi}^*) \lesssim \hat{L}(\theta; \hat{\pi}^*) + R_\rho^{(1)}(\theta; \hat{\pi}^*) + \sqrt{\frac{M_{loss}}{n}} + \Omega(d, n, \rho, \delta).$$

Plugging this into Equation 13 yields the final result:

$$\left|V^{\pi^*}(s_0) - V^{\hat{\pi}^*}(s_0)\right| \leq \frac{\gamma(1+C)}{(1-\gamma)^2} \left[\hat{L}(\theta; \hat{\pi}^*) + R_\rho^{(1)}(\theta; \hat{\pi}^*) + \sqrt{\tfrac{M_{loss}}{n}} + \Omega(d, n, \rho, \delta) \right].$$

This completes the proof. The terms $\hat{L}(\theta; \hat{\pi}^*)$ and $R_\rho^{(1)}(\theta; \hat{\pi}^*)$ refer to the empirical one-step model error and the first-order sharpness of the true model error landscape $L(\theta; \hat{\pi}^*)$, evaluated under the model-optimal policy $\hat{\pi}^*$. □

C On the Correlation Between Loss Sharpness and Model Lipschitz Continuity

In the context of model-based reinforcement learning, the smoothness of the learned dynamics model $\hat{M}_\theta$ (parameterized by θ) can be crucial for robust planning and generalization. Lipschitz continuity

is a common measure of such smoothness. Here, we briefly explore a tangential connection between the sharpness of the model's training loss and the Lipschitz properties of the model. While our main theoretical results (Theorems 1 and 2) directly incorporate first-order sharpness $R_\rho^{(1)}$, understanding its relationship with other smoothness measures like Lipschitz constants can provide additional intuition.

Let $\hat{f}_\theta(x)$ represent a component of our learned model (e.g., the next-state prediction function, where $x = (s, a)$ is a state-action pair) and $f(x)$ be the corresponding true environment dynamics. The model is trained to minimize a loss, commonly mean squared error (MSE) for continuous state predictions:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim P_x}\left[(f(x) - \hat{f}_\theta(x))^2\right],$$

where P_x is the distribution of training data.

[Lipschitz Continuity] A function $g : \mathcal{X} \to \mathcal{Y}$ (where $\mathcal{Y}$ could be $\mathbb{R}^k$) is Lipschitz continuous with constant L_g if for all $x_1, x_2 \in \mathcal{X}$:

$$\|g(x_1) - g(x_2)\|_\mathcal{Y} \leq L_g \|x_1 - x_2\|_\mathcal{X}.$$

[Second-Order Sharpness (Spectral Norm of Hessian)] A common measure of sharpness related to the curvature of the loss landscape $\mathcal{L}(\theta)$ at a point θ is the spectral norm of its Hessian matrix:

$$S_2(\theta) = \|\nabla_\theta^2 \mathcal{L}(\theta)\|_2.$$

While our main text focuses on first-order sharpness $R_\rho^{(1)}(\theta)$ (Eq. 1), $S_2(\theta)$ provides another perspective on local curvature. (Note: A related concept, $R_\rho^{(2)}(\theta)$, could involve maximizing $\lambda_{\max}(\nabla^2_{\theta\theta}\mathcal{L}(\theta + \epsilon))$ over a neighborhood, but $S_2(\theta)$ is the local Hessian norm at θ.)

We can show a relationship between this sharpness measure and the properties of $\hat{f}_\theta$. The Hessian of the MSE loss $\mathcal{L}(\theta)$ is:

$$\begin{aligned}
\nabla_\theta^2 \mathcal{L}(\theta) &= \nabla_\theta \left(\mathbb{E}_{x \sim P_x}\left[-2(f(x) - \hat{f}_\theta(x))\nabla_\theta \hat{f}_\theta(x)\right]\right) \\
&= 2\mathbb{E}_{x \sim P_x}\left[\nabla_\theta \hat{f}_\theta(x)\left(\nabla_\theta \hat{f}_\theta(x)\right)^\top - (f(x) - \hat{f}_\theta(x))\nabla_\theta^2 \hat{f}_\theta(x)\right].
\end{aligned}$$

Taking spectral norms and using the triangle inequality:

$$\begin{aligned}
S_2(\theta) = \|\nabla_\theta^2 \mathcal{L}(\theta)\|_2 &\leq 2\left\|\mathbb{E}_{x \sim P_x}\left[\nabla_\theta \hat{f}_\theta(x)\left(\nabla_\theta \hat{f}_\theta(x)\right)^\top\right]\right\|_2 \\
&\quad + 2\left\|\mathbb{E}_{x \sim P_x}\left[(f(x) - \hat{f}_\theta(x))\nabla_\theta^2 \hat{f}_\theta(x)\right]\right\|_2 \\
&\leq 2\mathbb{E}_{x \sim P_x}\left[\|\nabla_\theta \hat{f}_\theta(x)\|_2^2\right] + 2\sup_x |f(x) - \hat{f}_\theta(x)| \cdot \mathbb{E}_{x \sim P_x}\left[\|\nabla_\theta^2 \hat{f}_\theta(x)\|_2\right].
\end{aligned} \tag{14}$$

The term $\|\nabla_\theta \hat{f}_\theta(x)\|_2$ represents the sensitivity of the model's output to changes in its parameters θ at a given input x. The term $\|\nabla_\theta^2 \hat{f}_\theta(x)\|_2$ represents the curvature of the model function $\hat{f}_\theta$ with respect to its parameters.

If the model $\hat{f}_\theta(x)$ itself is "smooth" with respect to its inputs x (i.e., has a small Lipschitz constant $L_{\hat{f}_x} = \sup_x \|\nabla_x \hat{f}_\theta(x)\|_2$) and also with respect to its parameters (i.e., $\|\nabla_\theta \hat{f}_\theta(x)\|_2$ and $\|\nabla_\theta^2 \hat{f}_\theta(x)\|_2$ are bounded, perhaps by constants $L_{\hat{f}_\theta}$ and $H_{\hat{f}_\theta}$ respectively), then the sharpness $S_2(\theta)$ tends to be smaller, especially if the model error $|f(x) - \hat{f}_\theta(x)|$ is also small.

For instance, if we assume that for our parameterization, $\sup_x \|\nabla_\theta \hat{f}_\theta(x)\|_2 \leq K_1$ and $\sup_x \|\nabla_\theta^2 \hat{f}_\theta(x)\|_2 \leq K_2$, then:

$$S_2(\theta) \leq 2K_1^2 + 2\|f - \hat{f}_\theta\|_\infty K_2.$$

This inequality suggests that models whose output is less sensitive to parameter changes (smaller K_1, K_2) and that fit the data well (small $\|f - \hat{f}_\theta\|_\infty$) tend to reside in regions of lower second-order sharpness.

While this provides a connection to a specific type of sharpness ($S_2(\theta)$), the first-order sharpness $R_\rho^{(1)}$ targeted by SAM is related (as discussed in Section 4, $R_\rho^{(1)}$ is an indicator of local flatness). Methods that explicitly regularize the Lipschitz constant of the learned model (e.g., [2]) aim to enforce smoothness directly. SAM, by seeking flat minima of the training loss, indirectly promotes solutions where the loss (and thus often the model predictions) do not change drastically with small parameter perturbations. This implies a form of robustness that is conceptually related to having a small Lipschitz constant with respect to parameters, although SAM achieves this through a different mechanism than direct Lipschitz regularization. Exploring the interplay between SAM-induced flatness and explicit Lipschitz regularization of the world-model could be an interesting direction for future research, potentially leading to even more robust and generalizable models.

D t-statistic tests on HumanoidBench

We assess whether adding SAM improves return using one-tailed paired t-tests across n=4 seeds per task (null: no gain; alternative: *SAM* > *baseline*). Negative t indicates higher mean return with SAM under this convention.

Table 3: One-tailed paired t-tests: TD-MPC2+SAM vs. TD-MPC2 on HumanoidBench (4 seeds). Significance at α=0.05 in **bold**. The overall p-value using Fisher's method is $p = 0.0059$

Task	t-statistic	p-value	Significant?
`balance_hard`	-5.83	$\mathbf{0.002}$	**Yes**
`balance_simple`	-8.31	$\mathbf{0.002}$	**Yes**
`stair`	-1.64	0.170	No
`walk`	-1.69	0.184	No
`run`	-3.78	$\mathbf{0.032}$	**Yes**
`stand`	-1.87	0.148	No
`sit_simple`	-1.77	0.128	No
`sit_hard`	-2.20	0.108	No
`crawl`	1.69	0.160	No
`pole`	-3.19	$\mathbf{0.024}$	**Yes**
`hurdle`	-1.05	0.356	No

E SAM radius (ρ) ablation

We sweep ρ and observe a unimodal response: too small under-regularizes, too large over-perturbs, with a broad optimum in the middle.

E.1 Suite-level choices used in the paper

Table 4: ρ values used for main results.

Suite / Task set	ρ	Rationale
HumanoidBench (most tasks)	1.25×10^{-3}	Best overall trade-off across tasks
`sit_simple`, `sit_hard`	5.0×10^{-3}	Stabilizes sit tasks
High-DoF DMC	2.5×10^{-5}	Prefers smaller steps in state-based control
Atari-100k (TWISTER)	1.0×10^{-3}	Robust on pixel-based discrete control

E.2 Per-task sensitivity examples

F SAM on different TD-MPC2 components

Unless otherwise noted, SAM is applied only to the *dynamics* loss. Ablating the attachment point on `humanoid_run` (2M steps; 4 seeds):

Table 5: Return on `humanoid_run` vs. ρ settings (mean $\pm$ SEM over 4 seeds). Settings are ordered from smaller→larger ρ; **best** in bold.

Setting (ρ)	1×10^{-3}	5×10^{-6}	5×10^{-7}	1×10^{-7}	5×10^{-8}	1×10^{-8}
Episode return	348 ± 28	445 ± 56	$\mathbf{492 \pm 34}$	454 ± 17	439 ± 21	414 ± 21

Table 6: Score Bank Heist vs. ρ settings (mean $\pm$ SEM over 5 seeds).

Setting (ρ)	1×10^{-1}	1×10^{-2}	1×10^{-3}	1×10^{-4}	1×10^{-5}
Score	281 ± 277	714 ± 311	$\mathbf{886 \pm 445}$	785 ± 277	781 ± 213

Table 7: Where to apply SAM in TD-MPC2 (episode return; mean $\pm$ SEM).

Variant	Apply SAM to	Episode return
TD-MPC2 (baseline)	—	301 ± 11
TD-MPC2 + SAM (ours)	**Dynamics**	$\mathbf{484 \pm 19}$
TD-MPC2 + SAM	Reward & Value	10 ± 3
TD-MPC2 + SAM	Policy	463 ± 49

G Compute overhead

Vanilla SAM adds one inner ascent and one outer descent per model update. In our implementation this is a $\sim$**1.7**$\times$ training-time multiplier at fixed batch sizes, with modest memory overhead; environment interaction (sample complexity) is unchanged. When wall-clock is tight, apply SAM intermittently (every kth model update), anneal the inner step frequency late in training, or reuse cached activations to reduce the second backward pass cost.

H Max Hessian eigenvalue

We approximate the leading eigenvalue(s) of the Hessian of the *dynamics* loss via power iteration on replay mini-batches after training. Lower values indicate flatter minima.

Table 8: Max Hessian eigenvalue $\lambda_{\max}$ (arbitrary units; same scale across rows). Lower is flatter.

Task (2M steps)	TD-MPC2	TD-MPC2 + SAM
`humanoid_run`	141.6	**99.5**
`humanoid_walk`	92.1	**82.3**
`dog_run`	80.3	**46.5**
`dog_walk`	69.5	**40.1**
`dog_trot`	53.8	**31.6**

I HumanoidBench Results

Table 9: HumanoidBench returns (mean ± SEM over 4 seeds) for TD-MPC2 and TD-MPC2+SAM.

Metric	balance_hard	balance_simple	stair	walk	run	stand	sit_simple	sit_hard	crawl	pole	hurdle
TD-MPC2 Episode Reward	48 ± 6	28 ± 8	66 ± 6	644 ± 281	67 ± 8	639 ± 240	515 ± 187	508 ± 298	896 ± 53	207 ± 35	51 ± 12
TD-MPC2 w/ SAM Episode Reward	82 ± 10	145 ± 27	77 ± 12	885 ± 44	302 ± 124	870 ± 58	773 ± 223	843 ± 64	846 ± 26	273 ± 22	71 ± 36

Table 10: High-DoF DMC results at 2M environment steps (mean ± SEM over n=4 seeds). Bold is better.

Method	`humanoid_run`	`humanoid_walk`	`dog_run`	`dog_walk`	`dog_trot`
TD-MPC2	301 ± 11	883 ± 13	428 ± 39	887 ± 46	891 ± 21
TD-MPC2 w/ SAM	$\mathbf{484 \pm 19}$	$\mathbf{901 \pm 4}$	$\mathbf{552 \pm 17}$	$\mathbf{957 \pm 6}$	$\mathbf{920 \pm 14}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim that applying SAM to the world-model in MBRL leads to flatter minima and improved downstream policy performance, supported by theory and experiments on HumanoidBench with TD-MPC2. These claims are reflected in the paper's theoretical analysis (Section 4) and experimental results (Section 5).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations, such as the current scope of empirical evaluation (primarily TD-MPC2 on HumanoidBench) and the task-dependent nature of the SAM hyperparameter ρ, are discussed in the Experiments (Section 5) and Conclusion (Section 7).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be

used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main theoretical results (Theorems 1 and 2) are presented with their assumptions in Section 4. Detailed proofs are provided in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper details the experimental setup in Section 5, including the benchmark (HumanoidBench), base algorithm (TD-MPC2), SAM integration, and key hyperparameters like ρ values and training steps. Further details on TD-MPC2 hyperparameters are referenced to its original publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided with the submission/

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the environment, tasks, algorithm modifications, training duration, seeds, and key SAM hyperparameters. Further details for TD-MPC2 are referenced.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The learning curves in Figure 4, 2 report mean returns over 4 seeds with shaded regions representing the standard error of the mean (SEM), as stated in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper does not currently detail the specific compute resources (e.g., GPU type, number of GPUs, approximate training time per run). This information would be added to an appendix if the paper is accepted or if required by submission guidelines.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research focuses on algorithmic improvements in simulated environments and does not involve human subjects, sensitive data, or applications with immediate ethical concerns that would violate the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper currently focuses on the technical contributions. A discussion of broader impacts (e.g., improved robot capabilities, potential misuse of advanced AI) is not included but could be added.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The research involves training reinforcement learning agents in simulated environments. The models produced are specific to these simulations and do not pose a high direct risk for misuse in the manner of large pre-trained generative models or scraped datasets.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites TD-MPC2, HumanoidBench, and SAM, which are the primary existing assets. HumanoidBench is based on MuJoCo. Licenses for these assets are typically permissive for academic research (e.g., Apache 2.0 for TD-MPC2 and HumanoidBench, MuJoCo is now open-source).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper primarily introduces a modification to an existing algorithm (TD-MPC2 by integrating SAM) and evaluates it on an existing benchmark (HumanoidBench). No new datasets or standalone software packages are introduced as primary contributions. If code is released, it will be documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: The research does not involve crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The research does not involve human subjects, so IRB approval is not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Large language models were not used as a core component of the proposed methodology or experiments. Any LLM usage was restricted to assisting with writing, editing, or formatting, which does not impact the scientific contributions of the paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.