On Training in Imagination

Nadav Timor[*]
Weizmann Institute of Science

Ravid Shwartz-Ziv
New York University

Micah Goldblum
Columbia University

Yann LeCun
New York University
AMI Labs

David Harel
Weizmann Institute of Science

Abstract

State-of-the-art model-based reinforcement learning methods train policies on imagined rollouts. These rollouts are trajectories generated by a learned dynamics model and are scored by a learned reward model, but without querying the true environment during policy updates. We study this training paradigm by quantifying how errors in learned dynamics and reward models affect returns and policy optimization. First, we extend the analysis of Asadi et al. [2018b] to MDPs with learned reward models, and derive the optimal sample allocation—the ratio of dynamics samples to reward samples that minimizes a bound on return error under power-law scaling assumptions. We identify lower Lipschitz constants of the learned dynamics, reward, and policy as a representation desideratum that tightens this bound, and we connect this perspective to the temporal-straightening objective of Wang et al. [2026]. Second, we examine how policy optimization with REINFORCE tolerates noisy rewards, which are often cheaper to obtain. We show that zero-mean reward noise leaves the gradient estimator unbiased and adds at most a variance term that decreases with the number of rollouts. This introduces a practical tradeoff: given a fixed budget, should one buy more rollouts with cheaper but noisier rewards, or fewer rollouts with more expensive but less noisy rewards? We reduce this choice to a one-dimensional optimization problem and characterize the optimum.

1 Introduction

In *training in imagination*, the policy is trained on rollouts generated by a learned dynamics model and scored by a learned reward model, with no environment interaction during the policy update step itself. Recent state-of-the-art instantiations include Dreamer 3 [Hafner et al., 2025a], trained across diverse control tasks with a single configuration, and Dreamer 4 [Hafner et al., 2025b], which extends the paradigm to long-horizon offline control. Schrittwieser et al. [2020] earlier instantiate a closely related paradigm in board games and Atari, learning dynamics, reward, and value jointly.

These recent results provide strong empirical evidence that training in imagination can be effective on challenging control tasks. Existing simulation-lemma-style bounds [Kearns and Singh, 2002, Asadi et al., 2018b], however, do not assign independently controllable coefficients to dynamics-model and reward-model error, nor do they say how a sample budget should be split between dynamics samples and the typically more expensive reward samples (e.g., human preference labels in reinforcement learning from human feedback, or expert evaluation in robotics). The optimal trade-off between the two has not been theoretically characterized.

Four questions about this paradigm remain open. The first is error attribution: how much of the return gap comes from dynamics-model error versus reward-model error? The second is representation

[*]Corresponding author: nadav.timor@weizmann.ac.il.

Preprint.

properties: what properties of the learned representations and the maps acting on them tighten the return-error bound? The third is budget allocation: given a fixed sample budget, how should it be split between dynamics transitions and reward annotations? The fourth is reward fidelity: how does REINFORCE tolerate noisy or biased reward annotations, and when is it preferable to buy many cheap noisy annotations rather than fewer accurate ones?

Our contribution. This paper treats the learned reward model as a separate source of error with its own sample budget, distinct from the learned dynamics, and quantifies the resulting attribution, allocation, and noise-tolerance trade-offs under Lipschitz and power-law assumptions.

1. **Error attribution.** Lemma 1 extends Asadi et al. [2018b] by replacing the assumed ground-truth reward with a learned reward model, and gives a Lipschitz-based decomposition of the return gap with separable, independently controllable dynamics-error and reward-error coefficients.
2. **Representation desiderata.** Corollary 1 shows that the dynamics-error coefficient in Equation (1) is monotone non-decreasing in the Lipschitz constants L_f, L_r, L_π of the learned dynamics, reward, and policy, identifying lower Lipschitz constants of the learned models as a representation desideratum. Proposition 1 couples this perspective to the temporal-straightening objective of Wang et al. [2026], upper-bounding its curvature loss by a function of the Lipschitz constant of the latent velocity map.
3. **Budget allocation.** Theorem 1, under power-law error scaling for the dynamics and reward errors, gives a closed-form expression for the optimal ratio of dynamics samples to reward samples in terms of the power-law exponents, the per-sample costs, and the Lipschitz coefficient inherited from Lemma 1.
4. **Reward fidelity.** Theorem 2 shows that the multi-trajectory REINFORCE estimator under additive zero-mean reward noise is unbiased with bounded variance inflation; Corollary 2 reduces the optimal-fidelity allocation problem to a one-dimensional minimization in the per-rollout annotation cost; and Proposition 2 formalizes systematic reward bias as a gradient bias that trajectory averaging cannot remove.

Empirical evaluations of the assumptions and predictions of these results appear in Sections 3.1, 4.1 and 4.2.

Notation. We write $\mathcal{M} = (\mathcal{S}, \mathcal{A}, f, r, \gamma)$ for a Markov decision process (MDP) with state space $\mathcal{S} \subseteq \mathbb{R}^{d_s}$, action space $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, deterministic dynamics f, reward r, and discount $\gamma \in [0, 1)$. Starting from an initial state s_0, a policy π generates a trajectory by $a_t = \pi(s_t)$ and $s_{t+1} = f(s_t, a_t)$. We write $J(\pi, \mathcal{M}) := \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$ for the discounted return of π in $\mathcal{M}$. Hats denote estimated quantities. In particular, $\hat{f}$ is the learned dynamics, $\hat{r}$ the learned reward, and $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{f}, \hat{r}, \gamma)$ is the MDP obtained by replacing f and r with $\hat{f}$ and $\hat{r}$. $\varepsilon_{\mathrm{dyn}} := \sup_{s,a} \|\hat{f}(s, a) - f(s, a)\|$ and $\varepsilon_{\mathrm{rew}} := \sup_{s,a} |\hat{r}(s, a) - r(s, a)|$ are the worst-case model errors. Throughout, $\|\cdot\|$ denotes the Euclidean norm.

2 Related work

Training a policy on rollouts from a learned model of the environment dates back to the Dyna architecture of Sutton [1990], which interleaves real-environment transitions with updates on imagined transitions drawn from a learned dynamics model. Model-based policy optimization [Janner et al., 2019] uses short imagined rollouts from an ensemble dynamics model to augment policy updates. More recently, Hafner et al. [2025a] train across diverse control tasks with a single learned world model and reward predictor, and Hafner et al. [2025b] train agents inside scalable learned world models, calling this process "training in imagination". These works treat the learned dynamics model and the learned reward predictor as a single coupled object and tune it empirically; Section A reviews the broader latent-world-model lineage. Unlike the training-in-imagination lineage of Hafner et al. [2025b], this paper decomposes return error into separate, independently controllable dynamics-model and reward-model terms (Lemma 1), derives a closed-form split of a single sample budget between dynamics transitions and reward annotations (Theorem 1), and characterizes the policy

gradient inside this paradigm under both zero-mean noise and bias in the learned reward (Theorem 2 and Proposition 2).

Simulation return-error bounds have a long history in reinforcement learning theory, beginning with the simulation lemma of Kearns and Singh [2002], which bounds the value gap between a true and approximate Markov decision process in terms of one-step transition and reward errors. Closest to our setting, Asadi et al. [2018b] bound multi-step prediction error in model-based reinforcement learning under Lipschitz assumptions on the dynamics and policy, but they assume access to the ground-truth reward, so reward-model error never enters their bound. Section A surveys subsequent refinements of these bounds and value-aware model learning. Unlike Asadi et al. [2018b], Lemma 1 carries a learned reward model through the analysis and produces an explicit reward-error term alongside the dynamics-error term with independently controllable coefficients, which is the structural ingredient that makes a budget split between dynamics and reward data well-posed.

Allocating sample budget between dynamics samples and reward samples sits at the intersection of reward-aware data collection and neural scaling laws. Reward-free exploration separates a reward-agnostic data-collection phase from later reward-conditioned planning, with sample-complexity guarantees that hold uniformly over downstream reward functions [Jin et al., 2020]. Neural scaling laws fit power-law decays of loss in data and parameters [Kaplan et al., 2020] and, in the compute-optimal regime, split a single training budget between model parameters and tokens [Hoffmann et al., 2022]. Section A discusses related work on active observation, simulation budget allocation, and scaling laws specific to reinforcement learning and world-model pre-training. Unlike Hoffmann et al. [2022], Theorem 1 splits a single sample budget between two heterogeneous data streams—dynamics transitions and reward annotations—whose errors obey separately fitted power-law exponents, and yields a closed-form ratio in those exponents, the unit costs, and the Lipschitz coefficient inherited from Lemma 1.

Policy-gradient analysis under noisy rewards begins with the REINFORCE estimator of Williams [1992]. A line of work studies robustness to reward corruption: Zhang et al. [2021] analyze adversarial corruption in which an ε-fraction of episodes have their rewards or transitions arbitrarily modified and develop estimators with provable robustness guarantees. Cai et al. [2025] treat asymmetric verifier noise with false positives and false negatives in reinforcement learning with verifiable rewards. A parallel literature documents Goodhart-style overoptimization of learned reward models, in which the gap between proxy and gold rewards grows with optimization budget [Gao et al., 2023], a systematic failure mode rather than zero-mean reward noise. Section A reviews the policy-gradient and variance-reduction lineage and the broader reward-modeling literature. Unlike the adversarial setting of Zhang et al. [2021], Theorem 2 and Corollary 2 treat zero-mean i.i.d. reward noise as a per-rollout fidelity cost, which reduces annotation-budget allocation to a one-dimensional problem. Proposition 2 addresses reward bias separately, showing that any non-zero reward-bias gradient survives trajectory averaging.

The choice of latent representation, and the regularity of the maps acting on it, governs how reliably an imagined rollout tracks reality. LeCun et al. [2022] advocates joint-embedding predictive architectures, in which prediction takes place in a learned latent space rather than at the level of raw observations. Wang et al. [2026] propose a temporal-straightening loss that encourages consecutive latent differences along a rollout to be parallel, so that long-horizon predictions in latent space follow a near-linear trajectory. On the regularization side, Miyato et al. [2018] introduce spectral normalization, which controls the Lipschitz constant of a neural network by normalizing the spectral norm of each weight matrix; Section A discusses related joint-embedding instantiations and Lipschitz-regularization mechanisms. These representation-learning and Lipschitz-regularization proposals are motivated by stability or representational quality, and their connection to long-horizon policy value is left implicit. Unlike Wang et al. [2026] and Miyato et al. [2018], Corollary 1 and Proposition 1 couple these representation-learning desiderata to an explicit return-error coefficient, showing that the dynamics-error coefficient is monotone in the Lipschitz constants of the learned dynamics, reward, and policy and that the temporal-straightening loss is upper-bounded by a function of the latent-velocity-map Lipschitz constant.

3 Properties of representations for training in imagination

What makes a representation of the system useful for training in imagination? LeCun et al. [2022] hypothesized that designing the dynamics, reward, and policy models to operate on latent states z_t that capture a higher-level representation of the system—in place of raw observations s_t and past actions a_t—may improve prediction and planning across different time horizons. However, what properties such a representation should have has remained an open question. Lemma 1 and Corollary 1 give one answer to this open question: representations that lower the Lipschitz constants of the learned models tighten our bound on return error.

Similar to Asadi et al. [2018b], we assume that the dynamics f, reward r, and policy π satisfy the following Lipschitz conditions:

- f is L_f-Lipschitz: $\|f(s,a) - f(s',a')\| \leq L_f(\|s - s'\| + \|a - a'\|)$ for all s, s', a, a'.
- r is L_r-Lipschitz: $|r(s,a) - r(s',a')| \leq L_r(\|s - s'\| + \|a - a'\|)$ for all s, s', a, a'.
- π is L_π-Lipschitz: $\|\pi(s) - \pi(s')\| \leq L_\pi \|s - s'\|$ for all s, s'.

Lemma 1 (Simulation error decomposition)**.** *Let $\mathcal{M}, \hat{\mathcal{M}}$ be* MDP*s sharing $(\mathcal{S}, \mathcal{A}, \gamma)$ with deterministic dynamics $f, \hat{f}$ and rewards $r, \hat{r}$. Assume $\gamma L_f(1 + L_\pi) < 1$. Then for any L_π-Lipschitz policy π,*

$$\left| J(\pi, \mathcal{M}) - J(\pi, \hat{\mathcal{M}}) \right| \leq \frac{1}{1-\gamma} \varepsilon_{\text{rew}} + \frac{\gamma L_r (1 + L_\pi)}{(1-\gamma)(1 - \gamma L_f (1 + L_\pi))} \varepsilon_{\text{dyn}}. \tag{1}$$

Proof. See Section B.1. □

Corollary 1 (Lipschitz constants control the dynamics-error coefficient)**.** *Under the hypothesis $\gamma L_f(1 + L_\pi) < 1$, the coefficient $\frac{\gamma L_r (1+L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}$ of ε_{dyn} in Equation* (1) *is non-decreasing in each of L_f, L_r, and L_π. Hence, at fixed ε_{dyn} and ε_{rew}, lowering any of L_f, L_r, L_π tightens the bound in Equation* (1) *on return error.*

Corollary 1 formalizes this: representations that lower the Lipschitz constants of the learned models L_f, L_r, L_π tighten the bound in Equation (1) on return error at fixed $\varepsilon_{\text{dyn}}, \varepsilon_{\text{rew}}$.

A simple implementation may define the latent state z as an encoding of the current observation $z = \phi(s)$. The Lipschitz constants are then defined by comparing the outputs of the learned models at nearby latent states, rather than at nearby raw observations. For example, the Lipschitz constant of the dynamics model is then $\|f(\phi(s), a) - f(\phi(s'), a')\| \leq L_f\big(\|\phi(s) - \phi(s')\| + \|a - a'\|\big)$. The reward and policy maps are handled analogously, with absolute value replacing the output norm for scalar rewards. Higher-order representations are also possible: latent states z_t may encode the full trajectory $(s_0, a_0, \ldots, s_t, a_{t-1})$, states or activations of other learned models, and so on.

This Lipschitz perspective also connects to the temporal straightening objective of Wang et al. [2026], which compares consecutive latent differences generated by the dynamics model and encourages these differences to point in the same direction. To make the connection explicit, let $\mathcal{Z}$ denote the latent state space and write $z_t = \phi(s_t)$ for the latent state associated with observation s_t. Following Wang et al. [2026], we write f for the learned dynamics model on $\mathcal{Z}$, so that $z_{t+1} = f(z_t)$ along a latent rollout. Thus, in this discussion of temporal straightening, f no longer denotes the observation-level dynamics on $\mathcal{S}$.

Definition 1 (temporal straightening curvature loss, adapted from Wang et al. [2026])**.** Define the latent velocity map $v(z) := f(z) - z$. For a latent rollout $z_{t+1} = f(z_t)$ and any t such that $v(z_t) \neq 0$ and $v(z_{t+1}) \neq 0$, define

$$\mathcal{L}_{\text{curv}}(t) := 1 - \frac{v(z_t)^\top v(z_{t+1})}{\|v(z_t)\| \|v(z_{t+1})\|}.$$

On observed transitions, $v(\phi(s_t))$ approximates $\phi(s_{t+1}) - \phi(s_t)$, so changes in v along a latent rollout measure how smoothly the latent state moves along the trajectory. The temporal straightening objective maximizes the cosine similarity between $v(z_t)$ and $v(z_{t+1})$, or equivalently minimizes the curvature loss in Definition 1. The relevant Lipschitz quantity is not the Lipschitz constant of f itself, but the Lipschitz constant of the latent velocity map v.

Proposition 1 (temporal straightening from a Lipschitz latent velocity map)**.** *Let $z_{t+1} = f(z_t)$ be a latent rollout generated by the dynamics model, and define $v(z) := f(z) - z$. Assume v is ε-Lipschitz on the latent states visited by the rollout, with $0 < \varepsilon < 1$. For any t such that $v(z_t) \neq 0$ and $v(z_{t+1}) \neq 0$, the temporal straightening curvature loss from Definition 1 satisfies*

$$\mathcal{L}_{\mathrm{curv}}(t) \leq \frac{\varepsilon^2}{2(1-\varepsilon)}. \tag{2}$$

Proof. See Section B.2. □

Proposition 1 shows that making the latent velocity map slowly varying makes consecutive latent differences nearly parallel, which directly lowers the temporal straightening loss. Thus, minimizing the Lipschitz constant of v minimizes the upper bound in Equation (2).

Two caveats apply. First, a representation that lowers the Lipschitz constants might also increase $\varepsilon_{\mathrm{dyn}}$. The bound in Lemma 1 only tightens if the decrease in the dynamics-error coefficient outweighs the increase in $\varepsilon_{\mathrm{dyn}}$. Second, that bound assumes $\gamma L_f(1 + L_\pi) < 1$, and the behavior when $\gamma L_f(1 + L_\pi) \geq 1$ remains an open question for future work.

3.1 Numerical illustration of Lemma 1

We empirically test the inequality in Equation (1). For each test configuration, define the realized return gap $\mathrm{LHS} := |J(\pi, \mathcal{M}) - J(\pi, \hat{\mathcal{M}})|$ and the bound's right-hand side $\mathrm{RHS} := (1-\gamma)^{-1}\varepsilon_{\mathrm{rew}} + \frac{\gamma L_r(1+L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}\varepsilon_{\mathrm{dyn}}$, evaluated with the configuration's analytical Lipschitz constants L_f, L_r, L_π, discount factor γ, and realized per-step errors $\varepsilon_{\mathrm{dyn}}, \varepsilon_{\mathrm{rew}}$. We report the ratio $R := \mathrm{LHS}/\mathrm{RHS}$. The bound holds whenever $R \leq 1$, and smaller R indicates a looser bound. Each configuration consists of an MDP, an L_π-Lipschitz evaluation policy, and a perturbed model pair with realized per-step errors $\varepsilon_{\mathrm{dyn}}, \varepsilon_{\mathrm{rew}}$. We test on two benchmarks. The synthetic benchmark uses globally Lipschitz f and r, so the hypotheses of Lemma 1 hold by construction. The Linear–Quadratic–Gaussian (LQG) benchmark uses a quadratic reward that is only locally Lipschitz, testing Lemma 1 on a bounded operating domain. Section B.4 describes the per-configuration construction. Across $n = 525$ configurations (150 synthetic, 375 LQG) the bound holds on every configuration. The per-benchmark medians are $R^{\mathrm{med}}_{\mathrm{synth}} = 0.0035$ and $R^{\mathrm{med}}_{\mathrm{LQG}} = 0.034$, the pooled median is 0.015, and the per-benchmark maxima are $R^{\max}_{\mathrm{synth}} = 0.9995$ and $R^{\max}_{\mathrm{LQG}} = 0.999$. For per-benchmark empirical CDF (ECDF) shapes and full implementation details, see Section B.4. Every configuration satisfies the bound, but the typical bound is loose: at the median it overshoots by $1/R^{\mathrm{med}}_{\mathrm{LQG}} \approx 29\times$ on LQG and $1/R^{\mathrm{med}}_{\mathrm{synth}} \approx 286\times$ on the synthetic benchmark (factor $\approx 65\times$ at the pooled median). Both benchmarks use deterministic dynamics; extending the calibration to stochastic dynamics or unbounded operating domains would require revisiting the assumptions of Lemma 1 and is left to future work.

4 The optimal sample allocation to minimize return error

We distinguish between two types of samples: dynamics-transition samples and reward samples. A dynamics-transition sample $(s, a, f(s, a))$ consists of a state s, an action a, and the resulting next state $f(s, a)$. A reward sample $(s, a, r(s, a))$ consists of a state, an action, and the resulting reward $r(s, a)$. Let N_{dyn} and N_{rew} denote the numbers of dynamics-transition and reward samples, and let c_{dyn} and c_{rew} denote their per-sample costs. These costs may reflect any incurred costs, including environment interaction, annotation, and training on the samples. We reuse the symbols $\varepsilon_{\mathrm{dyn}}$ and $\varepsilon_{\mathrm{rew}}$ to denote the error levels achievable when the sample counts are N_{dyn} and N_{rew}, according to standard power laws

$$\varepsilon_{\mathrm{dyn}}(N_{\mathrm{dyn}}) = A_d \cdot N_{\mathrm{dyn}}^{-\alpha}, \qquad \varepsilon_{\mathrm{rew}}(N_{\mathrm{rew}}) = A_r \cdot N_{\mathrm{rew}}^{-\beta} \tag{3}$$

where $\alpha, \beta > 0$ are exponents fit from data, and A_d, A_r are constants. For each budget $B = c_{\mathrm{dyn}}N_{\mathrm{dyn}} + c_{\mathrm{rew}}N_{\mathrm{rew}}$, let $(N^*_{\mathrm{dyn}}, N^*_{\mathrm{rew}})$ denote a minimizer of the bound in Equation (1), and let $\varepsilon^*_{\mathrm{dyn}} := \varepsilon_{\mathrm{dyn}}(N^*_{\mathrm{dyn}})$ and $\varepsilon^*_{\mathrm{rew}} := \varepsilon_{\mathrm{rew}}(N^*_{\mathrm{rew}})$ be the dynamics and reward errors at those minimizing sample counts. We study the optimal ratio of dynamics samples to reward samples, $\frac{N^*_{\mathrm{dyn}}}{N^*_{\mathrm{rew}}}$, as $B \to \infty$.

Theorem 1 (Optimal dynamics-to-reward sample ratio)**.** *If Equation* (3) *holds with exponents* $\alpha, \beta > 0$ *and the bound of Lemma 1 holds, then the minimizing sample counts* $(N^*_{\mathrm{dyn}}, N^*_{\mathrm{rew}})$ *under the constraint* $c_{\mathrm{dyn}} N_{\mathrm{dyn}} + c_{\mathrm{rew}} N_{\mathrm{rew}} = B$ *satisfy*

$$\frac{N^*_{\mathrm{dyn}}}{N^*_{\mathrm{rew}}} := \frac{\alpha}{\beta} \cdot \frac{\gamma L_r (1 + L_\pi)}{1 - \gamma L_f (1 + L_\pi)} \cdot \frac{c_{\mathrm{rew}}}{c_{\mathrm{dyn}}} \cdot \frac{\varepsilon^*_{\mathrm{dyn}}}{\varepsilon^*_{\mathrm{rew}}}. \tag{4}$$

Proof. See Section B.3. □

Theorem 1 gives the optimal sample ratio $N^*_{\mathrm{dyn}}/N^*_{\mathrm{rew}}$ in terms of the error ratio $\varepsilon^*_{\mathrm{dyn}}/\varepsilon^*_{\mathrm{rew}}$, and shows that these two ratios are proportional. The multiplier in Equation (4) uses the global Lipschitz constants L_f, L_r assumed by Lemma 1, so it is an upper bound rather than an equality. Section 4.2 measures, on the same configurations, how much smaller the realized value-function sensitivities are than these global constants, and how close the resulting prediction of $N^*_{\mathrm{dyn}}/N^*_{\mathrm{rew}}$ becomes. For practitioners who design systems that train policies in imagination, changes in sample costs or planning horizon affect the optimal sample ratio as follows.

Sample costs. Costs enter through $c_{\mathrm{dyn}} N_{\mathrm{dyn}} + c_{\mathrm{rew}} N_{\mathrm{rew}} = B$ and the fitted error curves in Equation (3); in Equation (4), they appear as $c_{\mathrm{rew}}/c_{\mathrm{dyn}}$. Lowering c_{rew} lowers this factor, so $N^*_{\mathrm{dyn}}/N^*_{\mathrm{rew}}$ decreases and the allocation shifts toward reward samples. Lowering c_{dyn} raises the factor, so $N^*_{\mathrm{dyn}}/N^*_{\mathrm{rew}}$ increases and the allocation shifts toward dynamics-transition samples. However, since the optimal sample ratio also includes $\varepsilon^*_{\mathrm{dyn}}/\varepsilon^*_{\mathrm{rew}}$, these statements give the direction of the cost effect rather than a complete allocation rule by themselves.

Planning horizon. The discounting effect is the familiar one: smaller γ gives less weight to later rewards. As γ decreases, Equation (1) becomes tighter, and the dynamics multiplier in Equation (4), $\gamma L_r(1 + L_\pi)/(1 - \gamma L_f(1 + L_\pi))$, decreases. The optimal allocation therefore shifts toward reward samples, as expected for a shorter-horizon objective in which dynamics errors have fewer discounted steps to affect future rewards. A similar effect arises from lowering the Lipschitz constants of the learned models. By Corollary 1, lowering any of L_f, L_r, L_π tightens Equation (1) and decreases the dynamics multiplier in Equation (4). The optimal allocation therefore shifts toward reward samples, since dynamics errors then compound less strongly along rollouts.

4.1 Numerical experiments: scaling of dynamics and reward errors

We estimate how fast each model learns with more data and whether the dynamics and reward models learn at the same rate. To this end, we conduct an experiment that tests the power-law scaling assumptions in Equation (3) and estimates the exponents α and β for a particular architecture and environment. In this experiment, we estimate the dynamics and reward errors as a function of the number of training samples N_{dyn} and N_{rew}, respectively, and fit the standard power-law scaling laws from Equation (3). Figure 1 shows the fitted scaling laws, which are consistent with the power-law assumptions in Theorem 1. It measures $\varepsilon_{\mathrm{dyn}}$ and $\varepsilon_{\mathrm{rew}}$ on a fixed held-out set $\mathcal{D}_{\mathrm{val}}$ of transitions (s, a, s', r) in a synthetic continuous-control environment whose dynamics and reward function are defined by frozen, randomly-initialized 2-layer ReLU MLPs, and whose students $\hat{f}$ and $\hat{r}$ use the same architecture as the teachers. Specifically, $\varepsilon_{\mathrm{dyn}}(N_{\mathrm{dyn}}) := \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{(s,a,s',r) \in \mathcal{D}_{\mathrm{val}}} \|\hat{f}(s, a) - s'\|^2$ and $\varepsilon_{\mathrm{rew}}(N_{\mathrm{rew}}) := \frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{(s,a,s',r) \in \mathcal{D}_{\mathrm{val}}} (\hat{r}(s, a) - r)^2$, where $\hat{f}$ and $\hat{r}$ are trained on N_{dyn} dynamics-transition samples and N_{rew} reward samples drawn independently of $\mathcal{D}_{\mathrm{val}}$. We use $\varepsilon_{\mathrm{dyn}}$ and $\varepsilon_{\mathrm{rew}}$ as practical surrogates for the sup-norm errors in Lemma 1. The fitted laws are $\varepsilon_{\mathrm{dyn}}(N_{\mathrm{dyn}}) = 0.34\, N_{\mathrm{dyn}}^{-0.11}$ with $R^2 = 0.954$ and $\varepsilon_{\mathrm{rew}}(N_{\mathrm{rew}}) = 90.4\, N_{\mathrm{rew}}^{-0.96}$ with $R^2 = 0.997$. For bootstrap standard errors, 95% bootstrap confidence intervals, and full implementation details, see Section C.2. The exponent ratio $0.96/0.11 \approx 9$ is the central empirical observation: reward error decays nearly an order of magnitude faster per decade of training data than dynamics error. This ratio of exponents is consistent with $\hat{r}$ predicting a scalar while $\hat{f}$ predicts a d_s-dimensional next state, and its exact value depends on the dimensions and architectures used here.

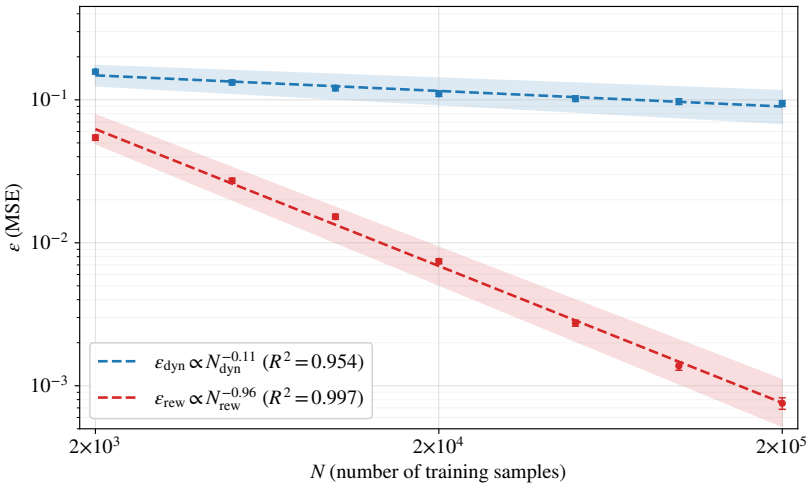


Figure 1: Dynamics and reward errors obey distinct power-law scaling laws. Reward error decays roughly $9\times$ faster per decade of training data than dynamics error ($\approx \frac{0.96}{0.11}$). Theorem 1 uses this ratio to characterize the optimal fraction of transitions that should carry a reward annotation.

4.2 Empirical evaluation of Equation (4)

Equation (4) predicts $N^*_{\text{dyn}}/N^*_{\text{rew}}$ as the product of two factors: the proportionality $N^*_{\text{dyn}}/N^*_{\text{rew}} \propto \varepsilon^*_{\text{dyn}}/\varepsilon^*_{\text{rew}}$ and a multiplier that depends on γ, L_f, L_r, L_π, $c_{\text{rew}}/c_{\text{dyn}}$, and α/β. We test the two factors separately: whether the proportionality holds when the multiplier is replaced by realized value-function sensitivities, and how loose the multiplier is when instantiated with the global Lipschitz constants assumed by Lemma 1. Define the log-ratio residual $\ell := \log(N^*_{\text{dyn}}/N^*_{\text{rew}}) - \log(\varepsilon^*_{\text{dyn}}/\varepsilon^*_{\text{rew}})$, so $|\ell| \le \log 3$ means predicted and realized ratios agree within a factor of 3. The realized value-function sensitivities S_f, S_r and the per-configuration construction are defined in Section B.5. Replacing the global constants L_f, L_r with the realized sensitivities S_f, S_r recovers the predicted ratio. We test this recovery on configurations whose value function V^π has a known parametric form: linear, $\tanh$, $\sin$, and a separate quadratic-value control group that isolates the looseness from using $\sup$-norm overestimates in place of the realized sensitivities. The linear configurations achieve median $|\ell| = 0$ and Spearman rank correlation $\rho = 1$ between predicted and realized ratios; this case is an algebraic consistency check rather than an empirical test, since a linear V^π forces the realized sensitivities to match the analytical ratio. The empirical content comes from the nonlinear configurations: $\tanh$ and $\sin$ give median $|\ell| = 0.054$. On the control group, substituting $\sup$-norm overestimates for the realized sensitivities inflates the median residual by roughly an order of magnitude. Figure 2 plots predicted against realized sample ratios $N^*_{\text{dyn}}/N^*_{\text{rew}}$ for each group. Instantiating the multiplier with the analytical global Lipschitz constants of Lemma 1, by contrast, overshoots realized ratios by about three orders of magnitude. On the LQG configurations every configuration's ℓ is positive, with median $\ell = 7.585$ (a factor $\exp(7.585) \approx 1968$ between predicted and realized ratios). Figure 4 shows the per-configuration distribution of ℓ. The contraction regime, sample sizes, statistical methodology, and a separate ratio isolating the dynamics-coefficient contribution to the multiplier are reported in Section B.5.

The proportionality $N^*_{\text{dyn}}/N^*_{\text{rew}} \propto \varepsilon^*_{\text{dyn}}/\varepsilon^*_{\text{rew}}$ in Equation (4) therefore carries the predictive content, while the multiplier built from global Lipschitz constants is loose at the order-of-magnitude scale. Recovering the predicted ratio requires evaluating V^π at perturbed models (the construction in Section B.5), which may be impractical at scale.

5 Cost-effective learning from noisy rewards

The reward-noise analysis in this section is independent of the allocation results in Section 4: the unbiasedness statement for REINFORCE under zero-mean reward noise applies to any stationary MDP

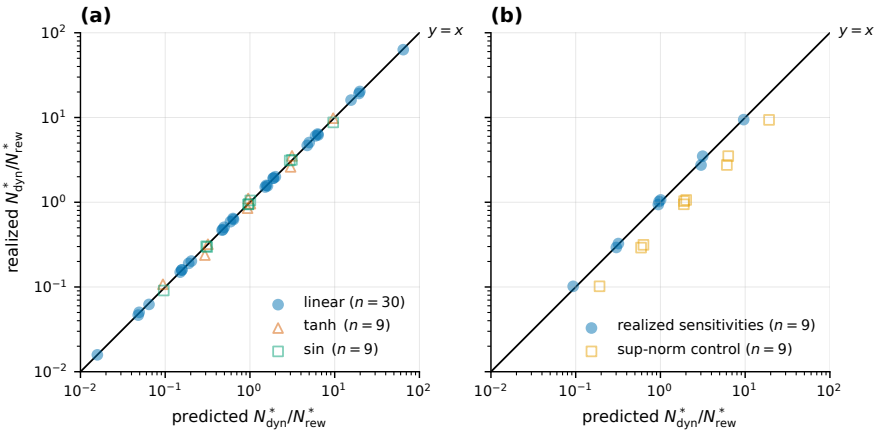


Figure 2: Predicted and realized sample ratios $N^*_{\mathrm{dyn}}/N^*_{\mathrm{rew}}$ agree when global Lipschitz constants are replaced by realized value-function sensitivities S_f, S_r. The linear, $\tanh$, and $\sin$ groups in panel (a) concentrate on the diagonal $y = x$, while the $\sup$-norm control in panel (b) lies systematically below it—using the global $\sup$-norm in place of the realized sensitivities reintroduces the looseness shown in Figure 4. See Section B.5 for plotting and methodology details.

satisfying the standard policy-gradient assumptions, regardless of how dynamics and reward samples are allocated.

We study policy optimization under noisy rewards through the classical REINFORCE estimator introduced by Williams [1992]. In this section let π_θ be a differentiable policy, fix a finite horizon H, and fix a discount factor $\gamma \in [0, 1)$. Define $J_H(\pi, \mathcal{M}) := \mathbb{E}[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)]$, where the expectation is over trajectories generated by executing π in $\mathcal{M}$. Define the discounted cumulative reward from time t onward as $G_t := \sum_{t'=t}^{H-1} \gamma^{t'-t} r(s_{t'}, a_{t'})$, so that G_0 is the discounted return of the full trajectory and $J_H(\pi, \mathcal{M}) = \mathbb{E}[G_0]$. Define the finite-horizon policy gradient by $g_H := \nabla_\theta J_H(\pi_\theta, \mathcal{M})$. Define the noisy counterpart of G_t as $\hat{G}_t := \sum_{t'=t}^{H-1} \gamma^{t'-t} \hat{r}_{t'}$, where $\hat{r}_t$ denotes the observed reward at time t. The REINFORCE estimator of g_H computed from a single trajectory is $\hat{g}^{(1)} := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \hat{G}_t$. To estimate g_H, sample $K \geq 1$ independent trajectories by executing π_θ in $\mathcal{M}$. For each $k \in \{1, \ldots, K\}$, let $\hat{g}^{(k)}$ denote the REINFORCE estimator computed on the kth trajectory, and define $\hat{g} := \frac{1}{K} \sum_{k=1}^{K} \hat{g}^{(k)}$. The estimators $\hat{g}^{(1)}, \ldots, \hat{g}^{(K)}$, $\hat{g}$, and their noise-free counterparts are vector-valued. To measure their dispersion with a single scalar, we use the natural generalization of scalar variance obtained by adding coordinate variances: for $z = (z_1, \ldots, z_d)$, write $\mathrm{Var}[z] := \sum_{i=1}^{d} \mathrm{Var}[z_i]$.

Theorem 2 (Finite-horizon REINFORCE under noisy rewards)**.** *Assume* $W_H^2 := \mathbb{E}\big[\max_{0 \leq t \leq H-1} \|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2\big] < \infty$. *Suppose the rewards are observed with additive noise,* $\hat{r}_t = r(s_t, a_t) + \eta_t$*, where the noise variables* η_t *are i.i.d. with* $\mathbb{E}[\eta_t] = 0$ *and* $\mathrm{Var}[\eta_t] = \sigma_\eta^2 < \infty$*, and each* η_t *is independent of the state-action history* $((s_0, a_0), \ldots, (s_t, a_t))$*. Then* $\hat{g}$ *satisfies*

$$\mathbb{E}[\hat{g}] = g_H, \qquad \mathrm{Var}[\hat{g}] \leq \mathrm{Var}[\hat{g}]_{\eta \equiv 0} + \frac{\sigma_\eta^2 H W_H^2}{K(1-\gamma)^2}. \tag{5}$$

where $\mathrm{Var}[\hat{g}]_{\eta \equiv 0}$ *denotes the variance of the same estimator when the rewards are noise-free.*

Proof. See Section B.6. □

In many settings, practitioners can pay more per reward annotation to obtain less noisy rewards, for example by averaging multiple annotators or using a more careful annotation pipeline. We use Theorem 2 to ask how a fixed budget for reward annotations should be split between fidelity (lower noise per annotation) and quantity (more rollouts at higher noise).

Let $c > 0$ denote the per-rollout cost of acquiring reward annotations along that rollout, and define $\sigma_\eta^2(c) := \mathrm{Var}[\eta_t \mid \text{per-rollout annotation cost equals } c] \in [0, \infty)$ as the variance of the reward noise η_t from Theorem 2 when reward annotations are acquired at per-rollout cost c. We assume that $\sigma_\eta^2 \colon (0, \infty) \to [0, \infty)$ is measurable. Given a budget $B > 0$ for reward annotations, the number of independent rollouts that fit in the budget at fidelity c is $K = B/c$.

Corollary 2 (Optimal fidelity for noise-induced variance)**.** *Define $\Phi(c) := c\,\sigma_\eta^2(c)$. Under the assumptions of Theorem 2 with $\mathrm{Var}[\eta_t] = \sigma_\eta^2(c)$ and $K = B/c$, the upper bound on the noise-induced excess variance from Theorem 2 equals $\Phi(c)\, H\, W_H^2/(B\,(1-\gamma)^2)$. Consequently, any $c^* \in \arg\min_{c>0} \Phi(c)$ minimizes this upper bound over $c > 0$.*

Proof. Substituting $K = B/c$ into the upper bound from Theorem 2 gives $\sigma_\eta^2(c)\, H\, W_H^2/(K\,(1-\gamma)^2) = \Phi(c)\, H\, W_H^2/(B\,(1-\gamma)^2)$. The prefactor $H\, W_H^2/(B\,(1-\gamma)^2)$ is non-negative and does not depend on c, so the upper bound and $\Phi(c)$ share their minimizers over $c > 0$. □

Corollary 2 reduces the choice of fidelity for reward annotations to minimizing $\Phi(c)$. We illustrate the consequences with three examples of how the noise variance $\sigma_\eta^2(c)$ depends on the cost c, summarized in Figure 5.

Power-law fidelity. Suppose $\sigma_\eta^2(c) = A\,c^{-p}$ for constants $A, p > 0$, so that each multiplicative increase in cost yields a fixed multiplicative reduction in noise variance. Then $\Phi(c) = A\,c^{1-p}$, which is strictly decreasing in c when $p > 1$, strictly increasing in c when $p < 1$, and constant when $p = 1$. The exponent $p = 1$ therefore separates two regimes: when $p > 1$, the bound is minimized by paying for the highest-fidelity annotations available; when $p < 1$, the bound is minimized by paying as little as possible per annotation and using the saved budget for additional rollouts.

Bounded fidelity. Suppose $\sigma_\eta^2(c) = \sigma_0^2\,(1 - c/c_{\max})$ on $(0, c_{\max}]$ for constants $\sigma_0^2, c_{\max} > 0$, modeling a setting in which annotations have variance σ_0^2 as $c \to 0$ and become noise-free at the finite cost $c_{\max}$. Then $\Phi(c) = \sigma_0^2\, c\,(1 - c/c_{\max})$ is a downward parabola in c that is maximized at $c = c_{\max}/2$, attains the value 0 at $c = c_{\max}$, and approaches 0 as $c \to 0$. Both extremes of the cost range therefore minimize the bound, and intermediate fidelities are strictly worse.

Irreducible noise floor. Suppose $\sigma_\eta^2(c) = \sigma_{\text{floor}}^2 + A/c$ for constants $\sigma_{\text{floor}}^2, A > 0$, modeling a setting in which no level of spending can drive the noise variance below σ_{floor}^2. Then $\Phi(c) = \sigma_{\text{floor}}^2\, c + A$ is strictly increasing in c, so the bound is minimized by $c \to 0$: when reward noise has a floor that money cannot remove, the optimal allocation buys the cheapest annotations and relies on the $1/K$ factor in Theorem 2 to reduce variance.

Together, these cases show that the shape of $\Phi(c)$ determines whether the budget should be spent on fidelity, on quantity, or on one of the two extremes.

The preceding results assume the reward noise is zero-mean. We note that this assumption is essential: averaging over more trajectories cannot remove a systematic reward bias.

Proposition 2 (Finite-horizon REINFORCE under biased rewards)**.** *Let $b : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be a reward bias function, define $\tilde{r}(s,a) := r(s,a) + b(s,a)$, and define $\tilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, f, \tilde{r}, \gamma)$. For trajectories sampled by executing π_θ under the true dynamics f, let $\tilde{g}^{(1)}, \ldots, \tilde{g}^{(K)}$ be independent* REINFORCE *estimators computed using the biased rewards $\tilde{r}(s_t, a_t)$, and define $\tilde{g} := \frac{1}{K}\sum_{k=1}^{K} \tilde{g}^{(k)}$. Define $B_H(\theta) := \mathbb{E}\Big[\sum_{t=0}^{H-1} \gamma^t b(s_t, a_t)\Big]$, where the expectation is over trajectories generated by executing π_θ under the true dynamics f. Then*

$$\mathbb{E}[\tilde{g}] = \nabla_\theta J_H(\pi_\theta, \tilde{\mathcal{M}}) = g_H + \nabla_\theta B_H(\theta). \tag{6}$$

If $\mathrm{Var}[\tilde{g}^{(1)}] < \infty$, then

$$\mathbb{E}\big[\|\tilde{g} - g_H\|^2\big] = \frac{1}{K}\mathrm{Var}[\tilde{g}^{(1)}] + \|\nabla_\theta B_H(\theta)\|^2. \tag{7}$$

Consequently, when $\nabla_\theta B_H(\theta) \neq 0$, averaging over more trajectories reduces the variance term but does not remove the bias as an estimator of g_H.

Proof. See Section B.8. □

6 Discussion

Although the dynamics $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and reward $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ depend only on the current state and action, this does not require the underlying environment to be memoryless: any dependence on past states and actions can be absorbed into $\mathcal{S}$ by taking the state to encode a summary of the past, e.g. via recurrent neural networks [Rumelhart et al., 1986, Elman, 1990, Hochreiter and Schmidhuber, 1997, Henaff et al., 2016].

7 Acknowledgments

Nadav Timor thanks GitHub for Startups and Lambda AI for generous financial support. Micah Goldblum was supported by Dream Sports and the Google Research Award. David Harel was supported by a research grant from Magnus Konow in honour of his mother Olga Konow Rappaport.

References

Kavosh Asadi, Evan Cater, Dipendra Misra, and Michael L Littman. Equivalence between wasserstein and value-aware loss for model-based reinforcement learning. *arXiv preprint arXiv:1806.01265*, 2018a.

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International conference on machine learning*, pages 264–273. PMLR, 2018b.

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

Colin Bellinger, Rory Coles, Mark Crowley, and Isaac Tamblyn. Active measure reinforcement learning for observation cost minimization. *arXiv preprint arXiv:2005.12697*, 2020.

Xin-Qiang Cai, Wei Wang, Feng Liu, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Reinforcement learning with verifiable yet noisy rewards under imperfect verifiers. *arXiv preprint arXiv:2510.00915*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025a.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Haoyu Han and Heng Yang. Non-uniform noise-to-signal ratio in the reinforce policy-gradient estimator. *arXiv preprint arXiv:2602.01460*, 2026.

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8387–8406. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent orthogonal networks and long-memory tasks. In *International Conference on Machine Learning*, pages 2034–2042. PMLR, 2016.

Jacob Hilton, Jie Tang, and John Schulman. Scaling laws for single-agent reinforcement learning. *arXiv preprint arXiv:2301.13442*, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Xilang Huang and Seon Han Choi. An efficient simulation-based policy improvement with optimal computing budget allocation based on accumulated samples. *Electronics*, 11(7):1141, 2022.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 267–274, 2002.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002.

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

Sam Lobel and Ronald Parr. An optimal tightness bound for the simulation lemma. In *Reinforcement Learning Conference*, 2024. URL https://openreview.net/forum?id=RcoIAfiM5g.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Rémi Munos. Error bounds for approximate policy iteration. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 560–567, 2003.

Tim Pearce, Tabish Rashid, David Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. In *International Conference on Machine Learning*, pages 48542–48562. PMLR, 2025.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

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

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Erik Talvitie. Learning the reward function for a misspecified model. In *International Conference on Machine Learning*, pages 4838–4847. PMLR, 2018.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim GJ Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Robust policy gradient against strong data corruption. In *International Conference on Machine Learning*, pages 12391–12401. PMLR, 2021.

A Additional related work

This appendix collects related works that inform the setting of Section 2 but are not load-bearing for the novelty claims of Lemma 1, Theorems 1 and 2, Corollaries 1 and 2, and Propositions 1 and 2.

Latent-state world models. Beyond the works cited in Section 2, several lines of work shape the modern training-in-imagination paradigm. Deisenroth and Rasmussen [2011] fit a Gaussian-process dynamics model and propagate uncertainty through imagined trajectories under a known reward. Ha and Schmidhuber [2018] learn a latent dynamics model from pixels and train a policy primarily inside this learned model. Dreamer [Hafner et al., 2020] learns a latent-space dynamics model together with a learned reward predictor and optimizes the policy entirely on imagined latent rollouts. Schrittwieser et al. [2020] learn a value-equivalent latent model whose dynamics, reward, and value predictions are trained jointly to support planning, and Hansen et al. [2022] combine a learned latent dynamics and reward model with short-horizon planning and a learned terminal value.

Simulation-style bounds and value-aware model learning. The simulation-lemma family extends in several directions. Kakade and Langford [2002] express the difference in expected discounted return between two policies as an expectation of single-step advantages under one of them, which underpins conservative policy iteration. Munos [2003] gives L_p-style error-propagation bounds for approximate policy iteration, and Lobel and Parr [2024] recently establish optimal tightness for the simulation lemma. A parallel line of work asks whether the model loss should target value prediction rather than raw transition accuracy: value-aware model learning [Farahmand et al., 2017] replaces the next-state likelihood objective with a loss measuring the worst-case discrepancy between the true and learned dynamics on expected values over a class of value functions, Asadi et al. [2018a] show that, restricted to a 1-Lipschitz value-function class, this loss is equivalent to the Wasserstein distance between the dynamics, and Talvitie [2018] addresses how to learn the reward function on states drawn from a misspecified dynamics model.

Active observation and simulation budget allocation. Active-measure reinforcement learning chooses when to pay for an observation under explicit observation costs [Bellinger et al., 2020]. In the simulation-optimization literature, optimal computing budget allocation divides a fixed simulation budget across candidate policies to maximize the probability of selecting the best one [Huang and Choi, 2022]. Power-law fits analogous to those of Kaplan et al. [2020] and Hoffmann et al. [2022] have since been reported for single-agent reinforcement learning and for pre-training of agents and world models [Hilton et al., 2023, Pearce et al., 2025].

Policy-gradient theory and reward modeling. Beyond Williams [1992], the policy-gradient theorem of Sutton et al. [1999] and the variance-reduction analysis of Greensmith et al. [2004] provide the standard analytic toolkit invoked by Theorem 2. Han and Yang [2026] characterize how the REINFORCE noise-to-signal ratio varies non-uniformly across the parameter landscape. Reinforcement learning from human feedback trains reward models from preferences [Christiano et al., 2017, Stiennon et al., 2020], the empirical setting in which Gao et al. [2023] document Goodhart-style overoptimization.

Representation learning and Lipschitz regularization. Assran et al. [2023] provide a concrete image instantiation of joint-embedding predictive architectures, and Gouk et al. [2021] propose projection-based mechanisms for enforcing Lipschitz continuity during training, with upper bounds applicable to multiple p-norms beyond the spectral norm.

B Full proofs

B.1 Proof of Lemma 1

Lemma 1 (Simulation error decomposition)**.** *Let $\mathcal{M}, \hat{\mathcal{M}}$ be* MDP*s sharing $(\mathcal{S}, \mathcal{A}, \gamma)$ with deterministic dynamics $f, \hat{f}$ and rewards $r, \hat{r}$. Assume $\gamma L_f(1 + L_\pi) < 1$. Then for any L_π-Lipschitz policy π,*

$$\left|J(\pi, \mathcal{M}) - J(\pi, \hat{\mathcal{M}})\right| \leq \frac{1}{1-\gamma}\,\varepsilon_{\mathrm{rew}} + \frac{\gamma L_r(1 + L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}\,\varepsilon_{\mathrm{dyn}}. \tag{1}$$

Proof. Fix an L_π-Lipschitz policy π and a shared initial state $s_0 = \hat{s}_0$. Let $\{s_t\}$ denote the state trajectory generated by executing π under the true dynamics f, and $\{\hat{s}_t\}$ the trajectory under $\hat{f}$, both starting from s_0. The actions differ because the policy is evaluated at different states:

$$a_t = \pi(s_t), \qquad \hat{a}_t = \pi(\hat{s}_t).$$

At each step t:

$$r(s_t, a_t) - \hat{r}(\hat{s}_t, \hat{a}_t) = \big[r(s_t, a_t) - r(\hat{s}_t, \hat{a}_t)\big] + \big[r(\hat{s}_t, \hat{a}_t) - \hat{r}(\hat{s}_t, \hat{a}_t)\big]. \tag{8}$$

Let $L_{\mathrm{comp}} := L_f(1 + L_\pi)$. By induction:

$$\|s_t - \hat{s}_t\| \leq \varepsilon_{\mathrm{dyn}} \sum_{k=0}^{t-1} L_{\mathrm{comp}}^k$$

Base case ($t = 0$): $s_0 = \hat{s}_0$, so $\|s_0 - \hat{s}_0\| = 0$.

Inductive step: Assume $\|s_t - \hat{s}_t\| \leq \varepsilon_{\text{dyn}} \sum_{k=0}^{t-1} L_{\text{comp}}^k$.

$$
\begin{aligned}
\|s_{t+1} - \hat{s}_{t+1}\| &= \|f(s_t, a_t) - \hat{f}(\hat{s}_t, \hat{a}_t)\| \\
&\leq \|f(s_t, a_t) - f(\hat{s}_t, \hat{a}_t)\| + \|f(\hat{s}_t, \hat{a}_t) - \hat{f}(\hat{s}_t, \hat{a}_t)\| \\
&\leq L_f\big(\|s_t - \hat{s}_t\| + \|a_t - \hat{a}_t\|\big) + \varepsilon_{\text{dyn}} \\
&= L_f\big(\|s_t - \hat{s}_t\| + \|\pi(s_t) - \pi(\hat{s}_t)\|\big) + \varepsilon_{\text{dyn}} \\
&\leq L_f\big(\|s_t - \hat{s}_t\| + L_\pi\|s_t - \hat{s}_t\|\big) + \varepsilon_{\text{dyn}} \\
&\leq L_f(1 + L_\pi)\|s_t - \hat{s}_t\| + \varepsilon_{\text{dyn}} \\
&= L_{\text{comp}}\|s_t - \hat{s}_t\| + \varepsilon_{\text{dyn}} \\
&\leq L_{\text{comp}}\left(\varepsilon_{\text{dyn}} \sum_{k=0}^{t-1} L_{\text{comp}}^k\right) + \varepsilon_{\text{dyn}} \\
&= \varepsilon_{\text{dyn}} \sum_{k=0}^{t-1} L_{\text{comp}}^{k+1} + \varepsilon_{\text{dyn}} \\
&= \varepsilon_{\text{dyn}} \sum_{k=0}^{t-1} L_{\text{comp}}^{k+1} + \varepsilon_{\text{dyn}} L_{\text{comp}}^0 \\
&= \varepsilon_{\text{dyn}} \sum_{j=1}^{t} L_{\text{comp}}^{j} + \varepsilon_{\text{dyn}} L_{\text{comp}}^0 \\
&= \varepsilon_{\text{dyn}} \sum_{k=0}^{t} L_{\text{comp}}^{k}.
\end{aligned}
$$

Error from using the learned reward function: $|r(\hat{s}_t, \hat{a}_t) - \hat{r}(\hat{s}_t, \hat{a}_t)| \leq \varepsilon_{\text{rew}}$, by the definition of ε_{rew}.

Error from evaluating the true reward at different state-action pairs:

$$
\begin{aligned}
|r(s_t, a_t) - r(\hat{s}_t, \hat{a}_t)| &\leq L_r\big(\|s_t - \hat{s}_t\| + \|a_t - \hat{a}_t\|\big) \\
&= L_r\big(\|s_t - \hat{s}_t\| + \|\pi(s_t) - \pi(\hat{s}_t)\|\big) \\
&\leq L_r\big(\|s_t - \hat{s}_t\| + L_\pi\|s_t - \hat{s}_t\|\big) \\
&= L_r(1 + L_\pi)\|s_t - \hat{s}_t\| \\
&\leq L_r(1 + L_\pi)\varepsilon_{\text{dyn}} \sum_{k=0}^{t-1} L_{\text{comp}}^k.
\end{aligned}
$$

Using the definition of discounted return and Equation (8),

$$
\begin{aligned}
|J(\pi, \mathcal{M}) - J(\pi, \hat{\mathcal{M}})| &= \left| \sum_{t=0}^{\infty} \gamma^t \big(r(s_t, a_t) - \hat{r}(\hat{s}_t, \hat{a}_t) \big) \right| \\
&\leq \sum_{t=0}^{\infty} \gamma^t \left| r(s_t, a_t) - \hat{r}(\hat{s}_t, \hat{a}_t) \right| \\
&= \sum_{t=0}^{\infty} \gamma^t \left| \big(r(s_t, a_t) - r(\hat{s}_t, \hat{a}_t) \big) + \big(r(\hat{s}_t, \hat{a}_t) - \hat{r}(\hat{s}_t, \hat{a}_t) \big) \right| \\
&\leq \sum_{t=0}^{\infty} \gamma^t \left| r(s_t, a_t) - r(\hat{s}_t, \hat{a}_t) \right| + \sum_{t=0}^{\infty} \gamma^t \left| r(\hat{s}_t, \hat{a}_t) - \hat{r}(\hat{s}_t, \hat{a}_t) \right| \\
&\leq L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{t=0}^{\infty} \gamma^t \left(\sum_{k=0}^{t-1} L_{\text{comp}}^k \right) + \varepsilon_{\text{rew}} \sum_{t=0}^{\infty} \gamma^t \\
&= L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{t=0}^{\infty} \sum_{k=0}^{t-1} \gamma^t L_{\text{comp}}^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{k=0}^{\infty} \sum_{\substack{t=0 \\ k \leq t-1}}^{\infty} \gamma^t L_{\text{comp}}^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{k=0}^{\infty} \sum_{t=k+1}^{\infty} \gamma^t L_{\text{comp}}^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{k=0}^{\infty} \sum_{j=0}^{\infty} \gamma^{k+1+j} L_{\text{comp}}^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= L_r(1 + L_\pi) \varepsilon_{\text{dyn}} \sum_{k=0}^{\infty} \left(\gamma^{k+1} \sum_{j=0}^{\infty} \gamma^j \right) L_{\text{comp}}^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= \frac{\gamma L_r(1 + L_\pi)}{1-\gamma} \varepsilon_{\text{dyn}} \sum_{k=0}^{\infty} (\gamma L_{\text{comp}})^k + \frac{1}{1-\gamma} \varepsilon_{\text{rew}}
\end{aligned}
$$

Because $\gamma L_{\text{comp}} < 1$, the geometric series converges.

$$
\begin{aligned}
&= \frac{\gamma L_r(1 + L_\pi)}{1-\gamma} \varepsilon_{\text{dyn}} \left(\frac{1}{1 - \gamma L_{\text{comp}}} \right) + \frac{1}{1-\gamma} \varepsilon_{\text{rew}} \\
&= \frac{1}{1-\gamma} \varepsilon_{\text{rew}} + \frac{\gamma L_r(1 + L_\pi)}{(1-\gamma)(1 - \gamma L_f(1 + L_\pi))} \varepsilon_{\text{dyn}}
\end{aligned}
$$

□

B.2 Proof of Proposition 1

Proposition 1 (temporal straightening from a Lipschitz latent velocity map)**.** *Let $z_{t+1} = f(z_t)$ be a latent rollout generated by the dynamics model, and define $v(z) := f(z) - z$. Assume v is ε-Lipschitz on the latent states visited by the rollout, with $0 < \varepsilon < 1$. For any t such that $v(z_t) \neq 0$ and $v(z_{t+1}) \neq 0$, the temporal straightening curvature loss from Definition 1 satisfies*

$$
\mathcal{L}_{\text{curv}}(t) \leq \frac{\varepsilon^2}{2(1-\varepsilon)}. \tag{2}
$$

Proof. Fix t such that $v(z_t) \neq 0$ and $v(z_{t+1}) \neq 0$, and set

$$
a := v(z_t), \qquad b := v(z_{t+1}).
$$

Along the rollout, $a = z_{t+1} - z_t$ and $b = z_{t+2} - z_{t+1}$. Using the ε-Lipschitz condition for v gives

$$\begin{aligned}\|b - a\| &= \|v(z_{t+1}) - v(z_t)\| \\ &\leq \varepsilon \|z_{t+1} - z_t\| \\ &= \varepsilon \|a\|.\end{aligned}$$

By the triangle inequality,

$$\begin{aligned}\|b\| &= \|a + (b - a)\| \\ &\geq \|a\| - \|b - a\| \\ &\geq (1 - \varepsilon)\|a\|.\end{aligned}$$

Define

$$C := \frac{a^\top b}{\|a\|\|b\|}.$$

Then

$$\begin{aligned}\|b - a\|^2 &= \|a\|^2 + \|b\|^2 - 2a^\top b \\ &= \|a\|^2 + \|b\|^2 - 2C\|a\|\|b\| \\ &= (\|a\| - \|b\|)^2 + 2\|a\|\|b\|(1 - C).\end{aligned}$$

Rearranging and using $(\|a\| - \|b\|)^2 \geq 0$ yields

$$\begin{aligned}1 - C &= \frac{\|b - a\|^2 - (\|a\| - \|b\|)^2}{2\|a\|\|b\|} \\ &\leq \frac{\|b - a\|^2}{2\|a\|\|b\|}.\end{aligned}$$

Substituting $\|b - a\| \leq \varepsilon\|a\|$ and $\|b\| \geq (1 - \varepsilon)\|a\|$ gives

$$\begin{aligned}1 - C &\leq \frac{\varepsilon^2\|a\|^2}{2\|a\|(1 - \varepsilon)\|a\|} \\ &= \frac{\varepsilon^2}{2(1 - \varepsilon)}.\end{aligned}$$

Since $1 - C = \mathcal{L}_{\text{curv}}(t)$ by definition, this proves the curvature-loss bound.

The function $\varepsilon \mapsto \varepsilon^2/(2(1 - \varepsilon))$ is increasing on $(0, 1)$ because

$$\frac{d}{d\varepsilon}\frac{\varepsilon^2}{2(1 - \varepsilon)} = \frac{\varepsilon(2 - \varepsilon)}{2(1 - \varepsilon)^2} > 0.$$

Therefore lowering the Lipschitz constant of v lowers the upper bound whenever the Lipschitz constant remains in $(0, 1)$. □

B.3 Proof of Theorem 1

Theorem 1 (Optimal dynamics-to-reward sample ratio)**.** *If Equation* (3) *holds with exponents* $\alpha, \beta > 0$ *and the bound of Lemma 1 holds, then the minimizing sample counts* $(N^*_{\text{dyn}}, N^*_{\text{rew}})$ *under the constraint* $c_{\text{dyn}}N_{\text{dyn}} + c_{\text{rew}}N_{\text{rew}} = B$ *satisfy*

$$\frac{N^*_{\text{dyn}}}{N^*_{\text{rew}}} := \frac{\alpha}{\beta} \cdot \frac{\gamma L_r(1 + L_\pi)}{1 - \gamma L_f(1 + L_\pi)} \cdot \frac{c_{\text{rew}}}{c_{\text{dyn}}} \cdot \frac{\varepsilon^*_{\text{dyn}}}{\varepsilon^*_{\text{rew}}}. \tag{4}$$

Proof. For sample counts $(N_{\text{dyn}}, N_{\text{rew}})$, Equation (3) gives

$$\varepsilon_{\text{rew}}(N_{\text{rew}}) = A_r N_{\text{rew}}^{-\beta}, \qquad \varepsilon_{\text{dyn}}(N_{\text{dyn}}) = A_d N_{\text{dyn}}^{-\alpha}.$$

Substituting these identities into the upper bound from Equation (1) gives

$$\frac{A_r}{1 - \gamma} N_{\text{rew}}^{-\beta} + \frac{\gamma L_r(1 + L_\pi)}{(1 - \gamma)(1 - \gamma L_f(1 + L_\pi))} A_d N_{\text{dyn}}^{-\alpha}.$$

Define the sample-count-independent coefficients by

$$C_r := \frac{A_r}{1-\gamma}, \qquad C_d := \frac{\gamma L_r(1+L_\pi)A_d}{(1-\gamma)(1-\gamma L_f(1+L_\pi))},$$

so the upper bound after substitution becomes

$$\mathcal{L}_{\text{bound}}(N_{\text{dyn}}, N_{\text{rew}}) := C_r N_{\text{rew}}^{-\beta} + C_d N_{\text{dyn}}^{-\alpha}.$$

For the fixed budget B, the feasible sample counts are exactly the pairs $(N_{\text{dyn}}, N_{\text{rew}})$ satisfying $c_{\text{dyn}}N_{\text{dyn}} + c_{\text{rew}}N_{\text{rew}} = B$. Therefore minimizing the upper bound from Equation (1) over all feasible sample counts is equivalent to minimizing $\mathcal{L}_{\text{bound}}(N_{\text{dyn}}, N_{\text{rew}})$ subject to the same budget constraint.

Because $\alpha, \beta > 0$, the objective decreases as either sample count increases, so the minimizer satisfies $N_{\text{dyn}}^* > 0$ and $N_{\text{rew}}^* > 0$. Define the Lagrangian

$$\Lambda(N_{\text{dyn}}, N_{\text{rew}}, \lambda) := C_d N_{\text{dyn}}^{-\alpha} + C_r N_{\text{rew}}^{-\beta} + \lambda(c_{\text{dyn}}N_{\text{dyn}} + c_{\text{rew}}N_{\text{rew}} - B).$$

Because the minimizing counts satisfy $N_{\text{dyn}}^* > 0$ and $N_{\text{rew}}^* > 0$, setting the partial derivatives of Λ with respect to N_{dyn} and N_{rew} equal to zero gives

$$\alpha\, C_d\, N_{\text{dyn}}^{-\alpha-1} = \lambda\, c_{\text{dyn}},$$
$$\beta\, C_r\, N_{\text{rew}}^{-\beta-1} = \lambda\, c_{\text{rew}}.$$

Multiply the equation for N_{dyn} by N_{dyn} and the equation for N_{rew} by N_{rew}:

$$\alpha\, C_d\, N_{\text{dyn}}^{-\alpha} = \lambda\, c_{\text{dyn}} N_{\text{dyn}},$$
$$\beta\, C_r\, N_{\text{rew}}^{-\beta} = \lambda\, c_{\text{rew}} N_{\text{rew}}.$$

Now divide the equation for N_{dyn} by the equation for N_{rew}:

$$\frac{\alpha\, C_d\, N_{\text{dyn}}^{-\alpha}}{\beta\, C_r\, N_{\text{rew}}^{-\beta}} = \frac{c_{\text{dyn}} N_{\text{dyn}}}{c_{\text{rew}} N_{\text{rew}}}.$$

Rearranging this identity gives

$$\frac{N_{\text{dyn}}}{N_{\text{rew}}} = \frac{\alpha}{\beta} \cdot \frac{c_{\text{rew}}}{c_{\text{dyn}}} \cdot \frac{C_d N_{\text{dyn}}^{-\alpha}}{C_r N_{\text{rew}}^{-\beta}}. \tag{9}$$

Evaluate Equation (9) at the minimizing counts $(N_{\text{dyn}}^*, N_{\text{rew}}^*)$. Using the definition of C_d together with Equation (3), we obtain

$$\begin{aligned} C_d (N_{\text{dyn}}^*)^{-\alpha} &= \frac{\gamma L_r(1+L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}\, A_d (N_{\text{dyn}}^*)^{-\alpha} \\ &= \frac{\gamma L_r(1+L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}\, \varepsilon_{\text{dyn}}^*, \\ C_r (N_{\text{rew}}^*)^{-\beta} &= \frac{A_r}{1-\gamma} (N_{\text{rew}}^*)^{-\beta} \\ &= \frac{1}{1-\gamma}\, \varepsilon_{\text{rew}}^*. \end{aligned}$$

Substituting these two identities yields

$$\begin{aligned} \frac{N_{\text{dyn}}^*}{N_{\text{rew}}^*} &= \frac{\alpha}{\beta} \cdot (1-\gamma) \frac{\gamma L_r(1+L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))} \cdot \frac{c_{\text{rew}}}{c_{\text{dyn}}} \cdot \frac{\varepsilon_{\text{dyn}}^*}{\varepsilon_{\text{rew}}^*} \\ &= \frac{\alpha}{\beta} \cdot \frac{\gamma L_r(1+L_\pi)}{1-\gamma L_f(1+L_\pi)} \cdot \frac{c_{\text{rew}}}{c_{\text{dyn}}} \cdot \frac{\varepsilon_{\text{dyn}}^*}{\varepsilon_{\text{rew}}^*}. \end{aligned}$$

Because $x \mapsto x^{-\alpha}$ and $x \mapsto x^{-\beta}$ are convex on $(0, \infty)$ for $\alpha, \beta > 0$, the objective is convex on the feasible set. Therefore the feasible point satisfying the derivative equations for N_{dyn} and N_{rew} is the unique minimizer. □

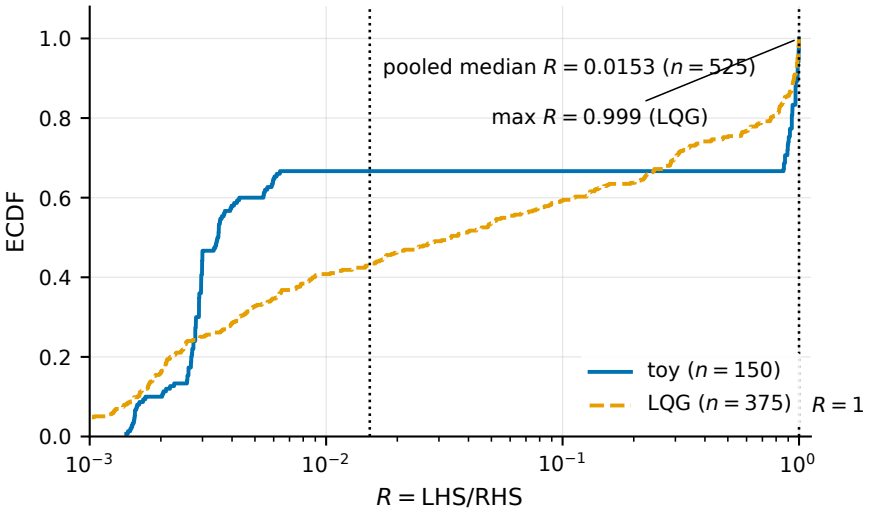


Figure 3: The bound of Equation (1) holds across all $n = 525$ test configurations on both benchmarks. The empirical CDF of $R := \text{LHS}/\text{RHS}$ stays at or below $R = 1$ on every configuration, supporting the use of Lemma 1 as a hypothesis of Theorem 1; full per-benchmark statistics are in this appendix.

B.4 Experiment details: empirical calibration of the decomposed bound

This appendix collects the protocol and per-configuration construction for the calibration experiment in Section 3.1, whose main-text result is summarized in Figure 3.

Each configuration is a tuple $(f, r, \pi, \hat{f}, \hat{r})$, where (f, r) is an MDP instance, π is an L_π-Lipschitz evaluation policy, and $(\hat{f}, \hat{r})$ is the perturbed model pair. The realized per-step errors $\varepsilon_{\text{dyn}}, \varepsilon_{\text{rew}}$ are computed from the $(f, \hat{f})$ and $(r, \hat{r})$ pair using the same definitions that the main-text formulas use.

The synthetic benchmark contains $n = 150$ configurations in which f and r are globally Lipschitz; on this benchmark L_f, L_r, L_π in RHS of Equation (1) are computed analytically as the global Lipschitz constants of the configuration's maps. The LQG benchmark contains $n = 375$ configurations in which the quadratic reward is Lipschitz only on a bounded operating domain (a fixed compact subset of state-action space within which the configuration's rollouts are confined); on this benchmark L_r is the Lipschitz constant of r restricted to that domain.

The empirical CDFs of R in Figure 3 have qualitatively different shapes on the two benchmarks. The LQG ECDF climbs steadily across $R \in [10^{-3}, 1]$. The synthetic ECDF rises sharply through its low-R mass, stays flat at value ≈ 0.66 across $R \in [10^{-2}, 5 \times 10^{-1}]$, and then climbs toward $R = 1$ on the remaining $\approx 25\%$ of configurations. The pooled median $R = 0.015$ is therefore dominated by the LQG arm. The pool maximum is $R = 0.9995$ on a synthetic configuration, and the LQG arm peaks at $R = 0.999$.

Two annotations on Figure 3 should be read as follows: `max R = 0.999 (LQG)` is the LQG-arm maximum (the pool maximum is the slightly larger synthetic value $R = 0.9995$), and the legend label `toy` denotes the synthetic benchmark.

We report only the per-benchmark medians, the pooled median, and the per-benchmark maxima; no bootstrap confidence intervals are used because the relevant claim is the universal $R \leq 1$, which is verified directly on every configuration.

B.5 Experiment details: empirical evaluation of the optimal sample-ratio formula

This appendix collects the per-configuration construction, definitions, statistical methodology, and numerical results for the allocation-evaluation experiment in Section 4.2, whose main-text findings are summarized in Figures 2 and 4.

The configurations used in Figures 2 and 4 are partitioned into four groups, each generated by a Cartesian product over a small set of axes; per-configuration values are recorded in the committed CSVs and we list the grids verbatim below. The interpretation of the axes `sigma_ratio`, `cost_ratio`, λ, and `theta_f0` is deferred to the open item at the end of this appendix.

Linear-value configurations ($n = 30$, panel (a) of Figure 2). The Cartesian product is $(L_f, \lambda) \in \{(0.5, 0.5),\ (2.0, 2.0)\}$ (with $L_f = \lambda$ enforced) crossed with `sigma_ratio` $\in \{0.1, 0.3, 1.0, 3.0, 10.0\}$ and `cost_ratio` $\in \{0.1, 1.0, 10.0\}$, for $2 \times 5 \times 3 = 30$ configurations. The reward Lipschitz constant is fixed at $L_r = 1$.

$\tanh$-value and $\sin$-value configurations ($n = 9$ each, panel (a) of Figure 2). For each of these two value-function families, the Cartesian product is `sigma_ratio` $\in \{0.3, 1.0, 3.0\}$ crossed with `cost_ratio` $\in \{0.1, 1.0, 10.0\}$, with $L_f = L_r = \lambda = 1$ fixed.

Quadratic-value $\sup$-norm-control configurations ($n = 9$, panel (b) of Figure 2). The Cartesian product is the same `sigma_ratio` $\times$ `cost_ratio` grid of size 9, with $\lambda = 1$ and `theta_f0` $= 0.5$ fixed. Each configuration is evaluated twice: once with $L_f^{\text{local}} = 1$ (the realized-sensitivity calculation, plotted as filled circles in panel (b)) and once with $L_f^{\sup} = 2$ (the $\sup$-norm control, plotted as open squares).

LQG configurations ($n = 30$, Figure 4). Configurations are indexed by `seed` $\in \{0, 1, \ldots, 29\}$ and parameterized so that, at $\gamma = 0.8$, the contraction quantity $\gamma L_f(1 + L_\pi)$ lies in $[0.224, 0.228]$ (margin $1 - \gamma L_f(1 + L_\pi) \in [0.772, 0.776]$). The realized constants per seed land in $L_f \in [0.280, 0.285]$, $L_\pi \in [1.6 \times 10^{-4}, 1.0 \times 10^{-3}]$, $L_r \in [0.82, 1.49]$.

For each seed we fit per-configuration power-law scaling laws $\varepsilon_{\text{dyn}}(N) = A_d \cdot N^{-\alpha_d}$ and $\varepsilon_{\text{rew}}(N) = A_r \cdot N^{-\alpha_r}$, with $A_d \in [1.0 \times 10^{-3}, 1.7 \times 10^{-3}]$, $A_r \in [2.2 \times 10^{-3}, 4.0 \times 10^{-3}]$, $\alpha_d \in [0.97, 1.05]$, $\alpha_r \in [0.99, 1.08]$, $R_d^2 \in [0.995, 0.9997]$, and $R_r^2 \in [0.996, 0.9998]$ (recorded per-configuration in the `A_d, A_r, alpha_d, alpha_r, R2_d, R2_r` columns of `lqg_per_instance.csv`). These per-seed exponents are distinct from the global α, β fit in Section 4.1.

The `boundary_excluded` column reports that $0/30$ configurations were filtered out as boundary cases (the filter would exclude any configuration whose realized contraction $\gamma L_f(1 + L_\pi)$ approached 1, which would invalidate the hypothesis of Lemma 1; none did).

We define the quantities used in this experiment. Let $V^\pi(s; f, r)$ denote the discounted return of a fixed evaluation policy π when the dynamics map is f and the reward map is r. For each test configuration, let $\Delta f := \hat{f} - f$ and $\Delta r := \hat{r} - r$ denote the realized model perturbations, and fix a step size $h > 0$. Define the realized value sensitivities by the one-sided finite-difference quotients

$$S_f(s, a) \ := \ \frac{|V^\pi(s; f + h\,\Delta f, r) - V^\pi(s; f, r)|}{h\,\|\Delta f\|},$$
$$S_r(s, a) \ := \ \frac{|V^\pi(s; f, r + h\,\Delta r) - V^\pi(s; f, r)|}{h\,|\Delta r|},$$

where $\|\cdot\|$ is the Euclidean norm; as $h \to 0$, S_f and S_r approach the directional derivatives $\partial_f V^\pi$ and $\partial_r V^\pi$, which provide the underlying intuition for these realized sensitivities. The global Lipschitz constant of the dynamics map in the spectral norm is any

$$L_f \ \geq \ \sup_{(s,a) \neq (s',a')} \frac{\|f(s, a) - f(s', a')\|}{\|s - s'\| + \|a - a'\|},$$

and L_r is defined analogously, on a bounded operating domain when the reward is only locally Lipschitz. Let $K_{\text{Lip}} := \frac{\gamma L_r(1 + L_\pi)}{(1-\gamma)(1-\gamma L_f(1+L_\pi))}$ denote the dynamics coefficient appearing in Equation (1), instantiated with the analytical global Lipschitz constants L_f, L_r, and let K' denote the same coefficient with L_f in the contraction factor $1 - \gamma L_f(1 + L_\pi)$ replaced by the realized value-function sensitivity S_f. The log-ratio residual ℓ is defined in Section 4.2.

We summarize each group of configurations by the across-configuration median of $|\ell|$ (or ℓ when overshoot direction is informative); Figure 4 jitters the vertical position of each LQG marker for

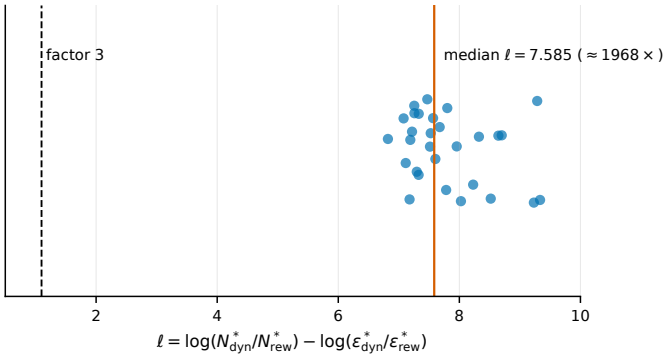


Figure 4: The multiplier in Equation (4) instantiated with the global Lipschitz constants of Lemma 1 overshoots realized ratios by about three orders of magnitude on LQG. Each of $n = 30$ LQG configurations at $\gamma = 0.8$ lies above the dashed horizontal line $\ell = \log 3$ (predicted and realized ratios within a factor of 3); the solid horizontal line marks the median $\ell = 7.585$.

visibility. To check that overshoot in the global-Lipschitz multiplier comparison is not consistent with chance, we run a one-sided sign test against the null that overprediction and underprediction are equally likely; with $30/30$ configurations overshooting, the test gives $p = 0.5^{30} \approx 9.31 \times 10^{-10}$.

The headline per-group medians are reported in Section 4.2; here we record the additional numbers underlying the sup-norm-control comparison. On the 9 quadratic-value configurations, the median residual rises from $|\ell_{\text{realized}}| = 0.074$ to $|\ell_{\sup}| = 0.684$ when realized sensitivities are replaced by their sup-norm overestimates, a factor of $\approx 9.25\times$. The maximum $|\ell|$ on the linear-value configurations is also 0, complementing the median value of 0 already reported in main text.

A separate statistic isolates the dynamics-coefficient slack itself. On the same $n = 30$ LQG configurations (committed per-configuration as the `K_prime` column of `lqg_per_instance.csv`), the multiplier ratio K_{Lip}/K' has median 2.37 and maximum 3.23 (minimum 1.71). This is a different quantity from the residual ℓ: the factor ≈ 1968 figure is $\exp(7.585)$ on the residual itself, while K_{Lip}/K' isolates the contribution of replacing the global L_f with the realized S_f in the dynamics-side coefficient alone.

Worst-case bounds in approximate dynamic programming, such as the simulation-lemma bounds on policy-evaluation and policy-improvement error stated in terms of model error, are classically conservative when instantiated with global Lipschitz or sup-norm constants [Munos, 2003, Kakade and Langford, 2002]. The pattern observed here matches that classical observation: the bound in Lemma 1 certifies a sufficient condition, while the proportionality in Equation (4) carries the predictive content for sample allocation.

B.6 Proof of Theorem 2

Theorem 2 (Finite-horizon REINFORCE under noisy rewards)**.** *Assume* $W_H^2 := \mathbb{E}\big[\max_{0\le t\le H-1} \|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2\big] < \infty$. *Suppose the rewards are observed with additive noise,* $\hat{r}_t = r(s_t, a_t) + \eta_t$*, where the noise variables* η_t *are i.i.d. with* $\mathbb{E}[\eta_t] = 0$ *and* $\mathrm{Var}[\eta_t] = \sigma_\eta^2 < \infty$*, and each* η_t *is independent of the state-action history* $((s_0, a_0), \dots, (s_t, a_t))$*. Then* $\hat{g}$ *satisfies*

$$\mathbb{E}[\hat{g}] = g_H, \qquad \mathrm{Var}[\hat{g}] \le \mathrm{Var}[\hat{g}]_{\eta\equiv 0} + \frac{\sigma_\eta^2 H W_H^2}{K(1-\gamma)^2}. \tag{5}$$

where $\mathrm{Var}[\hat{g}]_{\eta\equiv 0}$ *denotes the variance of the same estimator when the rewards are noise-free.*

Proof. A trajectory $\tau = ((s_0, a_0), \dots, (s_{H-1}, a_{H-1}))$ is sampled by executing π_θ in $\mathcal{M}$. Let G_t, $\hat{G}_t$, and $\hat{g}^{(1)}$ be as defined in Section 5. Define the corresponding REINFORCE estimator when the rewards are noise-free as $g^{(1)} := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, G_t$, so $\mathbb{E}_\tau[g^{(1)}] = g_H$.

Let $N_t := \sum_{t'=t}^{H-1} \gamma^{t'-t} \eta_{t'}$, so that $\hat{G}_t = G_t + N_t$. Let $\delta^{(1)} := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, N_t$, so that $\hat{g}^{(1)} = g^{(1)} + \delta^{(1)}$. Fix the sampled trajectory τ and take conditional expectation over the reward noise. Because each $\eta_{t'}$ is independent of the state-action history and has mean zero,

$$\begin{aligned}
\mathbb{E}[N_t \mid \tau] &= \sum_{t'=t}^{H-1} \gamma^{t'-t}\, \mathbb{E}[\eta_{t'} \mid \tau] \\
&= \sum_{t'=t}^{H-1} \gamma^{t'-t}\, \mathbb{E}[\eta_{t'}] \\
&= 0.
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathbb{E}[\delta^{(1)} \mid \tau] &= \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \mathbb{E}[N_t \mid \tau] \\
&= 0,
\end{aligned} \tag{10}$$

and hence

$$\mathbb{E}[\hat{g}^{(1)} \mid \tau] = g^{(1)}.$$

Therefore $\mathbb{E}[\hat{g}^{(1)}] = \mathbb{E}[g^{(1)}] = g_H$, and averaging over K trajectories gives $\mathbb{E}[\hat{g}] = g_H$.
Fix the sampled trajectory τ and define

$$w_t := \nabla_\theta \log \pi_\theta(a_t \mid s_t).$$

Then

$$\begin{aligned}
\delta^{(1)} &= \sum_{t=0}^{H-1} w_t \sum_{t'=t}^{H-1} \gamma^{t'-t} \eta_{t'} \\
&= \sum_{t=0}^{H-1} \sum_{t'=t}^{H-1} \gamma^{t'-t} w_t \eta_{t'} \\
&= \sum_{t'=0}^{H-1} \sum_{\substack{t=0 \\ t \le t'}}^{H-1} \gamma^{t'-t} w_t \eta_{t'} \\
&= \sum_{t'=0}^{H-1} \sum_{t=0}^{t'} \gamma^{t'-t} w_t \eta_{t'}.
\end{aligned}$$

Define

$$v_{t'} := \sum_{t=0}^{t'} \gamma^{t'-t} w_t.$$

Then

$$\delta^{(1)} = \sum_{t'=0}^{H-1} v_{t'} \eta_{t'}.$$

Using Equation (10),

$$\begin{aligned}
\mathrm{Var}[\delta^{(1)} \mid \tau] &= \sum_{i=1}^{d} \mathrm{Var}[\delta_i^{(1)} \mid \tau] \\
&= \sum_{i=1}^{d} \mathbb{E}\left[\left(\delta_i^{(1)} - \mathbb{E}[\delta_i^{(1)} \mid \tau]\right)^2 \middle| \tau\right] \\
&= \mathbb{E}\left[\|\delta^{(1)} - \mathbb{E}[\delta^{(1)} \mid \tau]\|^2 \mid \tau\right] \\
&= \mathbb{E}\left[\|\delta^{(1)}\|^2 \mid \tau\right] \\
&= \mathbb{E}\left[\left\|\sum_{t'=0}^{H-1} v_{t'}\eta_{t'}\right\|^2 \middle| \tau\right] \\
&= \mathbb{E}\left[\left\langle \sum_{t'=0}^{H-1} v_{t'}\eta_{t'}, \sum_{u=0}^{H-1} v_u\eta_u \right\rangle \middle| \tau\right] \\
&= \mathbb{E}\left[\sum_{t'=0}^{H-1}\sum_{u=0}^{H-1} \langle v_{t'}\eta_{t'}, v_u\eta_u\rangle \middle| \tau\right] \\
&= \mathbb{E}\left[\sum_{t'=0}^{H-1}\sum_{u=0}^{H-1} \eta_{t'}\eta_u\langle v_{t'}, v_u\rangle \middle| \tau\right] \\
&= \sum_{t'=0}^{H-1}\sum_{u=0}^{H-1} \mathbb{E}[\eta_{t'}\eta_u \mid \tau]\,\langle v_{t'}, v_u\rangle \\
&= \sum_{t'=0}^{H-1} \mathbb{E}[\eta_{t'}^2 \mid \tau]\,\langle v_{t'}, v_{t'}\rangle + \sum_{t'=0}^{H-1}\sum_{\substack{u=0\\u\neq t'}}^{H-1} \mathbb{E}[\eta_{t'}\eta_u \mid \tau]\,\langle v_{t'}, v_u\rangle
\end{aligned}$$

For $t' \neq u$, independence of the reward noise from τ, together with independence across time and zero mean, gives $\mathbb{E}[\eta_{t'}\eta_u \mid \tau] = \mathbb{E}[\eta_{t'}\eta_u] = \mathbb{E}[\eta_{t'}]\mathbb{E}[\eta_u] = 0$, while $\mathbb{E}[\eta_{t'}^2 \mid \tau] = \mathbb{E}[\eta_{t'}^2] = \sigma_\eta^2$. Therefore

$$\begin{aligned}
&= \sum_{t'=0}^{H-1} \sigma_\eta^2 \|v_{t'}\|^2 \\
&= \sigma_\eta^2 \sum_{t'=0}^{H-1} \|v_{t'}\|^2.
\end{aligned}$$

Next,

$$\begin{aligned}
\|v_{t'}\| &= \left\|\sum_{t=0}^{t'} \gamma^{t'-t} w_t\right\| \\
&\leq \sum_{t=0}^{t'} \gamma^{t'-t}\|w_t\| \\
&\leq \left(\sum_{t=0}^{t'} \gamma^{t'-t}\right) \max_{0\leq t\leq H-1} \|w_t\| \\
&\leq \frac{1}{1-\gamma} \max_{0\leq t\leq H-1} \|w_t\|.
\end{aligned}$$

Substituting the bound on $\|v_{t'}\|$ into the conditional variance gives

$$\begin{aligned}\mathrm{Var}[\delta^{(1)} \mid \tau] &\le \sigma_\eta^2 \sum_{t'=0}^{H-1} \left(\frac{1}{1-\gamma} \max_{0 \le t \le H-1} \|w_t\| \right)^2 \\ &= \frac{\sigma_\eta^2 H}{(1-\gamma)^2} \max_{0 \le t \le H-1} \|w_t\|^2.\end{aligned}$$

Taking expectations over τ and using the definitions of w_t and W_H^2 yields

$$\begin{aligned}\mathbb{E}[\mathrm{Var}[\delta^{(1)}|\tau]] &\le \frac{\sigma_\eta^2 H}{(1-\gamma)^2} \mathbb{E}\left[\max_{0 \le t \le H-1} \|w_t\|^2 \right] \\ &= \frac{\sigma_\eta^2 H}{(1-\gamma)^2} \mathbb{E}\left[\max_{0 \le t \le H-1} \|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2 \right] \\ &= \frac{\sigma_\eta^2 H W_H^2}{(1-\gamma)^2}.\end{aligned}$$

Now apply the law of total variance to each coordinate and sum over coordinates. Here the conditional variance is over the reward noise given a fixed trajectory τ:

$$\begin{aligned}\mathrm{Var}[\hat{g}^{(1)}] &= \mathrm{Var}\big(\mathbb{E}[\hat{g}^{(1)} \mid \tau]\big) + \mathbb{E}\big[\mathrm{Var}[\hat{g}^{(1)} \mid \tau]\big] \\ &= \mathrm{Var}[g^{(1)}] + \mathbb{E}\big[\mathrm{Var}[g^{(1)} + \delta^{(1)} \mid \tau]\big]\end{aligned}$$

Because $g^{(1)}$ depends only on τ, conditioning on τ makes $g^{(1)}$ deterministic. Hence $\mathrm{Var}[g^{(1)} + \delta^{(1)} \mid \tau] = \mathrm{Var}[\delta^{(1)} \mid \tau]$ almost surely, so

$$\begin{aligned}&= \mathrm{Var}[g^{(1)}] + \mathbb{E}[\mathrm{Var}[\delta^{(1)} \mid \tau]] \\ &\le \mathrm{Var}[g^{(1)}] + \frac{\sigma_\eta^2 H W_H^2}{(1-\gamma)^2}. \qquad (11)\end{aligned}$$

Since the trajectories and their reward noises are sampled independently across k, the estimators $\hat{g}^{(1)}, \ldots, \hat{g}^{(K)}$ are independent. Each estimator has mean g_H. Write $\hat{g}_i$ and $(g_H)_i$ for the ith

coordinates of $\hat{g}$ and g_H. Then

$$\begin{aligned}
\mathrm{Var}[\hat{g}] &= \sum_{i=1}^{d} \mathrm{Var}[\hat{g}_i] \\
&= \sum_{i=1}^{d} \mathbb{E}\big[(\hat{g}_i - \mathbb{E}[\hat{g}_i])^2\big] \\
&= \sum_{i=1}^{d} \mathbb{E}\big[(\hat{g}_i - (g_H)_i)^2\big] \\
&= \mathbb{E}\left[\sum_{i=1}^{d} (\hat{g}_i - (g_H)_i)^2\right] \\
&= \mathbb{E}\big[\|\hat{g} - g_H\|^2\big] \\
&= \mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^{K} \hat{g}^{(k)} - g_H\right\|^2\right] \\
&= \frac{1}{K^2}\mathbb{E}\left[\left\|\sum_{k=1}^{K} (\hat{g}^{(k)} - g_H)\right\|^2\right] \\
&= \frac{1}{K^2}\mathbb{E}\left[\left\langle \sum_{k=1}^{K} (\hat{g}^{(k)} - g_H), \sum_{\ell=1}^{K} (\hat{g}^{(\ell)} - g_H)\right\rangle\right] \\
&= \frac{1}{K^2}\sum_{k=1}^{K}\sum_{\ell=1}^{K} \mathbb{E}\Big[\big\langle \hat{g}^{(k)} - g_H, \hat{g}^{(\ell)} - g_H\big\rangle\Big]
\end{aligned}$$

For $k \neq \ell$, independence gives $\mathbb{E}[\langle \hat{g}^{(k)} - g_H, \hat{g}^{(\ell)} - g_H\rangle] = \langle \mathbb{E}[\hat{g}^{(k)} - g_H], \mathbb{E}[\hat{g}^{(\ell)} - g_H]\rangle = 0$. Therefore only the terms with $k = \ell$ remain, so

$$\begin{aligned}
&= \frac{1}{K^2}\sum_{k=1}^{K} \mathbb{E}\Big[\big\langle \hat{g}^{(k)} - g_H, \hat{g}^{(k)} - g_H\big\rangle\Big] \\
&= \frac{1}{K^2}\sum_{k=1}^{K} \mathbb{E}\left[\sum_{i=1}^{d} (\hat{g}_i^{(k)} - (g_H)_i)^2\right] \\
&= \frac{1}{K^2}\sum_{k=1}^{K}\sum_{i=1}^{d} \mathbb{E}\Big[(\hat{g}_i^{(k)} - (g_H)_i)^2\Big]
\end{aligned}$$

Since each coordinate of $\hat{g}^{(k)}$ has mean the corresponding coordinate of g_H,

$$= \frac{1}{K^2}\sum_{k=1}^{K}\sum_{i=1}^{d} \mathrm{Var}[\hat{g}_i^{(k)}]$$

By the definition of Var for vector-valued estimators,

$$= \frac{1}{K^2}\sum_{k=1}^{K} \mathrm{Var}[\hat{g}^{(k)}]$$

The estimators $\hat{g}^{(1)}, \ldots, \hat{g}^{(K)}$ have the same distribution, so each term in the sum equals $\mathrm{Var}[\hat{g}^{(1)}]$:

$$\begin{aligned}
&= \frac{1}{K^2}\sum_{k=1}^{K} \mathrm{Var}[\hat{g}^{(1)}] \\
&= \frac{1}{K}\mathrm{Var}[\hat{g}^{(1)}]. \qquad (12)
\end{aligned}$$

When the rewards are noise-free, the averaged estimator $\hat{g}$ equals $\hat{g} = \frac{1}{K}\sum_{k=1}^{K} g^{(k)}$. Repeating the expansion above gives

$$\begin{aligned}\mathrm{Var}[\hat{g}]_{\eta\equiv 0} &= \mathrm{Var}\left[\frac{1}{K}\sum_{k=1}^{K} g^{(k)}\right] \\ &= \frac{1}{K^2}\sum_{k=1}^{K}\sum_{\ell=1}^{K}\mathbb{E}\Big[\Big\langle g^{(k)} - g_H, g^{(\ell)} - g_H\Big\rangle\Big]\end{aligned}$$

For $k \neq \ell$, independence gives $\mathbb{E}[\langle g^{(k)} - g_H, g^{(\ell)} - g_H\rangle] = \langle \mathbb{E}[g^{(k)} - g_H], \mathbb{E}[g^{(\ell)} - g_H]\rangle = 0$, so

$$\begin{aligned}&= \frac{1}{K^2}\sum_{k=1}^{K}\mathbb{E}\Big[\|g^{(k)} - g_H\|^2\Big] \\ &= \frac{1}{K^2}\sum_{k=1}^{K}\mathrm{Var}[g^{(k)}] \\ &= \frac{1}{K^2}\sum_{k=1}^{K}\mathrm{Var}[g^{(1)}] \\ &= \frac{1}{K}\mathrm{Var}[g^{(1)}]. \qquad (13)\end{aligned}$$

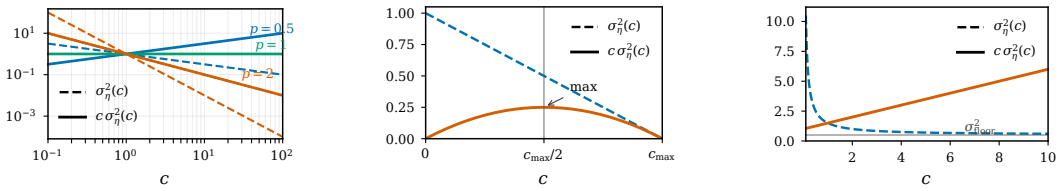
Combining Equation (12) with Equation (11) gives

$$\begin{aligned}\mathrm{Var}[\hat{g}] &= \frac{1}{K}\mathrm{Var}[\hat{g}^{(1)}] \\ &\leq \frac{1}{K}\left(\mathrm{Var}[g^{(1)}] + \frac{\sigma_\eta^2 H W_H^2}{(1-\gamma)^2}\right) \\ &= \frac{1}{K}\mathrm{Var}[g^{(1)}] + \frac{\sigma_\eta^2 H W_H^2}{K(1-\gamma)^2} \\ &= \mathrm{Var}[\hat{g}]_{\eta\equiv 0} + \frac{\sigma_\eta^2 H W_H^2}{K(1-\gamma)^2},\end{aligned}$$

where the last equality uses Equation (13). □

B.7 Visualization of the three fidelity-cost regimes from Corollary 2

Figure 5 plots $\sigma_\eta^2(c)$ and $\Phi(c) = c\,\sigma_\eta^2(c)$ for the three regimes analyzed in Section 5.



(a) Power-law fidelity. (b) Bounded fidelity. (c) Irreducible noise floor.

Figure 5: Three examples of how the noise variance $\sigma_\eta^2(c)$ depends on the cost c, and the corresponding $\Phi(c) = c\,\sigma_\eta^2(c)$ from Corollary 2.

B.8 Proof of Proposition 2

Proposition 2 (Finite-horizon REINFORCE under biased rewards)**.** *Let $b : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be a reward bias function, define $\tilde{r}(s,a) := r(s,a) + b(s,a)$, and define $\tilde{\mathcal{M}} := (\mathcal{S}, \mathcal{A}, f, \tilde{r}, \gamma)$. For trajectories*

sampled by executing π_θ under the true dynamics f, let $\tilde{g}^{(1)}, \ldots, \tilde{g}^{(K)}$ be independent REINFORCE *estimators computed using the biased rewards $\tilde{r}(s_t, a_t)$, and define $\tilde{g} := \frac{1}{K}\sum_{k=1}^{K} \tilde{g}^{(k)}$. Define $B_H(\theta) := \mathbb{E}\Big[\sum_{t=0}^{H-1} \gamma^t b(s_t, a_t)\Big]$, where the expectation is over trajectories generated by executing π_θ under the true dynamics f. Then*

$$\mathbb{E}[\tilde{g}] = \nabla_\theta J_H(\pi_\theta, \tilde{\mathcal{M}}) = g_H + \nabla_\theta B_H(\theta). \tag{6}$$

If $\mathrm{Var}[\tilde{g}^{(1)}] < \infty$, then

$$\mathbb{E}\big[\|\tilde{g} - g_H\|^2\big] = \frac{1}{K}\mathrm{Var}[\tilde{g}^{(1)}] + \|\nabla_\theta B_H(\theta)\|^2. \tag{7}$$

Consequently, when $\nabla_\theta B_H(\theta) \neq 0$, averaging over more trajectories reduces the variance term but does not remove the bias as an estimator of g_H.

Proof. For one trajectory $\tau = ((s_0, a_0), \ldots, (s_{H-1}, a_{H-1}))$ sampled by executing π_θ under the true dynamics f, define the biased finite-horizon return from time t onward by

$$\tilde{G}_t := \sum_{t'=t}^{H-1} \gamma^{t'-t} \tilde{r}(s_{t'}, a_{t'}).$$

The single-trajectory estimator using biased rewards is

$$\tilde{g}^{(1)} := \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \tilde{G}_t.$$

The estimator $\tilde{g}^{(1)}$ is the finite-horizon REINFORCE estimator for the MDP $\tilde{\mathcal{M}}$, which has the true dynamics f and the biased reward $\tilde{r}$. Therefore the finite-horizon policy-gradient identity gives

$$\mathbb{E}[\tilde{g}^{(1)}] = \nabla_\theta J_H(\pi_\theta, \tilde{\mathcal{M}}).$$

Averaging over K independent trajectories does not change the mean:

$$\begin{aligned}
\mathbb{E}[\tilde{g}] &= \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \tilde{g}^{(k)}\right] \\
&= \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[\tilde{g}^{(k)}] \\
&= \nabla_\theta J_H(\pi_\theta, \tilde{\mathcal{M}}).
\end{aligned}$$

By the definitions of $\tilde{r}$, J_H, and B_H,

$$\begin{aligned}
J_H(\pi_\theta, \tilde{\mathcal{M}}) &= \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t \tilde{r}(s_t, a_t)\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t (r(s_t, a_t) + b(s_t, a_t))\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)\right] + \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t b(s_t, a_t)\right] \\
&= J_H(\pi_\theta, \mathcal{M}) + B_H(\theta).
\end{aligned}$$

Taking gradients with respect to θ gives

$$\begin{aligned}
\nabla_\theta J_H(\pi_\theta, \tilde{\mathcal{M}}) &= \nabla_\theta J_H(\pi_\theta, \mathcal{M}) + \nabla_\theta B_H(\theta) \\
&= g_H + \nabla_\theta B_H(\theta).
\end{aligned}$$

Combining the preceding displays proves Equation (6).

Let

$$\mu_b := \mathbb{E}[\tilde{g}] - g_H.$$

By Equation (6), $\mu_b = \nabla_\theta B_H(\theta)$. We use the identity $\|x+y\|^2 = \|x\|^2 + 2\langle x, y\rangle + \|y\|^2$ with $x = \tilde{g} - \mathbb{E}[\tilde{g}]$ and $y = \mu_b$. We also use linearity of expectation and the fact that μ_b is a fixed vector once θ is fixed, so $\mathbb{E}[\|\mu_b\|^2] = \|\mu_b\|^2$. Then

$$\begin{aligned}
\mathbb{E}\big[\|\tilde{g} - g_H\|^2\big] &= \mathbb{E}\big[\|\tilde{g} - \mathbb{E}[\tilde{g}] + \mathbb{E}[\tilde{g}] - g_H\|^2\big] \\
&= \mathbb{E}\big[\|\tilde{g} - \mathbb{E}[\tilde{g}] + \mu_b\|^2\big] \\
&= \mathbb{E}\big[\|\tilde{g} - \mathbb{E}[\tilde{g}]\|^2 + 2\,\langle \tilde{g} - \mathbb{E}[\tilde{g}], \mu_b\rangle + \|\mu_b\|^2\big] \\
&= \mathbb{E}\big[\|\tilde{g} - \mathbb{E}[\tilde{g}]\|^2\big] + 2\,\mathbb{E}[\langle \tilde{g} - \mathbb{E}[\tilde{g}], \mu_b\rangle] + \|\mu_b\|^2 \\
&= \mathrm{Var}[\tilde{g}] + \|\mu_b\|^2
\end{aligned}$$

because $\mathbb{E}[\langle \tilde{g} - \mathbb{E}[\tilde{g}], \mu_b\rangle] = \langle \mathbb{E}[\tilde{g}] - \mathbb{E}[\tilde{g}], \mu_b\rangle = 0$. Since the estimators $\tilde{g}^{(1)}, \ldots, \tilde{g}^{(K)}$ are independent and identically distributed,

$$\begin{aligned}
\mathrm{Var}[\tilde{g}] &= \mathrm{Var}\left[\frac{1}{K}\sum_{k=1}^{K} \tilde{g}^{(k)}\right] \\
&= \frac{1}{K^2}\sum_{k=1}^{K} \mathrm{Var}[\tilde{g}^{(k)}] \\
&= \frac{1}{K}\mathrm{Var}[\tilde{g}^{(1)}].
\end{aligned}$$

Substituting $\mu_b = \nabla_\theta B_H(\theta)$ gives Equation (7). If $\nabla_\theta B_H(\theta) \neq 0$, the second term in Equation (7) is positive and does not depend on K. □

C Experimental details

C.1 Synthetic teacher environment

The environment is a synthetic continuous-control environment whose dynamics and reward function are defined by frozen, randomly-initialized neural networks. Decoupling the environment from any particular physics simulator gives exact control over dynamics complexity, reward fidelity, and partial observability, which enables the controlled scaling experiment of Section 4.1.

The teacher consists of two 2-layer ReLU MLPs. The dynamics teacher $f_{\mathrm{dyn}} : \mathbb{R}^{d_s+d_a} \to \mathbb{R}^{d_s}$ maps (s_t, a_t) to the next state via $s_{t+1} = \tanh(f_{\mathrm{dyn}}([s_t, a_t]))$. The reward teacher $f_{\mathrm{rew}} : \mathbb{R}^{d_s+d_a} \to \mathbb{R}$ produces the scalar reward $r_t = f_{\mathrm{rew}}([s_t, a_t])/\sqrt{d_h}$, where d_h is the teacher hidden width. All teacher weights are drawn from $\mathcal{N}(0, 1)$ using a fixed seed and never updated, so the same seed yields the identical MDP. The $\tanh$ activation in dynamics constrains the state to $[-1, 1]^{d_s}$, and the $1/\sqrt{d_h}$ reward scaling ensures comparable reward magnitude across teacher widths. Default dimensions: $d_s = 12$, $d_a = 4$, $d_h = 64$, episode length $T = 500$.

C.2 Experiment details: power-law scaling of dynamics and reward error

This appendix collects the protocol and engineering details for the error–sample-size scaling experiment in Section 4.1, whose main-text result is summarized in Figure 1.

We use the teacher of Section C.1, which keeps the experiment inside the regime where the assumptions of Theorem 1 hold. Training transitions are drawn i.i.d. from the teacher's induced state distribution by rolling out a smooth reference policy under the frozen teacher and subsampling (s, a, s', r) tuples; the released code records the exact rollout and subsampling procedure.

Power-law exponents are fit on the seven anchors $N \in \{2{,}000,\ 5{,}000,\ 10{,}000,\ 20{,}000,\ 50{,}000,\ 100{,}000,\ 200{,}000\}$, each with 100 independent seeds. A single $N = 500{,}000$ run is preserved in the released dataset for future analyses but is excluded from both the figure and the fit, because only one seed was completed at that anchor.

For each (N, seed) we train two independent students that mirror the teacher architecture: a dynamics head $f^{\theta}_{\text{dyn}} : \mathbb{R}^{d_s+d_a} \to \mathbb{R}^{d_s}$ and a reward head $f^{\phi}_{\text{rew}} : \mathbb{R}^{d_s+d_a} \to \mathbb{R}$, each implemented as a 2-layer ReLU MLP with hidden width $d_h = 64$ matching the teacher (no output activation, no shared trunk, no weight tying, no input/target normalization, default PyTorch `nn.Linear` initialization). Both heads are optimized independently with Adam at learning rate 10^{-3} (PyTorch defaults for $\beta_1, \beta_2, \epsilon$; no weight decay) and a mini-batch size of 256 drawn from the same N-sample training pool with reshuffling each epoch. Training proceeds for a fixed schedule of 200 epochs with no learning-rate schedule and no early stopping; the held-out validation MSE is recorded every epoch but is not used to gate the optimizer, and the reported $\varepsilon_{\text{dyn}}(N_{\text{dyn}})$ and $\varepsilon_{\text{rew}}(N_{\text{rew}})$ are the final-epoch validation MSEs of the two heads, computed under `torch.no_grad`. The training objective is the per-head mean-squared error, $\mathcal{L}_{\text{dyn}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(s,a,s') \in \mathcal{B}} \| f^{\theta}_{\text{dyn}}([s,a]) - s' \|_2^2$ and $\mathcal{L}_{\text{rew}}(\phi) = \frac{1}{|\mathcal{B}|} \sum_{(s,a,r) \in \mathcal{B}} (f^{\phi}_{\text{rew}}([s,a]) - r)^2$, evaluated on each mini-batch $\mathcal{B}$. At 200 training epochs, and for every anchor $N \geq 2{,}000$, the training MSE is fully saturated, so the final-epoch values used by the fit coincide with the converged held-out loss to within seed noise.

Each anchor evaluates against a fixed held-out set of $5{,}000$ transitions, drawn independently of the training pool. Holding the validation set size constant across N keeps the evaluation noise floor independent of the training pool size, so any N-dependence of the measured MSE reflects student generalization rather than a changing estimator variance.

Per anchor, we report the across-seed mean MSE in the figure with ± 1 standard error of the mean (SEM) error bars; this across-seed SEM is distinct from the bootstrap standard errors reported below for the fitted parameters. The power laws $\varepsilon_{\text{dyn}}(N_{\text{dyn}}) = A_d \cdot N_{\text{dyn}}^{-\alpha}$ and $\varepsilon_{\text{rew}}(N_{\text{rew}}) = A_r \cdot N_{\text{rew}}^{-\beta}$ are fit by ordinary log–log linear regression on the per-anchor means, where α is the dynamics exponent and β is the reward exponent reported in Section 4.1. Uncertainty on $(A_d, A_r, \alpha, \beta)$ is obtained by a stratified bootstrap with $1{,}000$ resamples that draws seeds with replacement *within* each anchor before refitting; we report the bootstrap standard error and the 2.5/97.5 percentile interval as the 95% confidence interval. The resulting bootstrap standard errors are $A_d \pm 0.04$, $\alpha \pm 0.01$, $A_r \pm 13.3$, and $\beta \pm 0.02$, and the 95% bootstrap confidence intervals are $A_d \in [0.28,\ 0.42]$, $\alpha \in [0.09,\ 0.13]$, $A_r \in [68.1,\ 121]$, and $\beta \in [0.93,\ 0.99]$.