SAMPLE EFFICIENT WORLD MODELS USING GRADIENT PENALIZED LATENT DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models are trained on one-step transitions and then rolled out for dozens of steps to train a policy. Where those transitions are sparse, the learned transition function can change rapidly between neighboring states, and long imagined rollouts compound the resulting errors. In continuous control, neighboring states evolve under similar transition laws, so a transition observed at one state is informative about its neighbors. To exploit all such information, we propose GPLD (Gradient Penalized Latent Dynamics), a method that makes possible the sharing of this information between neighboring states. We derive it as the small-neighborhood limit of a finite-difference penalty on transition laws in a finite MDP, which yields the squared Frobenius norm of the dynamics Jacobian. We apply it to DreamerV3's posterior latent distribution, the map through which observations shape the learned dynamics, and estimate it with Hutchinson-style stochastic probes. With a single configuration held fixed across every task, GPLD raises final return over DreamerV3 on 15 of 18 proprioceptive DeepMind Control tasks (sign test $p = 0.0075$) and mean final return by 9.8% on the 12 tasks where DreamerV3 has not already saturated (18.1% in normalized return), and by 20.7% on the six `humanoid` and `dog` tasks. On `quadruped` it reaches high return earlier, and its multi-step prediction error is lower than DreamerV3's on most measured quantities and significantly higher on none. Sharing information between neighboring states, as proposed here, improves the sample efficiency of world models.

1 INTRODUCTION

Reinforcement learning agents operating under a fixed interaction budget face two distinct questions: how the budget is *allocated* across states and actions, and how much is *extracted* from each transition once collected. World models address the second. They learn a model of the environment's dynamics and use it for planning and policy optimization (Ha & Schmidhuber, 2018; Hansen et al., 2022), and DreamerV3 has shown that latent world models scale to diverse continuous-control tasks (Hafner et al., 2025).

Yet world models do not exploit a basic structural property of continuous control: nearby states have similar transition behavior. This property matters because data is scarce per state. In a finite MDP, the error of an estimated transition row falls only as the inverse square root of that row's visit count (Weissman et al., 2003; Azar et al., 2013), so halving it requires four times the interaction, unless the row can access information from neighboring states with similar dynamics (Asadi et al., 2018; Ferns et al., 2011). We make this sharing of information across neighbors explicit. A finite-difference penalty between the transition laws of neighboring states, a Dirichlet energy on the state-neighborhood graph (Smola & Kondor, 2003; Belkin et al., 2006), induces a shrinkage estimator that shares information across neighbors most strongly where data is scarcest, and in the small-neighborhood limit the same penalty becomes the squared Frobenius norm of the dynamics Jacobian (Rifai et al., 2011). Applied to the posterior of DreamerV3's latent world model (Hafner et al., 2025), this gives *gradient penalized latent dynamics* (GPLD), defined in (7).

With one configuration fixed across every task, GPLD raises final return on 15 of the 18 proprioceptive DMC tasks. Where DreamerV3 still has room to improve, GPLD raises mean final return by 9.8% (18.1% in normalized return) and area under the learning curve by 20.0%, without hurting the tasks where DreamerV3 is already at the ceiling. The gains persist on the largest bodies in the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

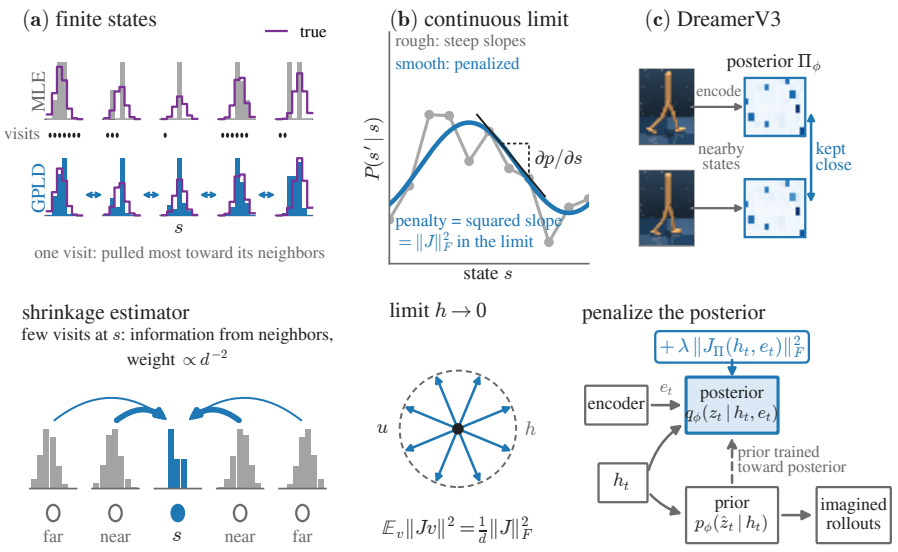


Figure 1: GPLD in one picture. Each column pairs what the penalty does (top) with how it arises (bottom). **(a)** Finite states (illustrative example). Top: maximum-likelihood transition estimates (gray) from few visits (dots) miss the true distributions (outlines); the penalized estimate (blue) shares information across neighboring states and moves rarely visited states the most. Bottom: the penalty induces a shrinkage estimator whose neighbor weights fall as d^{-2}. **(b)** Continuous limit. Top: penalizing differences between neighbors favors a smooth transition map over a rough one, and the penalty is the squared slope. Bottom: as the neighborhood shrinks, averaging over directions gives the Jacobian energy. **(c)** DreamerV3. Top: nearby states are encoded into posterior probability tables that the penalty keeps close (illustrative tables). Bottom: the penalty acts on the posterior; the prior, which drives imagined rollouts, is smoothed by being trained toward the posterior.

suite (20.7% on `humanoid` and `dog`), and on `quadruped` GPLD reaches high return earlier. Under pixel observations the gain largely disappears, and a warm-start diagnostic points to the learned representation as the limiting factor.

Contributions. Our contributions are threefold.

1. **A dynamics regularizer derived from transition estimation** (Sections 2 and 4). Whereas prior work assumes smooth dynamics to derive guarantees for reinforcement learning (Munos & Szepesvári, 2008), enforces a global Lipschitz bound on the learned model (Asadi et al., 2018), or introduces graph and Jacobian penalties as general-purpose smoothness regularizers (Smola & Kondor, 2003; Belkin et al., 2006; Rifai et al., 2011; Hoffman et al., 2019), we show that a finite-difference penalty on neighboring transition distributions induces a data-adaptive shrinkage estimator, whose neighbor weights and withdrawal as data accumulates follow from the penalty in the finite-state objective. Its continuous-state, small-neighborhood limit is the squared Frobenius norm of the dynamics Jacobian, which GPLD applies to DreamerV3's posterior (Section 4).
2. **A measured sample-efficiency mechanism** (Section 3). Complementing visit-count bounds for transition estimation (Weissman et al., 2003; Azar et al., 2013) and analyses of generalization across nearby states (Boyan & Moore, 1994; Ferns et al., 2011), we quantify the value of sharing information across neighbors as an effective-sample-size gain. In a controlled online tabular MDP this gain is largest when transition data is scarce and declines as observations accumulate, as the shrinkage analysis predicts.
3. **A tractable realization in DreamerV3** (Sections 4 and 5). Unlike prior reinforcement-learning smoothness methods, which regularize policies, actions or value functions (Peng et al., 2020; Fu et al., 2021; Chen et al., 2025; Wang et al., 2025), and unlike global or adversarial constraints such as spectral normalization and virtual adversarial training (Miyato et al., 2018; 2019), GPLD uses stochastic Jacobian probes to penalize the local proba-

bility sensitivity of DreamerV3's observation-conditioned latent posterior. With a single configuration across tasks, GPLD improves sample efficiency in proprioceptive continuous control, and its multi-step prediction error is lower than DreamerV3's on most measured quantities and significantly higher on none, and it outperforms the four comparator regularizers (Section 5.4), while the pixel-observation results identify the geometry of the learned representation as an important boundary of the method.

Appendix A discusses related work in more detail.

2 Sharing Information Across Neighboring States

The estimation bound and its single lever. Proofs of all statements in this section are in Appendix C.1. Fix an action a in a finite MDP and abbreviate the true row $p_s := P(\cdot \mid s, a)$ and $N_s := N(s, a)$. The maximum-likelihood row estimate $\widehat{p}_s(s') = N(s, a, s')/N_s$ satisfies

$$\mathbb{E}\,\|\widehat{p}_s - p_s\|_1 \;\le\; \sqrt{\frac{1 - \sum_{s'} p_s(s')^2}{N_s}}\,\sqrt{|\mathcal{S}|} \;\le\; \sqrt{\frac{|\mathcal{S}| - 1}{N_s}} \tag{1}$$

Every row is estimated in isolation: the estimate at (s, a) never uses data at (k, a), however similar s and k may be, so the visit count N_s is the only lever.

Local smoothness and the shrinkage estimator. Let $\psi : \mathcal{S} \to \mathbb{R}^d$ embed states in a metric space, write $d_{sk} := \|\psi(k) - \psi(s)\|_2$, and define $\mathcal{N}_\epsilon(s) := \{k : 0 < d_{sk} \le \epsilon\}$. Since $\mathcal{S}$ is finite, $\psi(\mathcal{S})$ is a finite set of points in $\mathbb{R}^d$, on which derivatives are not defined, so the assumption is placed on a differentiable extension of the transition law.

Assumption 1 (Local smoothness in the embedding)**.** There exist an open set $U \subseteq \mathbb{R}^d$ containing the closed ϵ-ball around every $\psi(s)$, a twice continuously differentiable $\widetilde{P} : U \to \Delta^{|\mathcal{S}|-1}$ with $\widetilde{P}(\psi(s)) = p_s$ for all s, and a constant $M < \infty$ with $\left\|D^2\widetilde{P}(u)\right\| \le M$ on U. Write $G_s := D\widetilde{P}(\psi(s))$.

In particular, ψ must separate states with differing dynamics. Let $w_{sk} \ge 0$ with $\sum_k w_{sk} = 1$ weight the neighbors in $\mathcal{N}_\epsilon(s)$. Pooling and blending with the isolated estimate gives

$$\bar{p}_s = \sum_k w_{sk}\widehat{p}_k, \qquad p_s^{\mathrm{FD}} = (1 - \theta_s)\widehat{p}_s + \theta_s\bar{p}_s, \qquad \theta_s \in [0, 1]. \tag{2}$$

Under Assumption 1 the bias $B := \|\mathbb{E}\bar{p}_s - p_s\|_2$ satisfies $B \le \|G_s\|\,\|\delta_s\|_2 + \frac{M}{2}\bar{\epsilon}^{\,2}$, where $\delta_s := \sum_k w_{sk}(\psi(k) - \psi(s))$ is the weighted mean displacement and $\bar{\epsilon}^{\,2} := \sum_k w_{sk} d_{sk}^2 \le \epsilon^2$. When the neighbors are placed symmetrically about $\psi(s)$, $\delta_s = 0$ and $B = \mathcal{O}(\bar{\epsilon}^{\,2})$; otherwise the first term is $\mathcal{O}(\bar{\epsilon})$. With isolated variance $V_s = (1 - \sum_{s'} p_s(s')^2)/N_s$ and pooled variance $V_{\mathrm{nbr}} = \sum_k w_{sk}^2 V_k$, minimizing mean squared error over θ_s gives

$$\theta_s^\star = \frac{V_s}{V_s + V_{\mathrm{nbr}} + B^2}, \qquad \mathrm{MSE}(\theta_s^\star) = (1 - \theta_s^\star)\,V_s. \tag{3}$$

Two properties matter. The error is never larger than V_s, so at the right mixing weight sharing cannot hurt; and since $V_s \propto 1/N_s$, $\theta_s^\star$ decreases in N_s, and once bias dominates it falls as $1/N_s$; this is the reason GPLD's coefficient decays (Section 4.3). A data-poor row moves far toward its neighbors, while a data-rich row barely moves. The gain is *self-limiting*, and Section 3 measures that decay.

A penalty that induces the estimator. The shrinkage estimator (2) pools one row at a time, using its neighbors' raw estimates. Its mixing weight $\theta_s^\star$ also depends on the unknown bias and variances. Pooling every row at once is harder. Each neighbor's estimate is then pooled as well, so it already contains row s's own data, and (3) no longer gives the optimal weights. A penalty treats all rows jointly. It penalizes the objective for producing transition laws that differ sharply between nearby states:

$$\mathcal{L}_{\mathrm{FD}}(P) = \sum_a \sum_s \sum_{k \in \mathcal{N}_\epsilon(s)} \sum_{s'} \left(\frac{P(s' \mid k, a) - P(s' \mid s, a)}{d_{sk}}\right)^2, \tag{4}$$

added to the negative log-likelihood with regularization weight λ. Dividing each difference by d_{sk} makes the term a measure of local *sensitivity*; it also sets how strongly each neighbor counts. The penalized objective is convex in P.

Restrict the objective to one row with its neighbors held fixed, and approximate the negative log-likelihood by an *isotropic* quadratic about its minimizer, $\frac{a}{2}\|P_s - \widehat{p}_s\|_2^2$, with scalar curvature $a \propto N_s$. Since row s enters (4) both as a center and as a neighbor of each k, setting the gradient to zero gives, with $S_s := \sum_k d_{sk}^{-2}$,

$$P_s^\star = (1-\theta_s)\widehat{p}_s + \theta_s \sum_k \frac{d_{sk}^{-2}}{S_s} P_k, \qquad \theta_s = \frac{4\lambda S_s}{a + 4\lambda S_s}. \tag{5}$$

This is exactly the shrinkage estimator (2), and three things assumed there are consequences here. The weights are $w_{sk} \propto d_{sk}^{-2}$, so nearer neighbors count for more, consistent with Assumption 1, under which a neighbor's bias grows with its distance. The mixing weight increases in λ, which therefore plays the role of θ_s. And since $a \propto N_s$, the mixing weight decreases in the visit count, as $\theta_s^\star$ does in (3); here the decrease comes from the curvature of the likelihood, not from choosing θ_s to minimize error. One qualification applies. Without isotropy the multinomial curvature is diagonal, with entry $N_s/\widehat{p}_s(s')$ for each observed successor, so the scalar θ_s becomes a vector of mixing weights, one per successor, each of the same form up to a shift that restores normalization; this does not affect the three conclusions, which depend only on signs and monotonicities.

Solving the penalized problem exactly. The induced estimator (5) updates one row with its neighbors held fixed. The full penalized problem is convex, and ADMM solves it exactly (Appendix F). It alternates two steps. The first fits each row to its own counts by maximum likelihood, so a well-visited row stays near its own estimate. The second smooths all rows jointly, in the form of (5). It ignores the visit counts, which enter only through the first step. Section 3 uses this exact solution.

3 Verification in Finite State Spaces

Estimation error can be measured only against the true transition law, which no learned or real-world model provides. We therefore test the mechanism in a controlled MDP where the transition law is known by construction and its local smoothness is set explicitly: `var-gaus`, a family of MDPs on 256 states arranged on a circle, whose successor distribution varies smoothly around it (Appendix G). An online ϵ-greedy agent that plans from the exact penalized estimate, computed by ADMM, is compared with one that plans from the maximum-likelihood estimate. The penalty uses either a *per-state* coefficient $\lambda(s,a) = 1/\sqrt{N(s,a)}$, which depends on that pair's own visit count, or a *global* coefficient $\lambda = \sqrt{|\mathcal{S}||\mathcal{A}|/T}$, shared by all states and decaying with the total number of transitions T; the two coincide under uniform visitation.

Accuracy and control. Early in training both penalized estimates are more accurate than maximum likelihood, which needs about eight times as many steps to match the per-state estimate's accuracy at 256 steps, and the per-state agent reaches the optimal reward sooner (Figure 4 and regret in Appendix G).

What a neighbor is worth. The benefit can be read as an effective sample size,

$$N_s^{\text{eff}} \;=\; \frac{\mathrm{Var}(\widehat{p}_s)}{\mathrm{Var}(p_s^{\text{FD}})}\; N_s, \tag{6}$$

the number of observations an isolated estimator would need to match the penalized one. Under equal visitation it is at most $(1 + m_{\text{eff}})N_s$, the row's own visits plus those its neighbors contribute, with m_{eff} the effective number of neighbors (Appendix C.1). In the experiment N_s^{eff}/N_s is largest while data is scarce and falls as data accumulates, as (3) predicts: sharing information across neighbors helps most when data is scarce and withdraws as data accumulates (Figure 5 in Appendix G).

4 TRANSFERRING THE IDEA TO LATENT WORLD MODELS

4.1 FROM A FINITE SUM TO A JACOBIAN

Let the transition map be a differentiable $f : \mathbb{R}^d \to \Delta^{C-1}$ and replace the discrete neighborhood by an isotropic ball of radius h about u. Writing $k = u + hv$ for a unit direction v, the inner term of (4) becomes $\|(f(u+hv) - f(u))/h\|_2^2 \to \|J_f(u)v\|_2^2$ as $h \to 0$, and averaging over directions uniformly on the sphere gives $\mathbb{E}_v \|J_f(u)v\|_2^2 = \frac{1}{d}\|J_f(u)\|_F^2$ (proof in Appendix C.2). The discrete penalty therefore passes, in the small-neighborhood limit under isotropy, to squared Frobenius Jacobian energy. The factor $1/d$ is absorbed into the coefficient, which is why a regularization weight tuned at one input dimension does not transfer unchanged to another.

4.2 WHICH MAP SHOULD BE REGULARIZED?

The derivation concerns a transition law, and a latent world model contains several maps that could play that role. DreamerV3 (Hafner et al., 2025) maintains a recurrent state $h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1})$, an embedding $e_t = \mathrm{enc}_\phi(o_t)$, a prior $p_\phi(\hat{z}_t \mid h_t)$, and a posterior $q_\phi(z_t \mid h_t, e_t)$ whose probability table $\Pi_\phi(u_t)$, with $u_t := [h_t, e_t]$, has K rows, one per categorical latent variable, of C classes each (Appendix B). We penalize the posterior for three reasons; Table 3 in Appendix B compares every candidate. It is the only distribution over latent states that depends on the observation, so it is the map through which environment data shapes the representation. Every gradient that reaches the encoder passes through Π_ϕ, so the penalty acts directly on that map. The prior needs no penalty of its own: DreamerV3 trains it toward the posterior in the dynamics loss, so a smoother posterior gives the prior smoother targets to predict, and the prior, though not penalized, is less sensitive to its input as well: its input-scaled Jacobian energy is lower than DreamerV3's on all three tasks measured, significantly on two, and its entropy is higher (Table 23). Applied to the prior instead, with the same coefficient, sampling fraction and seeds (three per arm), the penalty's normalized return across eight tasks falls from 1.106 to 1.008 in final return and from 1.100 to 0.997 in the area under the learning curve (nAUC, Section 5); the posterior is ahead on seven of the eight tasks (Appendix K). The derivation fixes the form of the penalty, and the choice of map is made on the grounds above. Imagination rolls the prior rather than the posterior, so whether a smoother posterior yields more accurate imagined dynamics is an empirical question; the comparison above and the open-loop measurements of Section 5.3 answer it.

4.3 GPLD

Writing $\pi_\phi^{(i)}(u_t) \in \Delta^{C-1}$ for the i-th row of the posterior table, GPLD penalizes

$$\mathcal{R}_{\mathrm{GPLD}}(u_t) = \frac{1}{K}\sum_{i=1}^{K} \left\| J_{\pi_\phi^{(i)}}(u_t) \right\|_F^2 . \tag{7}$$

The penalty is on probabilities rather than logits, because the smoothness assumption of Section 2 is stated on transition probabilities. The two differ by the softmax Jacobian factor, whose norm is at most $1/2$ and falls toward zero as a row approaches a one-hot distribution (proof in Appendix C.3). A uniform penalty on the logit Jacobian would therefore constrain the probability sensitivity of nearly one-hot rows far more tightly than that of other rows, effectively freezing the rows the model is most confident about. The same factor opens a shortcut for the probability penalty: it can be lowered by sharpening the rows rather than by becoming less sensitive. Section 5.3 tests for this shortcut and finds that GPLD leads to the opposite.

The full objective, with B sequences of length L per minibatch and n indexing optimizer updates, is

$$\mathcal{L}_{\mathrm{GPLD}} = \mathbb{E}_{q_\phi}\left[\frac{1}{L}\sum_{t=1}^{L}\left(\beta_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred},t} + \beta_{\mathrm{dyn}}\mathcal{L}_{\mathrm{dyn},t} + \beta_{\mathrm{rep}}\mathcal{L}_{\mathrm{rep},t} + \lambda_{(n)}\mathcal{R}_{\mathrm{GPLD}}(u_t)\right)\right], \tag{8}$$

with the three DreamerV3 terms and their weights unchanged.

In the finite-state objective the likelihood is a sum over transitions, so its curvature grows with the visit count and the mixing weight withdraws on its own (Section 2). A latent world model is trained

on a minibatch average, whose minimizer does not change as data accumulates, and a continuous state space has no visit counts to scale the penalty by. Without a schedule the penalty would keep its full weight, and its bias, throughout training. We therefore decay the coefficient as $\lambda_{(n)} = \lambda_0/\sqrt{1+n/c}$, so that the penalty withdraws as data accumulates. The analogous optimal rate is $1/n$ (Appendix C.1 and Theorem 4); the slower square-root rate is an empirical choice that keeps the penalty active longer; decaying the coefficient outperforms holding it constant (Appendix K), and a $1/n$ schedule is not compared.

4.4 Estimation

Evaluating (7) exactly costs KC backward passes per state. Instead, for each row we draw a Rademacher probe ξ_i (Hutchinson, 1989) and form $g_i = \nabla_{u_t}\langle \xi_i, \pi_\phi^{(i)}(u_t)\rangle = J_i^\top \xi_i$ with one vector–Jacobian product. Since $\mathbb{E}[\xi_i \xi_i^\top] = I$, $\frac{1}{K}\sum_i \|g_i\|_2^2$ is an unbiased estimate of $\mathcal{R}_{\mathrm{GPLD}}$. The K categoricals are independent given u_t, so each row receives its own probe; this also keeps the cross-row terms $\xi_i^\top J_i J_j^\top \xi_j$ out of the estimate and lowers its variance (Appendix H). The penalty is evaluated on a uniformly drawn fraction ρ of the minibatch states, which keeps the estimate of the minibatch term unbiased: ρ sets how many states receive the penalty, not how strongly it is applied. Training on the penalty requires double backpropagation (Drucker & LeCun, 1992): one backward pass forms each g_i, and a second passes through it to update the parameters. This second pass, not the number of probes, sets the runtime cost (Section 5.5).

5 Experiments

Both arms run the same DreamerV3 configuration, rerun rather than taken from published numbers so that the comparison isolates the penalty. It is the published setting for this suite except for an imagination horizon of 25 instead of 15, fixed before this study. GPLD uses a single configuration on every task, chosen in the ablations of Appendix K and listed in Appendix H. Both arms use the current official implementation of DreamerV3. The DreamerV3 scores released with it come from an earlier version of the code and are higher on several locomotion tasks, at either horizon (Appendix H, Table 5). Because both arms here share every setting, the comparison measures the effect of adding the penalty to a fixed agent. We report two metrics. **Final return** is the mean evaluation return over the last 100k environment steps of the budget. **nAUC** is the area under the evaluation-return curve divided by its span: the mean return an arm sustained across training, and the more direct measure of sample efficiency. A task's normalized return is GPLD's mean final return divided by DreamerV3's (normalized nAUC likewise), and aggregates average it over tasks. Throughout, $\pm$ is the standard error of the mean over seeds ($n = 5$ per arm unless stated), and a bracketed pair $[a, b]$ after a normalized value is its 95% confidence interval from a stratified bootstrap over seeds. In performance tables, bold marks an entry better than DreamerV3's by more than the sum of the two standard errors.

5.1 Proprioceptive DeepMind Control

Where DreamerV3 still has room to improve, GPLD improves on it substantially. Table 1 reports the 18-task proprioceptive benchmark at 1M environment steps, split by whether DreamerV3 has already saturated: a task is *saturated* when the baseline's normalized area under its learning curve, A, reaches 0.80, a split computed from the baseline alone. On the 12 unsaturated tasks GPLD raises mean final return from 589.5 to 647.1, a normalized return of 1.181 $[1.096, 1.664]$, and its normalized nAUC is 1.200 $[1.087, 1.612]$, a gain of 20.0% in the return sustained across training (Table 10; learning curves in Appendix I). On the 6 saturated tasks the normalized return is 1.009, with an interval $[0.997, 1.021]$ that contains 1.0: where the baseline already sits at the ceiling the difference is negligible, and the benchmark-wide normalized return of 1.124 is the large effect diluted by tasks with nothing left to win. The conclusion does not depend on where the saturation line is drawn: moving it anywhere from 0.70 to 0.90 changes the unsaturated gain only between $+49.7$ and $+61.9$ points (Table 11). Across all 18 tasks, GPLD's mean final return is higher on 15 (two-sided sign test, $p = 0.0075$).

Table 1: Proprioceptive DMC at 1M environment steps, mean ± SEM over seeds, split by saturation (Section 5.1). Normalized is GPLD's task mean over DreamerV3's; aggregates are its mean and median over tasks. *Three seeds per arm.

Task	A	DreamerV3	GPLD	Normalized
Unsaturated ($A < 0.80$)				
`hopper-hop`	0.046	99.0 ± 43.1	$\mathbf{171.1 \pm 16.2}$	1.729
`walker-run`	0.233	456.3 ± 61.5	$\mathbf{625.4 \pm 48.5}$	1.371
`acrobot-swingup-sparse`	0.034	91.9 ± 10.0	$\mathbf{124.4 \pm 14.5}$	1.353
`finger-spin`	0.525	614.9 ± 44.3	$\mathbf{732.9 \pm 57.3}$	1.192
`acrobot-swingup`	0.180	268.7 ± 6.1	$\mathbf{319.9 \pm 20.7}$	1.191
`cheetah-run`	0.476	638.8 ± 40.2	$\mathbf{741.8 \pm 30.6}$	1.161
`walker-walk`	0.656	877.6 ± 50.0	$\mathbf{937.8 \pm 9.4}$	1.069
`pendulum-swingup`	0.746	762.5 ± 22.8	798.5 ± 32.0	1.047
`finger-turn-easy`	0.777	891.4 ± 25.1	928.0 ± 17.8	1.041
`hopper-stand`	0.309	681.5 ± 85.2	702.4 ± 62.4	1.031
`cartpole-swingup-sparse`	0.572	789.6 ± 15.9	791.6 ± 19.4	1.003
`finger-turn-hard`	0.641	901.5 ± 18.3	891.8 ± 34.2	0.989
Unsaturated, mean over tasks				1.181 [1.096, 1.664]
Unsaturated, median over tasks				1.115 [1.041, 1.192]
Saturated ($A \geq 0.80$)				
`cartpole-swingup`*	0.819	838.4 ± 10.6	$\mathbf{871.0 \pm 3.8}$	1.039
`cartpole-balance`*	0.968	969.7 ± 22.7	$\mathbf{996.9 \pm 0.2}$	1.028
`walker-stand`	0.850	943.3 ± 4.1	$\mathbf{956.8 \pm 3.8}$	1.014
`reacher-hard`	0.865	941.9 ± 15.6	949.8 ± 4.9	1.008
`reacher-easy`	0.926	966.0 ± 4.0	965.6 ± 6.1	1.000
`cartpole-balance-sparse`*	0.962	999.8 ± 0.2	963.0 ± 25.9	0.963
Saturated, mean over tasks				1.009 [0.997, 1.021]
Saturated, median over tasks				1.011 [1.000, 1.024]
All 18, mean over tasks				1.124 [1.067, 1.457]
All 18, median over tasks				1.040 [1.016, 1.071]

Table 2: `humanoid` and `dog` at each task's horizon, mean ± SEM over each task's six seeds completed by both arms.

		Final return		nAUC	
Task	Horizon	DreamerV3	GPLD	DreamerV3	GPLD
`dog-run`	12M	424.2 ± 31.5	480.4 ± 36.1	105.1 ± 11.0	$\mathbf{133.6 \pm 12.6}$
`dog-stand`	8M	618.0 ± 38.3	$\mathbf{777.6 \pm 40.0}$	142.4 ± 10.2	$\mathbf{201.4 \pm 18.1}$
`dog-walk`	12M	459.2 ± 82.2	$\mathbf{735.2 \pm 47.7}$	87.8 ± 15.7	$\mathbf{153.3 \pm 23.5}$
`humanoid-run`	12M	281.2 ± 28.6	283.2 ± 6.7	72.5 ± 7.6	81.2 ± 2.7
`humanoid-stand`	12M	454.6 ± 36.4	$\mathbf{573.2 \pm 61.0}$	53.9 ± 13.8	$\mathbf{109.5 \pm 23.7}$
`humanoid-walk`	12M	688.3 ± 43.4	680.3 ± 38.1	190.8 ± 20.9	172.8 ± 22.3

Scaling with embodiment complexity. The gains persist on the largest bodies in DMC. The `humanoid` (67 observation and 21 action dimensions) and `dog` (223 and 38) bodies require coordinating many more actuated degrees of freedom, and Table 2 reports both metrics over horizons of eight to twelve million steps. GPLD improves final return on five tasks and nAUC on five tasks of the six. The normalized gain is larger in nAUC (mean +41.4%, median +34.2%) than in final return (mean +20.8%, median +19.5%), the ordering a sample-efficiency claim predicts. The one regression is `humanoid-walk` (-1.2% final, -9.4% nAUC). Learning curves and per-seed paired gains are in Appendix I.

Quadruped. On `quadruped` (78 observation and 12 action dimensions) both arms eventually solve both tasks, so what matters is how soon, and GPLD gets there first (Figure 2). At 3M steps its mean return is 857.0 on `run` and 845.9 on `walk`, against 637.3 and 516.5 for DreamerV3, and its median seed reaches a return of 800 about 0.39M steps earlier on `run` and 0.49M earlier on `walk`. Over the full 4M steps GPLD's nAUC is 24.1% higher on `run` and 32.9% higher on `walk`; DreamerV3 catches up once both arms reach the ceiling. One DreamerV3 seed on `walk` never learns stably, and with it removed from both arms GPLD is still 38% ahead at 3M (Table 13 in Appendix I).

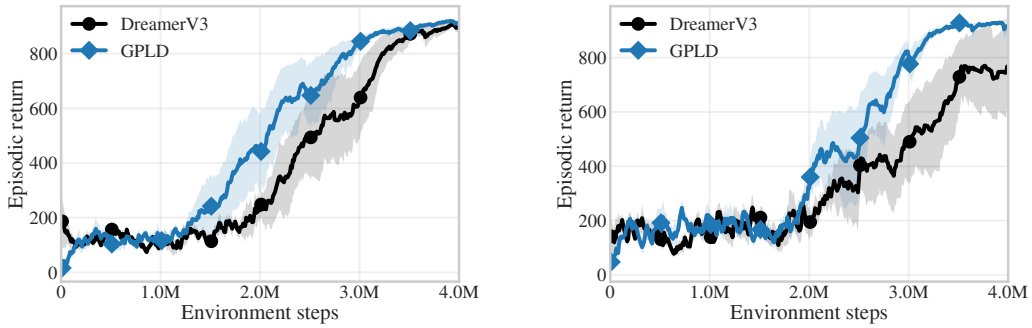


Figure 2: `quadruped-run` (left) and `quadruped-walk` (right): GPLD reaches high return earlier, and DreamerV3 catches up at the ceiling. Mean ± SEM over seeds.

5.2 Pixel observations and the representation bottleneck

Under pixel observations, where the agent sees rendered camera images instead of joint positions and velocities, the gain largely disappears. Across all 20 pixel tasks at 1M steps, GPLD's mean final return is 740.8 against 729.5 for DreamerV3 and its nAUC 580.1 against 590.3, and every interval for the normalized return contains 1 (Table 8 in Appendix I). The penalty limits sensitivity with respect to the encoder's output, so it can share information between states only once the encoder places similar states near each other; learned from pixels, that embedding is itself still forming. If this is the bottleneck, a pretrained encoder should restore the gain. In a controlled diagnostic on `quadruped-walk` (Appendix K.1), warm-starting the encoder and decoder moves GPLD from a deficit of 5.5% in nAUC to a lead of 18.2%. With one task at $n = 5$ seeds the interaction is directional rather than statistically separated, so we treat it as a diagnostic.

5.3 Is the learned model smoother?

A lower penalty does not by itself mean a smoother posterior. The penalty can be lowered in two ways: by making the posterior respond less to its input, which is smoothing, or by pushing its categorical rows toward one-hot distributions, where the softmax factor vanishes (Section 4) and the penalty falls with no change in how the posterior responds to its input. Two measurements at the 1M-step checkpoint tell them apart (Appendix L). First, sharpening would lower entropy, yet GPLD's posterior entropy is higher than DreamerV3's on all three tasks tested, with no more saturated rows. Second, the Jacobian of the logits, taken before the softmax, removes that factor, and pure sharpening would raise it; GPLD's is 27% to 35% lower than DreamerV3's. Both energies are measured relative to the scale of the posterior's inputs, since a model could otherwise lower the raw energy by enlarging its inputs without changing the function. GPLD makes the posterior less sensitive to its input rather than sharper.

A smoother posterior should also predict no worse over multi-step rollouts, which is how the policy optimization is performed. Over 25-step rollouts from real states on recorded actions, GPLD's cumulative prediction error is lower than DreamerV3's on ten of the 14 task and quantity pairs measured, significantly so ($|z| > 2$) on one, and significantly higher on zero, on windows from policies that neither arm generated (Appendix M).

5.4 Other smoothness regularizers

We compare GPLD with four regularizers that also act on the posterior's input sensitivity, each chosen on three tuning tasks: virtual adversarial training (VAT) (Miyato et al., 2019), which penalizes the change in the posterior under a worst-case perturbation of its input; noise consistency, the random-perturbation variant of VAT (Miyato et al., 2019), which penalizes the change under a random perturbation instead; noise injection, which trains on randomly perturbed inputs (Bishop, 1995); and spectral normalization (SN) (Miyato et al., 2018), which caps the spectral norm of the posterior's output layer. GPLD does better than all four (Figure 3). Across the 12 unsaturated tasks

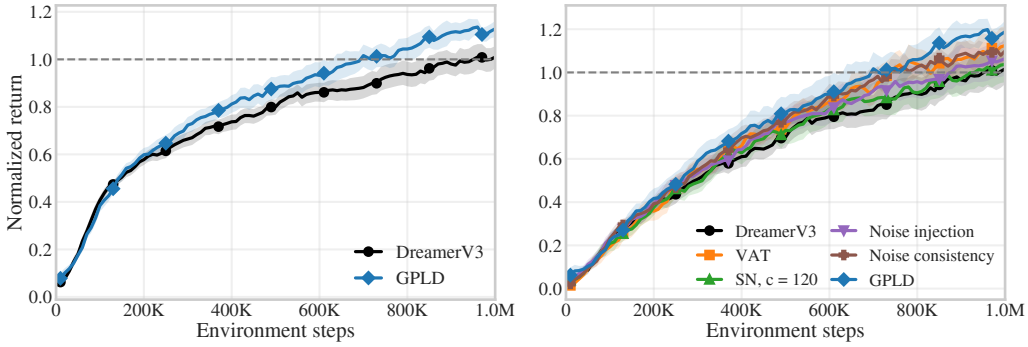


Figure 3: Normalized return over training (each task's return divided by DreamerV3's mean final return on that task, averaged over tasks). GPLD stays ahead of DreamerV3 for most of training, averaged over all 18 proprioceptive tasks (left), and finishes above VAT, SN ($c = 120$), noise injection and noise consistency on the 12 unsaturated tasks (right). Shaded: 95% bootstrap interval; curves smoothed for display (Appendix H).

its normalized return is 1.204, against 1.118 for VAT, 1.110 for noise consistency, 1.064 for noise injection and 1.042 for SN. Its mean final return is 650.6, against at most 622.9 for a comparator and 582.7 for DreamerV3. The tuning tasks use six seeds here, hence 1.204 rather than 1.181 (Appendix N).

5.5 Cost

Training on a penalty that is itself a gradient requires a second backward pass through part of each minibatch. This raises the wall-clock time of a complete run by a factor of 1.50 on average over the 20 proprioceptive tasks timed (median 1.47; Table 6), and the cost is set by the fraction ρ of states penalized rather than by the number of probes (Appendix H).

6 Scope and Limitations

The evaluation covers continuous control from `cartpole` to `dog` over pre-specified horizons, with one configuration per benchmark; the ManiSkill2 run, where GPLD reaches a 0.9 success rate earlier (Appendix J), and the warm-start diagnostic use a larger coefficient. Transfer to other agents, such as TD-MPC2 (Hansen et al., 2024), is not established. Comparisons are at equal environment steps: the penalty makes a run 1.50 times as long (Section 5.5), but 1.5 times as many gradient updates do not help DreamerV3 on the tuning tasks (Table 7): the gain comes from the penalty, not extra compute. The gain is not uniform: on `humanoid-walk` the arms finish level, and under pixels the learned representation appears to limit it. The penalty is coordinate-dependent: rescaling h_t or e_t changes $\left\| J_{\Pi_\phi} \right\|_F^2$ for a functionally equivalent model, so the smoothing of Section 5.3 describes the learned representation rather than the environment. The coefficient and sampling fraction are global and static; making them state-dependent needs a local smoothness diagnostic.

7 Conclusion

A finite-difference penalty on transition laws induces a shrinkage estimator that shares information across neighboring states most where data is scarce, and its small-neighborhood limit is Frobenius Jacobian energy; in a controlled tabular MDP the benefit appears as an effective-sample-size gain. Applied to DreamerV3's posterior under one fixed configuration, it raises final return on 15 of 18 proprioceptive tasks, mean final return by 9.8% and nAUC by 20.0% where the baseline has room to improve, and final return by 20.7% on the largest embodiments; on `quadruped` it reaches high return earlier. The posterior becomes less sensitive to its input without becoming sharper, as does the unpenalized prior on the three tasks measured, and multi-step prediction error is lower on most measured quantities and significantly higher on none.

Ethics statement

This work aims to improve the sample efficiency of model-based reinforcement learning by sharing information between neighboring states in learned latent dynamics. More sample-efficient world models reduce the interaction needed to train control agents, which lowers simulation cost and makes model-based reinforcement learning more practical. As with other reinforcement-learning methods, the downstream impact depends on the systems in which trained agents are deployed; use in physical control would require the same safety validation expected of other learned controllers. All experiments use simulated benchmarks (DeepMind Control Suite and ManiSkill2) and involve no human subjects, personal data or dataset release.

Reproducibility statement

Appendix H gives the configuration shared by both arms, the full hyperparameter list, per-task seed counts and hardware, and Section 5 the evaluation protocol. Assumption 1 is stated explicitly, and the results of Sections 2 and 4 are proved in Appendices C.1 to C.3. The tabular environment is specified in Appendix G. Per-task results for every benchmark appear in Appendix I so that all reported aggregates are recomputable. The supplementary material contains the anonymized source code, launch scripts for every experiment, the processed results of all runs, and scripts that regenerate every table and figure of the paper from them, together with the code for the tabular experiment and the smoothness and open-loop diagnostics.

AI use statement

We used generative AI tools (Claude) to edit and refine manuscript text, to write and review the analysis scripts that aggregate our training logs into the reported tables and statistics, to draft and check several appendix derivations, and to assist with running and monitoring experiments. The research questions, the method, and the experimental design are the authors'. All derivations were verified by the authors, all numbers are generated from logged results by scripts the authors reviewed, and the authors take full responsibility for the content.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 264–273. PMLR, 2018.

Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J. Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91(3): 325–349, 2013.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, Cham, 2 edition, 2017. ISBN 978-3-319-48310-8. doi: 10.1007/978-3-319-48311-5.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8):1289–1308, 2008.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Chris M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

Johan Bjorck, Carla P. Gomes, and Kilian Q. Weinberger. Towards deeper deep reinforcement learning with spectral normalization. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Justin A. Boyan and Andrew W. Moore. Generalization in reinforcement learning: Safely approximating the value function. In *Advances in Neural Information Processing Systems*, volume 7, 1994.

Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S. Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, and Xue Bin Peng. Learning smooth humanoid locomotion through Lipschitz-constrained policies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Guilherme Christmann, Ying-Sheng Luo, Hanjaya Mandala, and Wei-Chao Chen. Benchmarking smoothness and reducing high-frequency oscillations in continuous control policies. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024.

Damek Davis and Wotao Yin. Faster convergence rates of relaxed Peaceman–Rachford and ADMM under regularity assumptions. *Mathematics of Operations Research*, 42(3):783–805, 2017.

Benoit Dherin, Michael Munn, Mihaela Rosca, and David G. T. Barrett. Why neural networks find simple solutions: The many regularizers of geometric complexity. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Omar Darwiche Domingues, Pierre Ménard, Matteo Pirotta, Emilie Kaufmann, and Michal Valko. Kernel-based reinforcement learning: A finite-time analysis. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

Harris Drucker and Yann LeCun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1–3):293–318, 1992.

Qiaoyi Fang, Weiyu Du, Hang Wang, and Junshan Zhang. Towards unraveling and improving generalization in world models. arXiv preprint arXiv:2501.00195, 2024.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous Markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *Conference on Robot Learning*, 2021.

Daniel Gabay. Applications of the method of multipliers to variational inequalities. In Michel Fortin and Roland Glowinski (eds.), *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, volume 15 of *Studies in Mathematics and Its Applications*, pp. 299–331. North-Holland, 1983.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179, 2019.

Pontus Giselsson. Tight global linear convergence rate bounds for Douglas–Rachford splitting. *Journal of Fixed Point Theory and Applications*, 19:2241–2270, 2017.

Pontus Giselsson and Stephen Boyd. Linear convergence and metric selection for Douglas–Rachford splitting and ADMM. *IEEE Transactions on Automatic Control*, 62(2):532–544, 2017. doi: 10.1109/TAC.2016.2564160.

Florin Gogianu, Tudor Berariu, Mihaela Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalisation for deep reinforcement learning: An optimisation perspective. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

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.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

David Ha and Jürgen Schmidhuber. World models. `https://zenodo.org/record/1207631`, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization. arXiv preprint arXiv:1908.02729, 2019.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics – Simulation and Computation*, 18(3):1059–1076, 1989.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2–3):209–232, 2002.

Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. doi: 10.1137/0716071.

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

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(27):815–857, 2008.

Siddharth Mysore, Bassel Mabsout, Renato Mancuso, and Kate Saenko. Regularizing action policies for smooth control with reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2021.

Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2 edition, 2018. doi: 10.1007/978-3-319-91578-4.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine Learning*, 49(2–3): 161–178, 2002.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 2020.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Mihaela Rosca, Theophane Weber, Arthur Gretton, and Shakir Mohamed. A case for new neural network smoothness constraints. In Jessica Zosa Forde, Francisco Ruiz, Melanie F. Pradier, and Aaron Schein (eds.), *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, volume 137 of *Proceedings of Machine Learning Research*, pp. 21–32. PMLR, 12 Dec 2020. URL https://proceedings.mlr.press/v137/rosca20a.html.

Tankred Saanum, Peter Dayan, and Eric Schulz. Simplifying latent dynamics with softly state-invariant world models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Alex J. Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines*, pp. 144–158. Springer, 2003.

Dávid Terjék. Adversarial lipschitz regularization. In *International Conference on Learning Representations*, 2020.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Haoran Wang, Yaoru Sun, Zeshen Tang, Haibo Shi, and Chenyuan Jiao. Hg2p: Hippocampus-inspired high-reward graph and model-free q-gradient penalty for path planning and motion control. *Neural Networks*, 191:107781, 2025. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2025.107781. URL https://www.sciencedirect.com/science/article/pii/S0893608025006616.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J. Weinberger. Inequalities for the L_1 deviation of the empirical distribution. Technical Report HPL-2003-97, Hewlett-Packard Laboratories, 2003.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. RePo: Resilient model-based reinforcement learning by regularizing posterior predictability. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

CONTENTS OF THE APPENDIX

	Section	Contents
Theory		
A	Extended related work	Full discussion of related work
B	DreamerV3 background	World model, losses and imagination
C	Proofs and derivations	Finite-state construction, continuous limit, softmax factor
D	Bias–variance trade-off	Graph-ridge form, exact bias and variance, optimal λ, non-diagonal noise
E	Planning bound	From transition error to planning error
F	Exact solution (ADMM)	Splitting, updates and convergence
Experimental setup		
G	Tabular experiments	`var-gaus`, protocol, estimators and figures
H	Deep experiments	Hyperparameters, evaluation, hardware and wall-clock
Results		
I	Per-task results	nAUC on proprioceptive DMC, dog and humanoid, pixels
J	Manipulation	ManiSkill2 `PickCube-v0`
K	Ablations	Sampling fraction, penalized map, coefficient, schedule, warm start
L	Smoothness diagnostics	Entropy and scale-invariant Jacobian energy
M	Open-loop prediction	Protocol and per-task results
N	Other regularizers	SN, VAT and random perturbations: implementation, selection, results
O	Seed reliability	Paired per-seed tests

A Extended related work

Smoothness in RL. Smoothness has long formalized how information generalizes across nearby states (Boyan & Moore, 1994; Munos & Szepesvári, 2008). Kernel-based RL builds smoothed estimates of the transition law explicitly (Ormoneit & Sen, 2002; Domingues et al., 2021); we obtain the smoothing as a regularizer on a learned model, which is what transfers to deep world models, and use the finite-state case as a derivation rather than as a proposed tabular method. Further, Asadi et al. (2018) show that Lipschitz properties of a learned model bound how one-step error propagates under repeated application. In this literature smoothness is *assumed* to obtain guarantees, or enforced globally through a Lipschitz bound on the whole model; it does not identify which map inside a stochastic latent world model to regularize, or what the penalty does to estimation. Unconstrained networks can also produce high-frequency oscillations that degrade stability and generalization (Rosca et al., 2020; Christmann et al., 2024).

Gradient and Jacobian penalties. Gradient penalties encourage local smoothness in generative and adversarial training (Gulrajani et al., 2017; Terjék, 2020; Goodfellow et al., 2015). Contractive autoencoders penalize the Jacobian energy of an encoder (Rifai et al., 2011), and Jacobian regularization for classifiers estimates the same quantity with random projections for a decision function (Hoffman et al., 2019). Training with input noise is, to first order, the same Jacobian penalty (Bishop, 1995). Spectral normalization bounds operator norms architecturally (Miyato et al., 2018; Gogianu et al., 2021; Bjorck et al., 2021), a global, worst-case constraint that also restricts regions that are in fact regular. Virtual adversarial training smooths a model's output distribution against worst-case input perturbations (Miyato et al., 2019), and implicit regularizers act on comparable derivative quantities (Dherin et al., 2022). In RL, such penalties target Q-functions (Wang et al., 2025), policies (Chen et al., 2025), actions (Mysore et al., 2021) and energy use (Fu et al., 2021), and actions can also be smoothed with a low-pass filter (Peng et al., 2020). Graph-Laplacian penalties regularize estimates over a neighborhood graph in semi-supervised learning (Zhou et al., 2004). None is derived from an estimation argument about transition laws. Penalties on the model of a model-based agent exist: DeepMDP enforces Lipschitz latent transition and reward models with a gradient penalty (Gelada et al., 2019), and Fang et al. (2024) add a Jacobian regularizer to world-model dynamics to limit compounding error. Other world-model regularizers limit the information in the representation, such as the predictability of the posterior (Zhu et al., 2023) or how much an action's effect depends on the latent state (Saanum et al., 2024), rather than its local sensitivity. GPLD differs in the map it penalizes, the posterior probability table, and in deriving the penalty's form and schedule from transition estimation. Appendix N compares GPLD with spectral normalization, virtual adversarial training and two random-perturbation regularizers applied to the same posterior map.

Model error and multi-step training. That one-step error compounds under repeated model application is long established (Venkatraman et al., 2015). PlaNet proposed a multi-step objective, latent overshooting (Hafner et al., 2019); later Dreamer versions train on one-step terms (Hafner et al., 2025). GPLD addresses the same failure differently. It constrains the local sensitivity that governs how error is amplified, rather than training against longer rollouts.

B DreamerV3 background

This appendix defines the DreamerV3 components and losses referred to in the main text (Hafner et al., 2025). All settings are those of the `size12m` configuration used throughout (Appendix H).

The recurrent state-space model. At step t the agent observes o_t and the model state has two parts, a deterministic recurrent state h_t and a stochastic latent state z_t. The model consists of

$$\begin{aligned} \text{sequence model:} \quad & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}), & \text{encoder:} \quad & e_t = \mathrm{enc}_\phi(o_t), \\ \text{posterior:} \quad & z_t \sim q_\phi(z_t \mid h_t, e_t), & \text{prior:} \quad & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t), \\ \text{reward and continuation:} \quad & \hat{r}_t,\ \hat{c}_t \sim p_\phi(\cdot \mid h_t, z_t), & \text{decoder:} \quad & \hat{o}_t \sim p_\phi(\cdot \mid h_t, z_t). \end{aligned}$$

The posterior infers the latent state from the current observation; the prior predicts it without the observation, from h_t alone. The encoder output e_t enters the model only through the posterior. The latent z_t consists of $K = 32$ categorical variables with $C = 16$ classes each, and $\dim(h_t) = 2048$.

The probability table. The posterior head is a one-hidden-layer network that maps $u_t = [h_t, e_t]$ to a $K \times C$ array of logits. A softmax over each row, mixed with 1% of the uniform distribution to keep every probability away from zero, gives the posterior probability table $\Pi_\phi(u_t) \in \Delta^{C-1} \times \cdots \times \Delta^{C-1}$, whose i-th row is the distribution of the i-th categorical variable. The prior head produces its table from h_t in the same way, with two hidden layers. GPLD penalizes the Jacobian of Π_ϕ, after the uniform mixing, with respect to u_t.

Straight-through sampling. Each row of z_t is a one-hot sample from the corresponding row of Π_ϕ. Sampling is not differentiable, so DreamerV3 uses the straight-through estimator $z_t = \mathrm{onehot} + \Pi_\phi - \mathrm{sg}(\Pi_\phi)$, where the stop-gradient operator $\mathrm{sg}(\cdot)$ returns its argument unchanged in the forward pass and is treated as a constant when differentiating. The forward value is the one-hot sample, and gradients from every consumer of z_t reach the posterior parameters through Π_ϕ.

World-model loss. For each position t of a replayed sequence,

$$\begin{aligned} \mathcal{L}_{\mathrm{pred},t} &= -\log p_\phi(o_t \mid h_t, z_t) - \log p_\phi(r_t \mid h_t, z_t) - \log p_\phi(c_t \mid h_t, z_t), \\ \mathcal{L}_{\mathrm{dyn},t} &= \max\bigl(1,\ \mathrm{KL}\bigl[\mathrm{sg}(q_\phi(z_t \mid h_t, e_t)) \,\big\|\, p_\phi(\hat{z}_t \mid h_t)\bigr]\bigr), \\ \mathcal{L}_{\mathrm{rep},t} &= \max\bigl(1,\ \mathrm{KL}\bigl[q_\phi(z_t \mid h_t, e_t) \,\big\|\, \mathrm{sg}(p_\phi(\hat{z}_t \mid h_t))\bigr]\bigr), \end{aligned}$$

weighted by $\beta_{\mathrm{pred}} = 1$, $\beta_{\mathrm{dyn}} = 1$ and $\beta_{\mathrm{rep}} = 0.1$ and averaged over the $B \times L$ positions of the minibatch, as in (8). The two KL terms are the same divergence with the stop-gradient placed differently. The *dynamics* loss treats the posterior as a fixed target and trains the prior to predict it; the *representation* loss treats the prior as fixed and pulls the posterior, and through it the encoder, toward what the prior can predict. The asymmetric weights move the prior toward the posterior more strongly than the reverse. Each KL is summed over the K categorical variables, and the clip at one nat (*free nats*) is applied to that sum, so it is a single budget for all K variables at each position rather than one nat per variable. Observations are regressed in the symlog space and rewards with a two-hot discretized distribution (Hafner et al., 2025).

Imagination and actor–critic. From every posterior state in the replayed minibatch, the model rolls forward $H = 25$ steps using the prior alone: the policy samples $a_\tau \sim \pi(\cdot \mid h_\tau, z_\tau)$, the sequence model gives $h_{\tau+1} = f_\phi(h_\tau, z_\tau, a_\tau)$, and the next latent is drawn from the prior, $z_{\tau+1} \sim p_\phi(\cdot \mid h_{\tau+1})$. The reward and continuation heads score the imagined states. The critic regresses λ-returns ($\lambda = 0.95$, discount $1 - 1/333$) computed on these imagined trajectories and, with weight 0.3, on replayed ones. The actor maximizes the imagined returns, normalized by the 5th-to-95th percentile range, with an entropy bonus of 3×10^{-4}.

Consequences for GPLD. Three facts from this appendix are used in the main text. The posterior is the only map that receives the observation, so every gradient that reaches the encoder passes through Π_ϕ (Section 4). Since $u_t = [h_t, e_t]$, the Jacobian splits into two column blocks and $\bigl\|J_{\Pi_\phi}\bigr\|_F^2 = \bigl\|J_{\Pi_\phi}^{(h)}\bigr\|_F^2 + \bigl\|J_{\Pi_\phi}^{(e)}\bigr\|_F^2$. Imagination rolls the prior, not the posterior, and the prior

is trained toward stop-gradient posterior targets by $\mathcal{L}_{\mathrm{dyn}}$; this is the path by which a penalty on the posterior reaches imagination. And because the KL terms are summed over the K variables while $\mathcal{R}_{\mathrm{GPLD}}$ is averaged over them, the coefficient λ_0 and the weight β_{dyn} are not comparable in magnitude.

Which map to penalize. Table 3 scores each candidate map against the four properties the derivation of Section 2 calls for (Section 4).

Table 3: Candidate maps for the penalty inside the recurrent state-space model. The derivation calls for a distribution over next latent states, grounded in environment data, that shapes the representation and reaches imagination. Only the posterior satisfies the conjunction.

Map	Distribution over next latent state	Grounded in observations	Trains the representation	Reaches imagination
Encoder enc_ϕ	no	yes	yes	indirectly
Recurrent map f_ϕ	no	indirectly	yes	yes
Prior $p_\phi(\hat{z}_t \mid h_t)$	yes	**no**	no	yes
Prediction heads	no	yes	no	yes
Composite imagination map	yes	no	no	yes
Posterior $q_\phi(z_t \mid h_t, e_t)$	**yes**	**yes**	**yes**	**via dynamics KL**

C PROOFS AND DERIVATIONS

C.1 DETAILS OF THE FINITE-STATE CONSTRUCTION

Sampling model. Fix an action a. Expectations below are over the successor states recorded at each visit, with the visit counts N_s and $\{N_k\}$ held fixed. Each visit to (s, a) draws its successor independently from p_s, so given N_s each count $N(s, a, s')$ is $\mathrm{Binomial}(N_s, p_s(s'))$, and successor draws at distinct states are independent. The counts themselves are not independent across states, since a trajectory that visits s is likely to visit its neighbors as well; only the draws given the counts are. The embedding, the neighborhoods, the weights and the true rows are deterministic, so $\mathbb{E}\bar{p}_s = \sum_k w_{sk} p_k$ exactly and the bias below is a deterministic vector. Holding the counts fixed is exact when they are set in advance, as with a generative model. Under a policy that collects data adaptively the number of visits can depend on the successors already observed, and the statements hold for the estimate formed from any fixed number of visits; a random N_s requires a union bound over that number. The tabular experiment of Section 3 measures the effect directly rather than relying on these bounds. Every neighbor used in pooling is assumed visited, $N_k \geq 1$.

The row bound (1). Write $q_{s'} := p_s(s')$. Each $\widehat{p}_s(s')$ is unbiased with variance $q_{s'}(1 - q_{s'})/N_s$. Jensen's inequality for the concave map $x \mapsto \sqrt{x}$ gives $\mathbb{E}|Z| \leq \sqrt{\mathbb{E}Z^2}$ for $Z = \widehat{p}_s(s') - q_{s'}$, so $\mathbb{E}|\widehat{p}_s(s') - q_{s'}| \leq \sqrt{q_{s'}(1 - q_{s'})/N_s}$. Summing over s' and applying Cauchy–Schwarz with the constant vector,

$$\sum_{s'} \sqrt{q_{s'}(1 - q_{s'})} \;\leq\; \sqrt{|\mathcal{S}|}\,\sqrt{\textstyle\sum_{s'} q_{s'}(1 - q_{s'})} \;=\; \sqrt{|\mathcal{S}|\big(1 - \textstyle\sum_{s'} q_{s'}^2\big)},$$

using $\sum_{s'} q_{s'} = 1$, which is the first inequality. Cauchy–Schwarz again gives $\sum_{s'} q_{s'}^2 \geq 1/|\mathcal{S}|$, with equality exactly for the uniform row, which gives the second. The quantity $\sum_{s'} q_{s'}^2$ is the collision probability of the row, so the bound is tightest for near-deterministic rows and worst for uniform ones. Under a budget T shared over $|\mathcal{S}||\mathcal{A}|$ rows, uniform visitation gives $N_s = T/(|\mathcal{S}||\mathcal{A}|)$ and a per-row error of $\sqrt{|\mathcal{S}|(|\mathcal{S}| - 1)|\mathcal{A}|/T}$. Because $x \mapsto x^{-1/2}$ is convex, $\mathbb{E}[N_s^{-1/2}] \geq (\mathbb{E}N_s)^{-1/2}$, so substituting the mean count is optimistic, and policy-driven visitation, which is non-uniform, makes the average error larger still.

Consequences of Assumption 1. Because $\mathcal{S}$ is finite, $\psi(\mathcal{S})$ is a finite point set and there is no function on $\mathbb{R}^d$ to differentiate; what is assumed is the existence of a differentiable extension. Three consequences follow. G_s is a derivative of the extension, not of P. The assumption requires ψ to separate states with differing dynamics: if $\psi(s) = \psi(k)$ while $p_s \neq p_k$, no such $\widetilde{P}$ exists. And the second derivative is uniformly bounded on U, by the one constant M at every point, which is what permits remainders at different neighbors to be aggregated.

Bias of the pooled estimate. Taylor's theorem with remainder, applied along the segment from $\psi(s)$ to $\psi(k)$, gives $p_k = p_s + G_s(\psi(k) - \psi(s)) + R_{sk}$ with $\|R_{sk}\|_2 \leq \frac{M}{2} d_{sk}^2$. The segment lies in U, since $d_{sk} \leq \epsilon$ and U contains the closed ϵ-ball around $\psi(s)$. Averaging with weights that sum to one,

$$b := \mathbb{E}\bar{p}_s - p_s = G_s \delta_s + \sum_k w_{sk} R_{sk}, \qquad \delta_s := \sum_k w_{sk}\big(\psi(k) - \psi(s)\big),$$

so $B := \|b\|_2 \leq \|G_s\|\,\|\delta_s\|_2 + \frac{M}{2}\bar{\epsilon}^{\,2}$ with $\bar{\epsilon}^{\,2} := \sum_k w_{sk} d_{sk}^2 \leq \epsilon^2$. Call the weights *balanced* about s when $\delta_s = 0$. An isotropic weighting, one that depends on d_{sk} alone, is balanced whenever the neighborhood is symmetric about $\psi(s)$, since equal and opposite displacements then receive equal weight; it is asymmetry of the neighborhood, not the choice of weighting, that breaks the condition. Balance is an idealized property of the embedded neighborhood, not one guaranteed for an arbitrary embedding. Under it the bias is $\mathcal{O}(\bar{\epsilon}^{\,2})$. Without it, $\|\delta_s\|_2 \leq \sum_k w_{sk} d_{sk} \leq \bar{\epsilon}$ by Jensen's inequality, and the bias is first order, $\mathcal{O}(\bar{\epsilon})$.

Variance and the optimal mixing weight (3). The isolated estimate has $V_s := \mathbb{E}\,\|\widehat{p}_s - p_s\|_2^2 = \sum_{s'} q_{s'}(1 - q_{s'})/N_s = (1 - \sum_{s'} q_{s'}^2)/N_s$. Since successor draws at distinct states are independent

given the counts, $V_{\mathrm{nbr}} := \mathbb{E}\,\|\bar{p}_s - \mathbb{E}\bar{p}_s\|_2^2 = \sum_k w_{sk}^2 V_k$. The error of the blend splits into three terms,

$$p_s^{\mathrm{FD}} - p_s = (1-\theta_s)\big(\widehat{p}_s - p_s\big) + \theta_s\big(\bar{p}_s - \mathbb{E}\bar{p}_s\big) + \theta_s\, b.$$

The first two have mean zero and b is deterministic, so the two cross terms involving b vanish in expectation. The remaining cross term vanishes because $\widehat{p}_s$ is built from the draws at s and $\bar{p}_s$ from the draws at its neighbors, which are independent given the counts. Hence

$$\mathbb{E}\,\big\|p_s^{\mathrm{FD}} - p_s\big\|_2^2 = (1-\theta_s)^2 V_s + \theta_s^2\big(V_{\mathrm{nbr}} + B^2\big),$$

a quadratic in θ_s with positive leading coefficient. Setting its derivative to zero gives $\theta_s^\star = V_s/(V_s + V_{\mathrm{nbr}} + B^2)$, and substituting back gives $V_s(V_{\mathrm{nbr}} + B^2)/(V_s + V_{\mathrm{nbr}} + B^2) = (1-\theta_s^\star)V_s$. Since $V_{\mathrm{nbr}} + B^2$ does not depend on N_s and $V_s \propto 1/N_s$, $\theta_s^\star$ decreases in N_s.

Inverse-square weights and the effective number of neighbors. With the weights induced by the penalty, $w_{sk} = d_{sk}^{-2}/S_s$ and $S_s := \sum_k d_{sk}^{-2}$, the weighted mean-square radius is $\bar{\epsilon}^{\,2} = m/S_s$, where $m := |\mathcal{N}_\epsilon(s)|$. Define the effective number of neighbors

$$m_{\mathrm{eff}} := \frac{1}{\sum_k w_{sk}^2} = \frac{\big(\sum_k d_{sk}^{-2}\big)^2}{\sum_k d_{sk}^{-4}} \;\le\; m,$$

the inequality by Cauchy–Schwarz, with equality exactly when all neighbors are equidistant from s. Concentrating weight on the nearest neighbors lowers $\bar{\epsilon}$, and so the bias, at the price of a smaller m_{eff}, and so a smaller variance reduction.

The ceiling on the effective sample size (6). Each row's variance carries its own visit count. By the binomial calculation above, $V_k = c_k/N_k$ with $c_k := 1 - \sum_{s'} p_k(s')^2$, one minus the collision probability of row k, and likewise $V_s = c_s/N_s$, so the pooled variance is $V_{\mathrm{nbr}} = \sum_k w_{sk}^2\, c_k/N_k$. For any mixing weight, the variance of the blend is $(1-\theta_s)^2 V_s + \theta_s^2 V_{\mathrm{nbr}}$. It is smallest at $\theta_s = V_s/(V_s + V_{\mathrm{nbr}})$, where it equals $V_s V_{\mathrm{nbr}}/(V_s + V_{\mathrm{nbr}})$, so

$$\frac{N_s^{\mathrm{eff}}}{N_s} = \frac{V_s}{(1-\theta_s)^2 V_s + \theta_s^2 V_{\mathrm{nbr}}} \;\le\; 1 + \frac{V_s}{V_{\mathrm{nbr}}},$$

$$\text{equivalently} \qquad N_s^{\mathrm{eff}} \;\le\; N_s + \Big[\sum_k w_{sk}^2\, \frac{c_k}{c_s}\, \frac{1}{N_k}\Big]^{-1}.$$

The bracketed term is the number of visits the neighbors are worth to row s. Each neighbor enters through $1/N_k$, weighted by w_{sk}^2, so a neighbor visited more often is worth more, and through c_k/c_s, so a neighbor with a more nearly deterministic row is worth more as well. With unequal counts the ceiling is therefore higher when the neighbors are better sampled than the row itself and lower when they are worse; if every neighbor is visited r times as often as s and the collision probabilities are equal, it is $1 + r\, m_{\mathrm{eff}}$.

Under equal visitation, $N_k = N_s$ and $c_k = c_s$ for every neighbor, the bracket is $N_s/\sum_k w_{sk}^2 = m_{\mathrm{eff}} N_s$, so $V_{\mathrm{nbr}} = V_s/m_{\mathrm{eff}}$ and $N_s^{\mathrm{eff}} \le (1 + m_{\mathrm{eff}})N_s$: at best the row behaves as though it had been visited $(1 + m_{\mathrm{eff}})N_s$ times, its own visits plus those its neighbors contribute. For the ± 4 neighbor graph of `var-gaus` with inverse-square weights, $m_{\mathrm{eff}} = 3.76$ and this ceiling is 4.76.

The same computation applies to each successor separately, with c replaced by $p(s')(1-p(s'))$; this is the per-cell form measured in Section 3. Two further points bound its use. N_s^{eff} is a variance ratio and ignores bias, and the mean-squared-error-optimal $\theta_s^\star$ of (3) is smaller than the variance-optimal weight used here, so the realized ratio sits below the ceiling. And the measured ratio is for the jointly penalized estimator, whose neighboring rows are themselves smoothed, so the ceiling is a reference level for it rather than a strict bound.

Rate of the mixing weight. Suppose every visit count grows in proportion to the amount of data n, $N_s = r_s n$ and $N_k = r_k n$. Then $V_s = c_s/(r_s n)$, $V_{\mathrm{nbr}} = v/n$ with $v := \sum_k w_{sk}^2 c_k/r_k$, and B does not depend on n, so

$$\theta_s^\star(n) = \frac{c_s/r_s}{c_s/r_s + v + B^2 n}.$$

It is nearly constant while $n \ll (c_s/r_s + v)/B^2$, where variance dominates, and falls as $1/n$ once bias dominates. Theorem 4 reaches the same rate from the graph-ridge form of the penalized estimator: the mean-squared-error-optimal effective strength is $\tilde{\lambda}_n^\star \propto \sigma_n^2 \propto 1/n$. How that rate is realized depends on the likelihood. The tabular negative log-likelihood is a sum over transitions, so its curvature a grows in proportion to N_s and a constant λ in (5) already gives $\theta_s \propto 1/N_s$; a coefficient that itself decays makes the mixing weight fall faster. A latent world model is trained on a minibatch average, whose curvature does not grow with the data, so for small λ the mixing weight is proportional to λ and the rate must come from the coefficient: the analogue of $\theta_s^\star$ is $\lambda \propto 1/n$.

Convexity and uniqueness. The negative log-likelihood is convex in P because $-\log$ is convex; (4) is a sum of squares of linear forms and hence a positive semidefinite quadratic; and the product of simplices is convex, so the penalized problem is convex. For uniqueness, fix a and connect s and k when $k \in \mathcal{N}_\epsilon(s)$. If some connected component of this graph contains no visited state, the objective on that component is the penalty alone, which any set of rows equal across the component minimizes, so the minimizer is not unique. Suppose instead that every component contains a visited state and $\lambda > 0$, and let P and P' be two minimizers. By convexity the objective is constant on the segment between them, and a sum of two convex functions that is affine on a segment is the sum of two functions each affine on it. The penalty is then affine along $v := P' - P$, so v lies in the null space of the neighbor-graph Laplacian and is constant across the rows of each component. The negative log-likelihood is affine along v only if $v(s') = 0$ for every successor s' observed from some state of the component, since $-N \log x$ is strictly convex for $N > 0$. A successor never observed within a component carries zero probability at every minimizer: moving that mass to an observed successor in every row of the component leaves the penalty unchanged and strictly lowers the negative log-likelihood. So v vanishes there too, and $P = P'$. The experiments compute the minimizer by ADMM with 200 iterations (Appendix G).

The induced estimator (5). Hold every row but P_s fixed. Because $d_{sk} = d_{ks}$, the neighbor relation is symmetric and row s appears in (4) both as a center and as a neighbor of each k, so the penalty contributes $2\lambda \sum_k \|P_s - P_k\|_2^2 / d_{sk}^2$. With the isotropic quadratic in place of the negative log-likelihood, the row objective is

$$\tfrac{a}{2}\,\|P_s - \widehat{p}_s\|_2^2 + 2\lambda \sum_k \frac{\|P_s - P_k\|_2^2}{d_{sk}^2}, \qquad \text{with gradient} \quad a(P_s - \widehat{p}_s) + 4\lambda \sum_k \frac{P_s - P_k}{d_{sk}^2}.$$

Setting the gradient to zero gives $(a + 4\lambda S_s)P_s = a\widehat{p}_s + 4\lambda \sum_k d_{sk}^{-2} P_k$, which is (5). The right-hand side is a convex combination of points of the simplex, so $P_s^\star$ lies on the simplex and the constraint is inactive. The numerical constant multiplying λS_s depends on the pair-counting convention and is absorbed into λ. The derivation uses one coefficient for all rows, the global form that the latent world model also uses (Section 4.3). With a per-state coefficient, as in one arm of Section 3, the pair (s, k) is weighted by $(\lambda(s) + \lambda(k))/2$, which keeps the penalty symmetric in s and k, and the same steps give (5) with $4\lambda S_s$ replaced by $2\sum_k (\lambda(s) + \lambda(k)) d_{sk}^{-2}$ and neighbor weights proportional to $(\lambda(s) + \lambda(k)) d_{sk}^{-2}$. Two qualifications apply. First, the exact curvature of $-\sum_{s'} N(s, a, s') \log P_s(s')$ at $\widehat{p}_s$ is diagonal, with entry $N_s/\widehat{p}_s(s')$ for each observed successor and zero for successors not observed at s; the scalar $a \propto N_s$ is its isotropic approximation, and it is what makes a single mixing weight sufficient. With the diagonal curvature, stationarity on the simplex with a multiplier μ gives, for each successor,

$$P_s(s') = \big(1 - \theta_s(s')\big)\widehat{p}_s(s') + \theta_s(s')\,\bar{P}_s(s') - \frac{\theta_s(s')\,\mu}{4\lambda S_s}, \qquad \theta_s(s') = \frac{4\lambda S_s}{N_s/\widehat{p}_s(s') + 4\lambda S_s},$$

where $\bar{P}_s := \sum_k (d_{sk}^{-2}/S_s) P_k$, $\theta_s(s') = 1$ for unobserved successors, μ is fixed by normalization, and the non-negativity constraints are imposed where they bind. The scalar θ_s becomes one mixing weight per successor, each increasing in λ and, at fixed $\widehat{p}_s$, decreasing in N_s. Second, the induced estimator (5) solves for one row with its neighbors held fixed, whereas the minimizer couples all rows, so the induced estimator is one coordinate-descent step rather than the joint solution. Neither qualification affects the three conclusions in the main text, which depend only on signs and monotonicities. As $\lambda \to \infty$ the mixing weights tend to one and the transition law becomes constant across each component, so λ exchanges fidelity for smoothing from the outset.

C.2 THE CONTINUOUS LIMIT

Finite differences converge to directional derivatives. Let $f : \mathbb{R}^d \to \mathbb{R}^m$ be Fréchet differentiable at u, so that $f(u+w) = f(u) + J_f(u)w + r(w)$ with $\|r(w)\|_2 / \|w\|_2 \to 0$ as $w \to 0$. For a unit vector v and $h > 0$, $(f(u+hv) - f(u))/h = J_f(u)v + r(hv)/h$, and $\|r(hv)\|_2 / h \to 0$ as $h \to 0$. By continuity of the norm, $\|(f(u+hv) - f(u))/h\|_2^2 \to \|J_f(u)v\|_2^2$. Because Fréchet differentiability makes $\sup_{\|v\|_2=1} \|r(hv)\|_2 / h \to 0$, the convergence is uniform in v, so it survives averaging over directions.

The isotropic average is Frobenius energy. Let v be uniform on the unit sphere $\mathbb{S}^{d-1}$ and $J := J_f(u)$. The distribution of v is invariant under orthogonal maps, so $\mathbb{E}[vv^\top] = cI_d$, and taking traces gives $c = 1/d$. Then $\mathbb{E}\,\|Jv\|_2^2 = \mathbb{E}\,\mathrm{tr}(J^\top Jvv^\top) = \mathrm{tr}(J^\top J\,\mathbb{E}[vv^\top]) = \frac{1}{d}\,\|J\|_F^2$. The isotropic ball is an idealization. When the data lies near a lower-dimensional manifold, penalties built from data neighborhoods converge instead to the gradient along the manifold (Belkin & Niyogi, 2008), so the full Frobenius norm also penalizes sensitivity in directions the data does not explore.

C.3 THE SOFTMAX FACTOR

For logits $\ell \in \mathbb{R}^C$ and $\pi = \mathrm{softmax}(\ell)$, the Jacobian is $\Sigma := \mathrm{diag}(\pi) - \pi\pi^\top$, the covariance matrix of a one-hot vector drawn from π; it is symmetric and positive semidefinite. With the uniform mixing of Appendix B, $\Pi_\phi = (1-\alpha)\pi + \alpha/C$ and the Jacobian with respect to the logits is $(1-\alpha)\Sigma$, with $\alpha = 0.01$. Row i of Σ has diagonal entry $\pi_i(1-\pi_i)$ and off-diagonal entries of total magnitude $\sum_{j\neq i} \pi_i\pi_j = \pi_i(1-\pi_i)$. By the Gershgorin circle theorem every eigenvalue of Σ lies in a disc of center and radius $\pi_i(1-\pi_i)$ for some i, so

$$\|\Sigma\|_2 \;\le\; \max_i 2\pi_i(1-\pi_i) \;\le\; \tfrac{1}{2},$$

with equality for two classes at $\pi = (\frac{1}{2}, \frac{1}{2})$. Consequently $\left\|J_{\Pi_\phi}\right\|_F \le (1-\alpha)\,\|\Sigma\|_2\,\|J_\ell\|_F \le \frac{1}{2}\,\|J_\ell\|_F$, where J_ℓ is the Jacobian of the logits: a logit-space penalty bounds probability-space sensitivity. The bound is not uniform, since $\max_i \pi_i(1-\pi_i) \to 0$ as the row approaches a one-hot distribution.

D BIAS–VARIANCE TRADE-OFF

Fix an action a and destination s'; we estimate the column $\theta^\star = (P(s' \mid s,a))_{s\in\mathcal{S}} \in \mathbb{R}^N$ ($N = |\mathcal{S}|$) of the transition law from counts $N(s,a,s')$ with per-state visit totals n_s and $T = \sum_s n_s$. This is the column slice, at fixed (a,s'), of the joint matrix program: the multinomial likelihood and the simplex constraint act *row*-wise (over destinations, per source s) and are treated jointly in Theorem 1, while the graph penalty acts *column*-wise (across source states). The column analysis below proceeds through the Fisher surrogate of Lemma 1, which decouples the columns. We write $\hat{P}_\lambda$ for the full transition matrix whose columns are the column estimates $\hat{\theta}_\lambda$.

Let $\mathcal{V} := \{s : n_s > 0\}$ be the visited states, $\mathcal{U} := \mathcal{S} \setminus \mathcal{V}$ the unvisited ones, and $N_\mathcal{V} := |\mathcal{V}|$. Conditioning on the visit counts $\{n_s\}$ (adaptively determined, hence random), successor draws from distinct source states are independent, so the empirical frequency vector $y := (N(s,a,s')/n_s)_{s\in\mathcal{V}}$ satisfies $\mathbb{E}[y \mid \{n_s\}] = \theta^\star_\mathcal{V}$ and $\mathrm{Cov}(y \mid \{n_s\}) = \Sigma := \mathrm{diag}(v_s/n_s)_{s\in\mathcal{V}}$, $v_s := P(s' \mid s,a)(1 - P(s' \mid s,a))$; marginal unbiasedness follows by iterated expectation. This Σ is diagonal because it is the noise *across source states at a fixed destination*; the within-row multinomial correlation (across destinations) is the separate object of Remark 2.

Unvisited states. Let $L_0 = D - W \succeq 0$ be the (symmetric) finite-difference Dirichlet Laplacian on $\mathcal{S}$, so GPLD's penalty is the Dirichlet energy $\theta^\top L_0 \theta$, and partition it into blocks $L_{\mathcal{V}\mathcal{V}}, L_{\mathcal{V}\mathcal{U}}, L_{\mathcal{U}\mathcal{V}}, L_{\mathcal{U}\mathcal{U}}$. We assume every connected component of the neighbor graph contains a visited state; then $L_{\mathcal{U}\mathcal{U}} \succ 0$. For fixed $\theta_\mathcal{V}$, the penalty is minimized over $\theta_\mathcal{U}$ by the harmonic extension $\theta_\mathcal{U} = -L_{\mathcal{U}\mathcal{U}}^{-1} L_{\mathcal{U}\mathcal{V}} \theta_\mathcal{V}$, with minimum value $\theta_\mathcal{V}^\top L_\mathcal{V} \theta_\mathcal{V}$, where

$$L_\mathcal{V} := L_{\mathcal{V}\mathcal{V}} - L_{\mathcal{V}\mathcal{U}} L_{\mathcal{U}\mathcal{U}}^{-1} L_{\mathcal{U}\mathcal{V}} \succeq 0$$

is the Schur complement (Kron reduction) of L_0, itself a graph Laplacian on $\mathcal{V}$; $L_\mathcal{V} = L_0$ when every state is visited. Unvisited states carry no data, so they enter the column analysis only through $L_\mathcal{V}$, and their estimates are set by their visited neighbors. The solver of Appendix F works with L_0 on all of $\mathcal{S}$ and needs no such reduction.

The μ-weighted geometry. Let $\mu_s := n_s/T$ be the visitation on $\mathcal{V}$, $D_\mu := \mathrm{diag}(\mu_s)_{s\in\mathcal{V}} \succ 0$, and $\langle f, g\rangle_\mu := f^\top D_\mu g$, $\|f\|_\mu^2 := \langle f, f\rangle_\mu$ for $f, g \in \mathbb{R}^\mathcal{V}$. Set $\mathcal{L} := D_\mu^{-1} L_\mathcal{V}$, the associated *$\mu$-self-adjoint* operator ($\langle f, \mathcal{L}g\rangle_\mu = f^\top L_\mathcal{V} g = \langle \mathcal{L}f, g\rangle_\mu$). Its eigenpairs $\mathcal{L}v_j = \omega_j v_j$, $0 = \omega_1 \le \cdots \le \omega_{N_\mathcal{V}}$, are μ-orthonormal, $\langle v_i, v_j\rangle_\mu = \delta_{ij}$, and $S_\lambda = (I + \lambda\mathcal{L})^{-1}$ is μ-self-adjoint. The μ-norm gives unvisited states zero weight, so the statements below measure error on $\mathcal{V}$ only.

Assumption 2 (Homoscedastic column noise)**.** $v_s \equiv v$ for all $s \in \mathcal{V}$.

Under Assumption 2 the noise is *white in this metric*: since $n_s = T\mu_s$,

$$\Sigma = \mathrm{diag}\big(v/(T\mu_s)\big) = \sigma^2 D_\mu^{-1}, \qquad \sigma^2 := v/T,$$

so Σ is isotropic in $\langle\cdot,\cdot\rangle_\mu$. In general v_s varies with s; that case is covered by Theorem 6. All norms, traces, and eigen-expansions below are μ-weighted, with $\alpha_j := \langle v_j, \theta^\star_\mathcal{V}\rangle_\mu$. The penalized estimator is

$$\hat{\theta}_\lambda = \arg\min_\theta \; \mathcal{L}_{\mathrm{MLE}}(\theta) + \lambda\, \theta^\top L_0 \theta, \tag{9}$$

where $\mathcal{L}_{\mathrm{MLE}}$ is the multinomial negative log-likelihood of the joint row problem.

Lemma 1 (Graph-ridge form)**.** *Replacing $\mathcal{L}_{\mathrm{MLE}}$ by its quadratic (Fisher) surrogate $\frac{1}{2}(\theta_\mathcal{V} - y)^\top W_F(\theta_\mathcal{V} - y)$ with $W_F := \Sigma^{-1} = \sigma^{-2} D_\mu$, the column estimator becomes the linear smoother*

$$\hat{\theta}_{\lambda,\mathcal{V}} = S_\lambda\, y, \qquad S_\lambda := (W_F + 2\lambda L_\mathcal{V})^{-1} W_F = (I + 2\lambda\sigma^2\mathcal{L})^{-1}, \tag{10}$$

extended to unvisited states by $\hat{\theta}_{\lambda,\mathcal{U}} = -L_{\mathcal{U}\mathcal{U}}^{-1} L_{\mathcal{U}\mathcal{V}} \hat{\theta}_{\lambda,\mathcal{V}}$. S_λ is μ-self-adjoint, with effective penalty $\tilde{\lambda} = 2\lambda\sigma^2$; we write $S_\lambda = (I + \lambda\mathcal{L})^{-1}$ below (absorbing the constant into λ).

Proof. The quadratic surrogate is the second-order (LAN/Fisher) expansion of $\mathcal{L}_{\mathrm{MLE}}$ about y; it need not be exact, as Theorem 1 bounds the resulting error. The surrogate does not involve $\theta_\mathcal{U}$, so minimizing over $\theta_\mathcal{U}$ first gives the harmonic extension and the reduced penalty $\lambda\theta_\mathcal{V}^\top L_\mathcal{V}\theta_\mathcal{V}$. Strict convexity ($W_F \succ 0$, $L_\mathcal{V} \succeq 0$) and stationarity $W_F(\theta_\mathcal{V} - y) + 2\lambda L_\mathcal{V}\theta_\mathcal{V} = 0$ give $(W_F + 2\lambda L_\mathcal{V})\theta_\mathcal{V} = W_F y$, i.e. (10); substituting $W_F = \sigma^{-2} D_\mu$ yields $(D_\mu + 2\lambda\sigma^2 L_\mathcal{V})^{-1} D_\mu = (I + 2\lambda\sigma^2\mathcal{L})^{-1}$. □

We analyze $S_\lambda = (I + \lambda\mathcal{L})^{-1}$ acting on $y = \theta^\star_{\mathcal{V}} + \xi$, $\mathbb{E}\xi = 0$, $\mathrm{Cov}(\xi) = \sigma^2 D_\mu^{-1}$ ($\langle\cdot,\cdot\rangle_\mu$-white). Lemma 1 is exact for the surrogate; the next theorem controls the gap to the *true* penalized multinomial MLE on the observed support, with no $n \to \infty$ limit.

Theorem 1 (Self-concordant finite-sample exactness)**.** *Assume every source state is visited ($n_s \geq 1$ for all s). Let $\mathcal{O}_s = \{s' : N(s,a,s') > 0\}$ be the observed support of row s; the MLE $\hat{p}$ lies in the* relative interior of the observed-support face $\prod_s \Delta(\mathcal{O}_s)$ *(each observed destination has positive frequency). Let $H := \nabla^2 \mathcal{L}_{\mathrm{MLE}}(\hat{p})$ be the multinomial Fisher information on that face, $\hat{P}^{\mathcal{O}}_\lambda$ the minimizer of the joint penalized problem (11) on that face, $\hat{\theta}^c_\lambda$ the face-constrained Newton step from $\hat{p}$ for (9), and $\delta := \|2\lambda L_0 \hat{p}\|_{H^{-1},\mathcal{T}}$ the constrained Newton decrement ($\mathcal{T}$ = tangent of the face). If $\delta < \frac{1}{4}$ then*

$$\left\| \hat{P}^{\mathcal{O}}_\lambda - \hat{\theta}^c_\lambda \right\|_H \;\leq\; c\,\delta^2 \qquad (c \text{ an absolute constant}),$$

so every bias–variance statement for the smoother holds for $\hat{P}^{\mathcal{O}}_\lambda$ up to $O(\delta^2)$, where $\delta^2 = O(\lambda^2 \|L_0 \hat{p}\|^2_{H^{-1}})$.

Proof. Object. $\hat{p}, \hat{P}^{\mathcal{O}}_\lambda$ are the full transition matrices restricted to the observed-support face; H is block-diagonal across rows, each block a row's multinomial Fisher $\mathrm{diag}(n_s/\hat{p}_{s,s'})_{s' \in \mathcal{O}_s}$; the penalty $\mathrm{Tr}(\Theta^\top L_0 \Theta)$ couples rows; and $\mathcal{T} = \prod_s \{u_s : \mathbf{1}^\top u_s = 0\}$. The likelihood decouples over rows, the penalty over columns; $\hat{P}^{\mathcal{O}}_\lambda$ minimizes the joint problem. We use the affine-invariant theory of self-concordant functions (Nesterov, 2018, §5): F is standard self-concordant if $|D^3 F(x)[h,h,h]| \leq 2(D^2 F(x)[h,h])^{3/2}$.

Step 1: $F := \mathcal{L}_{\mathrm{MLE}} + \lambda(\cdot)^\top L_0 (\cdot)$ is self-concordant. Per row, $\mathcal{L}_{\mathrm{MLE}}(\theta) = -\sum_{s' \in \mathcal{O}_s} c_{s'} \log \theta_{s'} + \mathrm{const}$ with *integer* counts $c_{s'} = N(s,a,s') \geq 1$ on the observed support (zero-count terms are absent). Each $-\log \theta_{s'}$ is standard self-concordant; since $a\,f$ is $M_f/\sqrt{a}$-self-concordant and $c_{s'} \geq 1$, scaling by $c_{s'}$ does *not* increase the parameter, and a sum of self-concordant functions is self-concordant, so $\mathcal{L}_{\mathrm{MLE}}$ is standard self-concordant on the (relative-interior) face. The penalty $q(\theta) = \lambda \theta^\top L_0 \theta$ is a convex quadratic, $D^3 q \equiv 0$, so $|D^3 F[h,h,h]| = |D^3 \mathcal{L}_{\mathrm{MLE}}[h,h,h]| \leq 2(D^2 \mathcal{L}_{\mathrm{MLE}}[h,h])^{3/2} \leq 2(D^2 F[h,h])^{3/2}$. Restriction to the affine face preserves self-concordance (affine invariance), so the constrained problem is self-concordant with $(\nabla F)_{\mathcal{T}}, (\nabla^2 F)_{\mathcal{T}}$.

Step 2: the Newton decrement at $\hat{p}$ is δ. Since $\hat{p}$ is the constrained MLE, $(\nabla \mathcal{L}_{\mathrm{MLE}}(\hat{p}))_{\mathcal{T}} = 0$, so $(\nabla F(\hat{p}))_{\mathcal{T}} = (2\lambda L_0 \hat{p})_{\mathcal{T}}$ and $\delta(\hat{p}) = \|2\lambda L_0 \hat{p}\|_{H^{-1},\mathcal{T}} = \delta$, with $H = \mathrm{diag}(c_{s'}/\hat{p}^2_{s'}) = \mathrm{diag}(n_s/\hat{p}_{s'})$.

Step 3: existence, then one Newton step. By strict convexity of F on the face (the multinomial Hessian $\mathrm{diag}(c_{s'}/\theta^2_{s'})$ is PD on the tangent for $c_{s'} \geq 1$, and the penalty is convex), the constrained minimizer $\hat{P}^{\mathcal{O}}_\lambda$ *exists and is unique*, characterized by $(\nabla F(\hat{P}^{\mathcal{O}}_\lambda))_{\mathcal{T}} = 0$. The self-concordant local-convergence theorem (Nesterov, 2018, Thm. 5.2.2) then bounds the single Newton step: if $\delta(\hat{p}) < \frac{1}{4}$, then $\delta(\hat{\theta}^c_\lambda) \leq (\delta/(1-\delta))^2$, and since the local distance is controlled by the decrement, $\left\|\hat{\theta}^c_\lambda - \hat{P}^{\mathcal{O}}_\lambda\right\|_H \leq \delta(\hat{\theta}^c_\lambda)/(1-\delta(\hat{\theta}^c_\lambda))$; composing gives $\left\|\hat{P}^{\mathcal{O}}_\lambda - \hat{\theta}^c_\lambda\right\|_H \leq c\,\delta^2$. Finally $\delta = O(\lambda)$, so the remainder is $O(\lambda^2)$, largest in the sparse-data (H small), strong-smoothing regime. □

Scope of Theorem 1. The theorem covers rows whose penalized estimate stays on the observed support. The estimator $\hat{P}_\lambda$ of (11), which Appendix F computes, is the minimizer on the full product of simplices, and it equals $\hat{P}^{\mathcal{O}}_\lambda$ only when it stays on the face. In the sparse regime the penalty moves mass to successors observed only at neighboring states, which is how it shares information; there the estimator leaves the face, and the smoother formulas below describe it without a certified remainder.

Remark 1 (The remainder is a caveat, not the source of gains)**.** The $O(\delta^2)$ term is largest in the sparse-data regime, so the surrogate is least accurate there. It is not the source of gains: those come from Theorem 4, where the variance $\sim \sigma^2$ is largest when data is scarce.

The following are the standard bias–variance/shrinkage facts of graph-ridge (Laplacian-penalized) estimation (Smola & Kondor, 2003; Belkin et al., 2006); the existence of a strictly beneficial penalty (Theorem 3(2)) is the graph analogue of the Hoerl–Kennard ridge result (Hoerl & Kennard, 1970).

We state them for completeness and because the value-target and planning results build on them, not as new contributions. Recall $\mathcal{L}v_j = \omega_j v_j$ with $\{v_j\}$ μ-orthonormal and $\alpha_j = \langle v_j, \theta^\star_{\mathcal{V}}\rangle_\mu$.

Theorem 2 (Exact bias, variance, MSE)**.** *Under Assumption 2, for $\hat\theta_\lambda = S_\lambda y$, $S_\lambda = (I+\lambda\mathcal{L})^{-1}$, in the μ-metric,*

$$\mathrm{Bias}(\hat\theta_\lambda) = -\lambda(I+\lambda\mathcal{L})^{-1}\mathcal{L}\theta^\star_{\mathcal{V}}, \qquad \|\mathrm{Bias}\|_\mu^2 = \sum_j \left(\tfrac{\lambda\omega_j}{1+\lambda\omega_j}\right)^2 \alpha_j^2,$$

$$\mathbb{E}\,\|S_\lambda\xi\|_\mu^2 = \sigma^2 \sum_j (1+\lambda\omega_j)^{-2},$$

so $\mathrm{MSE}_\mu(\lambda) = \mathbb{E}\left\|\hat\theta_\lambda - \theta^\star_{\mathcal{V}}\right\|_\mu^2 = \sum_{j=1}^{N_{\mathcal{V}}} \left[\left(\tfrac{\lambda\omega_j}{1+\lambda\omega_j}\right)^2\alpha_j^2 + \tfrac{\sigma^2}{(1+\lambda\omega_j)^2}\right]$.

Proof. $\mathrm{Bias} = (S_\lambda - I)\theta^\star_{\mathcal{V}} = -\lambda(I+\lambda\mathcal{L})^{-1}\mathcal{L}\theta^\star_{\mathcal{V}}$; expanding $\theta^\star_{\mathcal{V}} = \sum_j \alpha_j v_j$ in the μ-orthonormal basis gives $\|\mathrm{Bias}\|_\mu^2$. For the variance, S_λ is μ-self-adjoint and $\mathrm{Cov}(\xi) = \sigma^2 D_\mu^{-1}$, so $\mathbb{E}\,\|S_\lambda\xi\|_\mu^2 = \mathbb{E}[\xi^\top S_\lambda^\top D_\mu S_\lambda \xi] = \sigma^2\,\mathrm{Tr}(S_\lambda^\top D_\mu S_\lambda D_\mu^{-1}) = \sigma^2\,\mathrm{Tr}(S_\lambda^2) = \sigma^2\sum_j(1+\lambda\omega_j)^{-2}$, using $S_\lambda^\top = D_\mu S_\lambda D_\mu^{-1}$ and cyclicity. Adding gives the MSE. □

As $\lambda \to 0$: zero bias, variance $N_{\mathcal{V}}\sigma^2$ (the raw MLE). As $\lambda\to\infty$: S_λ projects onto $\ker\mathcal{L}$, variance $\to \sigma^2\cdot(\#\text{components})$, bias $\to \sum_{\omega_j>0}\alpha_j^2$.

Theorem 3 (standard; Hoerl–Kennard for (2))**.** *Assume $\sigma^2 > 0$ and $L_{\mathcal{V}} \neq 0$. Then:*

1. *$\|\mathrm{Bias}(\lambda)\|_\mu^2$ is nondecreasing and the variance $\sigma^2\sum_j(1+\lambda\omega_j)^{-2}$ nonincreasing on $[0,\infty)$, strictly on modes with $\omega_j > 0$ (and $\alpha_j \neq 0$ for the bias).*
2. *$\frac{d}{d\lambda}\mathrm{MSE}_\mu(0^+) = -2\sigma^2\,\mathrm{Tr}(\mathcal{L}) < 0$; hence the MSE-optimal $\lambda^\star > 0$ for* every *$\theta^\star$ (Hoerl & Kennard, 1970).*
3. *With $t_j := (1+\lambda\omega_j)^{-1}$, the mode-$j$ risk is minimized at $t_j^\star = \alpha_j^2/(\alpha_j^2+\sigma^2)$, value $\alpha_j^2\sigma^2/(\alpha_j^2+\sigma^2) < \sigma^2$.*

Proof. *(1)* $b_j'(\lambda) = 2\alpha_j^2\omega_j(\lambda\omega_j)/(1+\lambda\omega_j)^3 \geq 0$ and $c_j'(\lambda) = -2\sigma^2\omega_j/(1+\lambda\omega_j)^3 \leq 0$, strict for $\omega_j > 0$. *(2)* $b_j'(0) = 0$, $c_j'(0) = -2\sigma^2\omega_j$, so the derivative at 0^+ is $-2\sigma^2\sum_j\omega_j = -2\sigma^2\,\mathrm{Tr}(\mathcal{L}) < 0$; continuity and the finite $\lambda\to\infty$ limit force $\lambda^\star > 0$. *(3)* $f_j(t_j) = (1-t_j)^2\alpha_j^2 + \sigma^2 t_j^2$ is strictly convex with the stated minimizer. □

Let $\sigma^2 = \sigma_T^2 = v/T \to 0$ as $T\to\infty$; allow $\lambda = \lambda_T$.

Theorem 4 (standard consistency/rate)**.** *With the graph and the visited set $\mathcal{V}$ fixed:*

1. *$\|\mathrm{Bias}(\lambda)\|_\mu^2 \leq \lambda^2\|\mathcal{L}\theta^\star_{\mathcal{V}}\|_\mu^2$ and the variance $\leq N_{\mathcal{V}}\sigma_T^2$, so $\mathrm{MSE}_\mu(\lambda) \leq \lambda^2\|\mathcal{L}\theta^\star_{\mathcal{V}}\|_\mu^2 + N_{\mathcal{V}}\sigma_T^2$.*
2. *$\lambda_T\to 0$, $\sigma_T^2\to 0 \Rightarrow \mathrm{MSE}_\mu(\lambda_T)\to 0$.*
3. *$\mathrm{MSE}_\mu(\lambda) = N_{\mathcal{V}}\sigma_T^2 - 2\lambda\sigma_T^2\,\mathrm{Tr}(\mathcal{L}) + \lambda^2\|\mathcal{L}\theta^\star_{\mathcal{V}}\|_\mu^2 + o(\lambda^2)$,[1] minimized at $\lambda_T^\star = \sigma_T^2\,\mathrm{Tr}(\mathcal{L})/\|\mathcal{L}\theta^\star_{\mathcal{V}}\|_\mu^2 \asymp \sigma_T^2$, $\mathrm{MSE}_\mu(\lambda_T^\star) = N_{\mathcal{V}}\sigma_T^2 - \sigma_T^4\,\mathrm{Tr}(\mathcal{L})^2/\|\mathcal{L}\theta^\star_{\mathcal{V}}\|_\mu^2$, strictly below the MLE risk by $O(\sigma_T^4)$.*

Proof. *(1)* $(\lambda\omega_j/(1+\lambda\omega_j))^2 \leq (\lambda\omega_j)^2$ and $(1+\lambda\omega_j)^{-2}\leq 1$. *(2)* both bounds vanish. *(3)* expand Theorem 2 to second order, differentiate, substitute. □

[1] The λ^2 coefficient also carries $+3\sigma_T^2\sum_j\omega_j^2$, $O(\sigma_T^2)$ smaller and immaterial to $\lambda_T^\star$ and the rate.

General (non-diagonal) noise covariance. Theorem 2 took the μ-white noise $\Sigma = \sigma^2 D_\mu^{-1}$. We now drop Assumption 2 and allow an arbitrary symmetric PSD Σ *on the source axis* (e.g. heteroscedastic v_s, or correlated source sampling). Two smoothers: the *plain* $S_\lambda = (I + \lambda\mathcal{L})^{-1}$ and the *Fisher-weighted* $S_\lambda = (\Sigma^{-1} + 2\lambda L_\mathcal{V})^{-1}\Sigma^{-1}$.

Theorem 5 (Bias and covariance, arbitrary source-axis Σ)**.** *For any fixed linear smoother S_λ and any source-axis Σ, $\mathrm{Bias}(\hat\theta_\lambda) = (S_\lambda - I)\theta^\star_\mathcal{V}$ and $\mathrm{Cov}(\hat\theta_\lambda) = S_\lambda \Sigma S_\lambda^\top$. For the plain smoother the μ-variance is $\mathrm{Tr}(D_\mu S_\lambda \Sigma S_\lambda^\top)$; in the μ-orthonormal basis only the diagonal $\langle v_j, \Sigma v_j\rangle$-type terms survive the trace, giving $\mathrm{MSE}_\mu(\lambda) = \sum_j [(\frac{\lambda\omega_j}{1+\lambda\omega_j})^2 \alpha_j^2 + \widetilde\sigma_j^2 (1+\lambda\omega_j)^{-2}]$ with $\widetilde\sigma_j^2$ the j-th diagonal of Σ in that basis.*

Proof. Bias depends on $\mathbb{E}\xi = 0$ only; $\mathrm{Cov}(\hat\theta_\lambda) = S_\lambda \Sigma S_\lambda^\top$ since ξ is the only randomness. The μ-trace picks the basis diagonal by the same cyclicity computation as Theorem 2. □

Theorem 6 (Whitening reduction, source axis)**.** *For $S_\lambda = (\Sigma^{-1} + 2\lambda L_\mathcal{V})^{-1}\Sigma^{-1}$ with $\Sigma \succ 0$, and the noise-adapted Laplacian $\widetilde L := \Sigma^{1/2} L_\mathcal{V} \Sigma^{1/2}$ (eigenpairs $\widetilde\omega_j, \widetilde v_j$), the whitened error $\eta := \Sigma^{-1/2}(\hat\theta_\lambda - \theta^\star_\mathcal{V})$ has*

$$\mathbb{E}\left\|\hat\theta_\lambda - \theta^\star_\mathcal{V}\right\|^2_{\Sigma^{-1}} = \sum_j \left[\left(\tfrac{2\lambda\widetilde\omega_j}{1+2\lambda\widetilde\omega_j}\right)^2 \widetilde\alpha_j^2 + (1+2\lambda\widetilde\omega_j)^{-2}\right], \quad \widetilde\alpha_j := \widetilde v_j^\top \Sigma^{-1/2}\theta^\star_\mathcal{V},$$

i.e. Theorem 2 with $(\omega_j, \alpha_j, \sigma^2) \mapsto (\widetilde\omega_j, \widetilde\alpha_j, 1)$ and penalty 2λ; Theorems 3–4 transfer verbatim. Both Σ and $L_\mathcal{V}$ act on the same *(source) axis, so $\Sigma^{1/2} L_\mathcal{V} \Sigma^{1/2}$ is well defined.*

Proof. With $M := \Sigma^{-1} + 2\lambda L_\mathcal{V}$, $S_\lambda = M^{-1}\Sigma^{-1}$ and $\mathrm{Cov} = M^{-1}\Sigma^{-1}M^{-1}$; $\Sigma^{1/2} M \Sigma^{1/2} = I + 2\lambda\widetilde L$ gives $\mathrm{Cov} = \Sigma^{1/2}(I + 2\lambda\widetilde L)^{-2}\Sigma^{1/2}$, so $\mathrm{Cov}(\eta) = (I + 2\lambda\widetilde L)^{-2}$; the bias in η is $-2\lambda(I + 2\lambda\widetilde L)^{-1}\widetilde L \Sigma^{-1/2}\theta^\star_\mathcal{V}$. Expanding in $\{\widetilde v_j\}$ yields the display. □

Remark 2 (Two distinct axes; the row simplex is not composed with the source Laplacian)**.** The within-row multinomial covariance $C_s := \frac{1}{n_s}(\mathrm{diag}(p_s) - p_s p_s^\top)$ is indexed by *destinations* and is singular ($C_s \mathbf{1} = 0$); its null direction is exactly the row simplex constraint, handled on the row tangent by Theorem 1. It lives on a *different* axis from the source-state Laplacian L_0, so the two cannot be composed as $\Sigma^{1/2} L_0 \Sigma^{1/2}$; Theorem 6 is a source-axis statement (its Σ is source-state noise). The joint treatment of source smoothing and within-row multinomial correlation is the vectorized problem on $\mathrm{vec}(\Theta^\top)$ (rows stacked), with penalty matrix $L_0 \otimes I$ and covariance $\mathrm{blkdiag}(C_s)$. These two commute only when all blocks C_s are equal, which fails in general because n_s and p_s vary with s. The joint problem therefore does not separate into the two axes, and the per-axis results above are not its marginals; Theorem 1 is the statement that treats both axes at once.

E PLANNING BOUND

Let P be the true transition (Markov) operator with data distribution μ, $\hat{P}$ the GPLD estimate, $\gamma \in (0,1)$, bounded reward r, policy π; $v, \hat{v}$ solve $v = r + \gamma P v$, $\hat{v} = r + \gamma \hat{P}\hat{v}$. Set $g := Pv$, $\hat{g} := \hat{P}v$, so $(\hat{P} - P)v = \hat{g} - g$; $\|\cdot\|_\mu, \|\cdot\|_\nu$ are $L^2(\mu), L^2(\nu)$.

Assumption 3 (Coverage)**.** $\hat{P}$ is a Markov operator, and the normalized discounted occupancy $\hat{d}^\nu := (1-\gamma)\sum_{k\geq 0}\gamma^k \nu \hat{P}^k$ satisfies $C := \left\| \hat{d}^\nu / \mu \right\|_\infty < \infty$.

Theorem 7 (GPLD planning bound; a simulation-lemma argument)**.** *Under Assumption 3,* $\|\hat{v} - v\|_\nu \leq \frac{\gamma\sqrt{C}}{1-\gamma}\|\hat{g} - g\|_\mu$*, in the style of the simulation lemma and concentrability bounds of Kearns & Singh (2002); Munos & Szepesvári (2008).*

Proof. Step 1. $(I - \gamma\hat{P})(\hat{v} - v) = \gamma(\hat{g} - g)$; since $\hat{P}$ is Markov, $\left\|\hat{P}\right\|_\infty = 1$ and $\gamma\left\|\hat{P}\right\|_\infty = \gamma < 1$, so the Neumann series converges (in sup norm) and $\hat{v} - v = \gamma\sum_{k\geq 0}\gamma^k\hat{P}^k e$, $e := \hat{g} - g$. *Step 2.* With $w_k = (1-\gamma)\gamma^k$, convexity of $\|\cdot\|_\nu^2$ gives $\|\hat{v} - v\|_\nu^2 \leq (\frac{\gamma}{1-\gamma})^2 \sum_k w_k \left\|\hat{P}^k e\right\|_\nu^2$. *Step 3.* $\hat{P}$ averaging $\Rightarrow (\hat{P}^k e)^2 \leq \hat{P}^k(e^2)$ pointwise, so $\sum_k w_k \left\|\hat{P}^k e\right\|_\nu^2 \leq \langle \hat{d}^\nu, e^2\rangle$. *Step 4.* $\langle \hat{d}^\nu, e^2\rangle \leq C\,\mathbb{E}_\mu[e^2] = C\,\|e\|_\mu^2$. □

The value-target error is a GPLD estimation problem: $g(s) = \mathbb{E}[v(x') \mid s]$ is estimated by the smoothed empirical conditional mean $\hat{g}_\lambda = S_\lambda \hat{g}_{\mathrm{MLE}}$, $\hat{g}_{\mathrm{MLE}}(s) = \sum_{s'} \frac{N(s,\cdot,s')}{n_s} v(s')$, unbiased with variance $\mathrm{Var}(v(x') \mid s)/n_s$.

Proposition 1 (Bias–variance for the value-target)**.** *In the μ-metric, with $\beta_j := \langle v_j, g\rangle_\mu$, homoscedastic $\sigma_g^2 := \bar{\sigma}_v^2/T$ (so the noise is again $\sigma_g^2 D_\mu^{-1}$), and* $\mathrm{d_{eff}}(\lambda) := \mathrm{Tr}(S_\lambda^2) = \sum_j (1+\lambda\omega_j)^{-2}$,

$$\mathbb{E}\,\|\hat{g}_\lambda - g\|_\mu^2 = \sum_j \left[\left(\tfrac{\lambda\omega_j}{1+\lambda\omega_j}\right)^2 \beta_j^2 + \tfrac{\sigma_g^2}{(1+\lambda\omega_j)^2} \right] = \|\mathrm{Bias}\|_\mu^2 + \sigma_g^2\,\mathrm{d_{eff}}(\lambda),$$

and at the optimal λ, $\mathbb{E}\,\|\hat{g}_\lambda - g\|_\mu^2 \asymp \sigma_g^2\,\mathrm{d_{eff}}$.

Proof. $\hat{g}_{\mathrm{MLE}} = g + \zeta$, $\mathrm{Cov}(\zeta) = \sigma_g^2 D_\mu^{-1}$, so Theorem 2 applies with truth g and μ-white noise σ_g^2; the optimal-λ order is Theorem 4(3). □

Corollary 1 (Self-contained planning rate)**.** *Under Assumption 3 and smoothness of g, at the GPLD-optimal penalty,*

$$\mathbb{E}\,\|\hat{v} - v\|_\nu \;\leq\; \frac{\gamma\sqrt{C}}{1-\gamma}\sqrt{\mathbb{E}\,\|\hat{g} - g\|_\mu^2} \;\lesssim\; \frac{\gamma\sqrt{C}}{1-\gamma}\sqrt{\sigma_g^2\,\mathrm{d_{eff}}} \;=\; \frac{\gamma\sqrt{C}}{1-\gamma}\sqrt{\frac{\bar{\sigma}_v^2\,\mathrm{d_{eff}}}{T}}.$$

Proof. Take expectations in Theorem 7, apply Jensen, and substitute Proposition 1; $\sigma_g^2 = \bar{\sigma}_v^2/T$ gives the last form (rate $\sqrt{\mathrm{d_{eff}}/T}$). □

F SOLVING THE FINITE-STATE PENALIZED ESTIMATOR EXACTLY (ADMM)

The program and its splitting. Let $\Theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the transition matrix (rows indexed by source s, columns by destination s'), $c_{s,s'} = N(s,a,s')$ the counts, and Δ the probability simplex. The penalized MLE

$$\hat{P}_\lambda = \underset{\Theta:\,\Theta_{s,\cdot}\in\Delta\ \forall s}{\arg\min} \underbrace{-\sum_{s,s'} c_{s,s'} \log \Theta_{s,s'}}_{\mathcal{L}_{\mathrm{MLE}}(\Theta)} + \lambda\ \mathrm{Tr}(\Theta^\top L_0 \Theta) \tag{11}$$

is convex: the likelihood is convex and separable over rows, the penalty $\mathrm{Tr}(\Theta^\top L_0\Theta) = \sum_{s'} \Theta_{\cdot,s'}^\top L_0\, \Theta_{\cdot,s'}$ is a convex quadratic coupling rows through the symmetric PSD source-state Laplacian $L_0 = D - W$, and the feasible set is a product of simplices. Here

$$\mathrm{Tr}(\Theta^\top L_0 \Theta) = \sum_{\{s,k\}} W_{sk}\, \|\Theta_{s,\cdot} - \Theta_{k,\cdot}\|^2$$

sums over unordered neighbor pairs, the same convention as (9). The released code sums over ordered pairs, which counts each pair twice and writes the penalty as $2\lambda'\,\mathrm{Tr}(\Theta^\top L_0\Theta)$; its coefficient λ' corresponds to $\lambda = 2\lambda'$ here. Introduce a copy $Z = \Theta$ and split

$$f(\Theta) = \mathcal{L}_{\mathrm{MLE}}(\Theta) + \sum_s \iota_\Delta(\Theta_{s,\cdot}) \qquad \text{(row-separable)},$$

$$g(Z) = \lambda\ \mathrm{Tr}(Z^\top L_0 Z) \qquad \text{(smooth, couples rows)}.$$

With scaled dual U and penalty $\rho > 0$, the augmented Lagrangian is $L_\rho(\Theta, Z, U) = f(\Theta) + g(Z) + \frac{\rho}{2}\,\|\Theta - Z + U\|_F^2 - \frac{\rho}{2}\,\|U\|_F^2$.

The ADMM iteration

$$\Theta^{k+1} = \underset{\Theta}{\arg\min}\ f(\Theta) + \tfrac{\rho}{2}\left\|\Theta - Z^k + U^k\right\|_F^2 \qquad \text{(data fit; row-separable)} \tag{12}$$

$$Z^{k+1} = \underset{Z}{\arg\min}\ g(Z) + \tfrac{\rho}{2}\left\|\Theta^{k+1} - Z + U^k\right\|_F^2 \qquad \text{(graph smoothing)} \tag{13}$$

$$U^{k+1} = U^k + \Theta^{k+1} - Z^{k+1}. \qquad \text{(dual)}$$

Θ-step: a per-row simplex prox. The Θ-step (12) decouples across rows; row s solves, with $a := Z^k_{s,\cdot} - U^k_{s,\cdot}$,

$$\min_{\theta\in\Delta} -\sum_{s'} c_{s,s'} \log \theta_{s'} + \tfrac{\rho}{2}\,\|\theta - a\|^2 .$$

Its KKT conditions, with multiplier η for $\mathbf{1}^\top\theta = 1$, give the closed form $\theta_{s'}(\eta) = \frac{1}{2\rho}\big[(\rho a_{s'} - \eta) + \sqrt{(\rho a_{s'} - \eta)^2 + 4\rho c_{s,s'}}\big]$, and η is the unique root of the decreasing map $\eta \mapsto \sum_{s'}\theta_{s'}(\eta) - 1$, found by one-dimensional bisection/Newton. For an observed successor ($c_{s,s'} > 0$) the formula gives $\theta_{s'} > 0$; for a zero count it reduces to $\theta_{s'} = \max(\rho a_{s'} - \eta, 0)/\rho$, which can be 0. The closed form is exact in both cases, so the step has no active-set combinatorics. It is strongly convex and embarrassingly parallel over rows.

Z-step: the graph-ridge smoother. The Z-step (13) is a quadratic with stationarity $2\lambda L_0 Z + \rho(Z - (\Theta^{k+1} + U^k)) = 0$, i.e.

$$Z^{k+1} = \rho\,(2\lambda L_0 + \rho I)^{-1}(\Theta^{k+1} + U^k).$$

This is the graph-ridge smoother of Lemma 1, $(W_F + 2\lambda L_0)^{-1} W_F$ with $W_F = \rho I$, applied column-wise. Because $\rho I \succ 0$ on every state, no reduction to visited states is needed. The matrix $2\lambda L_0 + \rho I$ is fixed, sparse, and SPD, so a single Cholesky (or a few CG iterations) is precomputed and reused every iteration.

Remark 3 (The alternation is GPLD's "fit then smooth")**.** The Θ-step is a per-row constrained maximum-likelihood fit; the Z-step is graph smoothing by $(2\lambda L_0 + \rho I)^{-1}$; the dual U enforces their consistency. GPLD's estimator is thus the fixed point of a data fit alternated with the graph smoother of Lemma 1, and ADMM makes that alternation exact and convergent (Theorem 8).

F.1 CONVERGENCE

This is the averaged-operator / Douglas–Rachford argument of Bauschke & Combettes (2017), instantiated for the consensus splitting $\Theta = Z$ above. Fix $\gamma = 1/\rho > 0$ and write the resolvents (proximity operators) $J_f = \mathrm{prox}_{\gamma f} = (\mathrm{Id} + \gamma \partial f)^{-1}$, $J_g = \mathrm{prox}_{\gamma g}$, and the reflected resolvents $R_f = 2J_f - \mathrm{Id}$, $R_g = 2J_g - \mathrm{Id}$. We use four facts, all from Bauschke & Combettes (2017): (F1) $\partial f, \partial g$ are maximal monotone (f, g closed proper convex); (F2) each resolvent $J = (\mathrm{Id} + \gamma \partial \phi)^{-1}$ is firmly nonexpansive; (F3) hence each reflected resolvent $R = 2J - \mathrm{Id}$ is nonexpansive; (F4) if $T = (1-\alpha)\mathrm{Id} + \alpha R$ with R nonexpansive and $\alpha \in (0,1)$ (an α-averaged operator) has $\mathrm{Fix}\, T \neq \emptyset$, then for any w^0 the iterates $w^{k+1} = Tw^k$ converge to a point of $\mathrm{Fix}\, T$ (Krasnosel'skiĭ–Mann).

The Douglas–Rachford operator. Define the Douglas–Rachford operator (Lions & Mercier, 1979) $T := \frac{1}{2}(\mathrm{Id} + R_f R_g) = \mathrm{Id} - J_g + J_f(2J_g - \mathrm{Id})$. By (F2)–(F3), R_f and R_g are nonexpansive, so $R_f R_g$ is nonexpansive and T is $\frac{1}{2}$-averaged.

Lemma 2 (Fixed points are solutions)**.** *$w \in \mathrm{Fix}\, T$ iff $x := J_g(w)$ satisfies $0 \in \partial f(x) + \partial g(x)$, i.e. x (with $z = x$) minimizes $f + g$; equivalently $x = \hat{P}_\lambda$.*

Proof. $w = Tw \Leftrightarrow J_g(w) = J_f(2J_g(w) - w)$. Writing $x = J_g(w)$, the resolvent inclusions $w - x \in \gamma \partial g(x)$ and $(2x - w) - x = x - w \in \gamma \partial f(x)$ add to $0 \in \gamma(\partial f(x) + \partial g(x))$, and $\partial f + \partial g = \partial(f + g)$ because g is finite and continuous. Conversely, if $u \in \partial g(x)$ and $-u \in \partial f(x)$, then $w := x + \gamma u$ satisfies $J_g(w) = x$ and $J_f(2x - w) = J_f(x - \gamma u) = x$, so $w \in \mathrm{Fix}\, T$ (Bauschke & Combettes, 2017, Prop. 26.1). □

ADMM is Douglas–Rachford. For the consensus splitting $\Theta = Z$, ADMM for (f, g) is Douglas–Rachford splitting on $\partial f + \partial g$ (Gabay, 1983; Eckstein & Bertsekas, 1992). With $\gamma = 1/\rho$, the updates (12)–(13) read $\Theta^{k+1} = J_f(Z^k - U^k)$ and $Z^{k+1} = J_g(\Theta^{k+1} + U^k)$. Set $w^k := \Theta^{k+1} + U^k$. Then

$$Z^{k+1} = J_g(w^k), \qquad U^{k+1} = w^k - Z^{k+1}, \qquad \Theta^{k+2} = J_f(Z^{k+1} - U^{k+1}) = J_f\big(2J_g(w^k) - w^k\big),$$

so $w^{k+1} = \Theta^{k+2} + U^{k+1} = w^k - J_g(w^k) + J_f(2J_g(w^k) - w^k) = Tw^k$. ADMM therefore inherits the convergence of T, and its iterates are recovered from w^k by the three identities above.

Theorem 8 (Convergence via the fixed-point argument)**.** *The functions f, g are closed proper convex and (11) has a minimizer, so $\mathrm{Fix}\, T \neq \emptyset$. For any initialization, $w^k \to w^\star \in \mathrm{Fix}\, T$ and the ADMM iterates converge: $Z^k \to \hat{P}_\lambda = J_g(w^\star)$, $\Theta^k \to \hat{P}_\lambda$, and $U^k \to w^\star - J_g(w^\star)$. If moreover every count is positive, $c_{s,s'} \geq 1$ for all (s, s'), then f is κ-strongly convex with $\kappa := \min_{s,s'} c_{s,s'}$, ∇g is ℓ_g-Lipschitz with $\ell_g := 2\lambda \left\| L_0 \right\|_2$, and $w^k \to w^\star$ linearly.*

Proof. Existence. The feasible set, a product of simplices, is compact; $f + g$ is lower semicontinuous and finite at the uniform matrix, so it attains its minimum. Lemma 2 (converse direction) then gives $\mathrm{Fix}\, T \neq \emptyset$. *Convergence.* (F4) applied to the $\frac{1}{2}$-averaged T gives $w^k \to w^\star$. By continuity of the resolvents, $Z^{k+1} = J_g(w^k) \to J_g(w^\star)$, which is $\hat{P}_\lambda$ by Lemma 2; $\Theta^{k+2} = J_f(2J_g(w^k) - w^k) \to J_f(2J_g(w^\star) - w^\star) = J_g(w^\star)$ by the fixed-point identity; and $U^{k+1} = w^k - J_g(w^k) \to w^\star - J_g(w^\star)$. *Rate.* When every $c_{s,s'} \geq 1$, each term $-c_{s,s'} \log \theta$ has curvature $c_{s,s'}/\theta^2 \geq c_{s,s'} \geq \kappa$ on $(0, 1]$, so f is κ-strongly convex on the product of simplices and ∂f is κ-strongly monotone; $\nabla g = 2\lambda L_0(\cdot)$ is ℓ_g-Lipschitz, so ∂g is $\frac{1}{\ell_g}$-cocoercive. This is the strongly-monotone-plus-cocoercive split condition of Giselsson (2017, Assumption 5.1 and Thm. 5.6) (see also Davis & Yin (2017)), not the hypothesis of Giselsson & Boyd (2017, Thm. 2) that one function be both strongly convex and smooth. Under it T is a strict contraction with modulus < 1 depending on κ, ℓ_g, γ, and Banach's theorem gives geometric convergence of w^k, transferred to the ADMM iterates by nonexpansiveness of the resolvents. □

The rate hypothesis requires every successor to be observed from every source state. If some count is zero, f is not strongly convex on the product of simplices (that entry carries only the simplex indicator, not a log term). In the sparse regime our tabular experiments target, many counts are zero, so there only the first, rate-free claim of Theorem 8 applies.

Remark 4 (What the fixed-point view isolates, and its scope)**.** The only problem-specific inputs are maximal monotonicity of $\partial f, \partial g$ (automatic for closed proper convex f, g) and existence of a minimizer (compactness of the feasible set); when every count is positive, strong convexity of f upgrades convergence to a geometric rate through the Banach contraction. The guarantee is *computational*: Theorem 8 bounds the algorithm's convergence to $\hat{P}_\lambda$ and says nothing about the estimator's statistical error, which is a property of $\hat{P}_\lambda$ independent of the solver. It is complementary to Theorem 1, which bridges the closed-form smoother to $\hat{P}_\lambda$ so the bias–variance formulas transfer, whereas ADMM *computes* $\hat{P}_\lambda$.

G TABULAR EXPERIMENT DETAILS

The `var-gaus` environment. The $S = 256$ states lie on a circle, and there are three actions. With $k = S/16 = 16$ and offsets $j \in \{1, \dots, k\}$, each state s has a transition width $\sigma(s) = (k/S)\,(1 + \frac{1}{4}\cos(2\pi s/S))$ and weights

$$w_j(s) \;\propto\; \exp\left(-\frac{\left(2\pi\,|j - k/2|/S\right)^2}{2\,\sigma(s)^2}\right), \qquad \sum_j w_j(s) = 1.$$

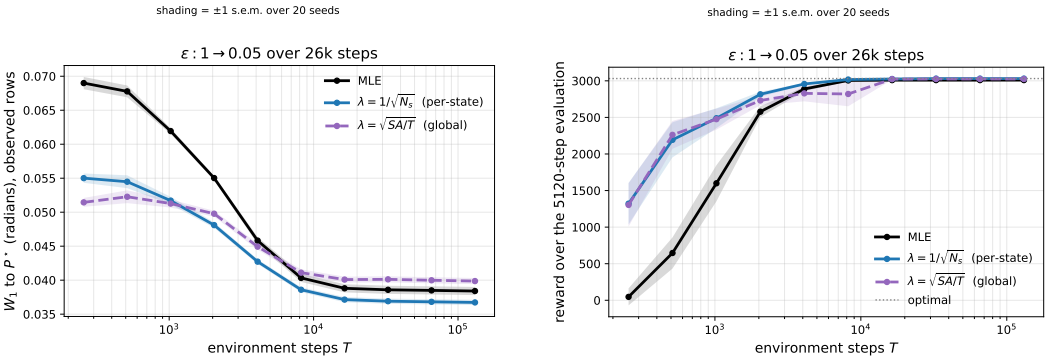
Action 0 moves to $s+j$ and action 1 to $s-j$ (modulo S) with probability $w_j(s)$. Action 2 stays with probability 0.8 and moves to $s \pm j$ with probability $0.1\,w_j(s)$ each. The kernel therefore changes smoothly with s, which is the local structure the penalized estimator exploits, while no two states share the same transition law. Rewards depend on the current state only: a goal region of 17 states centered at $5S/8$ pays 0.5 and its center pays 1, $\lfloor S/10 \rfloor = 25$ randomly placed states pay 0.1 and another 25 pay -0.8, and all other states pay 0. The discount is 0.99.

Protocol. Each of 20 seeds is a single online interaction of 131,072 steps. The agent acts ϵ-greedily with respect to the policy obtained by value iteration on its current transition estimate, so each estimator steers its own data collection. The exploration rate decreases linearly from 1 to 0.05 over the first 26,214 steps (20% of the budget) and then stays at 0.05. At $T = 256 \cdot 2^i$ steps, $i = 0, \dots, 9$, each estimate is refit from the transitions collected so far and scored in two ways: the Wasserstein-1 distance to $P^\star$ on the circle, in radians, averaged over the observed state and action pairs, and the mean episodic reward of its greedy policy over 10 evaluation episodes of 5,120 steps on the true MDP.

Estimators. As in Section 2, $N_s := N(s,a)$ is the visit count of the state and action pair. The maximum-likelihood estimate is the empirical transition frequency, $N(s,a,s')/N_s$, with the uniform distribution for state and action pairs not yet visited. The penalized estimator of (5) uses a neighborhood graph that links each state to the four nearest states on either side along the circle, with weights $1/(\delta^2 + d_{sk}^2)$ for the distance d_{sk} between their embeddings and $\delta = 10^{-5}$, rescaled to a mean row sum of 1. It is solved by ADMM with 200 iterations. Two coefficients are compared. The per-state coefficient $\lambda(s,a) = 1/\sqrt{N_s}$ depends on the visit count of that state and action pair and is 1 for pairs not yet visited. The global coefficient $\lambda = \sqrt{|\mathcal{S}||\mathcal{A}|/T}$ is shared by all pairs and decays with the total number of transitions T, since each transition adds one to the total count. The per-state coefficient enters the penalty of (4) pair by pair: the term for s and k is weighted by $(\lambda(s,a) + \lambda(k,a))/2$, which keeps the penalty symmetric, and ADMM solves the resulting problem exactly (Appendix C.1 gives the induced estimator in this case). The global coefficient is the form used for the latent world model. Under uniform visitation, $N_s = T/(|\mathcal{S}||\mathcal{A}|)$, the two are equal. Both coefficients decay, although the summed likelihood already makes the mixing weight fall as $1/N$ (Appendix C.1); with them it falls as $N^{-3/2}$, faster than $\theta_s^\star$. The coefficient sets the bias–variance trade-off at each count, and the square-root form mirrors the schedule used for the latent world model (Section 4).

Results. Each arm's estimator drives its own data collection, so the arms differ in the data they see as well as in how they use it. At 32k steps the mean regret against the optimal evaluation reward of 3031 is 0.3 ± 0.2 for the per-state coefficient (worst seed 2.9), 6.1 ± 5.8 for the global one (one seed at 115.6) and 21.8 ± 12.8 for maximum likelihood. At 131k steps the per-state estimate remains more accurate than maximum likelihood (Wasserstein distance 0.0367 against 0.0384), and the global one is slightly less accurate (0.0399).

Effective sample size. The ratio N_s^{eff}/N_s of (6), over the state and action pairs visited in at least five of the 20 seeds, peaks while data is scarce, at 1.8 for the per-state coefficient and 2.3 for the global one (Figure 5). Under the global coefficient, which shrinks with every transition, it returns to 1.0 by 32k steps. Under the per-state coefficient it holds at 1.2, because the coefficient stays large on the state and action pairs the settled policy no longer visits. Both remain well below 4.8, the ceiling $1 + m_{\text{eff}}$ of the one-step estimator under equal visitation for the ± 4 neighbor graph used here (Appendix C.1).



(a) Wasserstein distance to $P^\star$.

(b) Mean episodic reward.

Figure 4: Online tabular RL on `var-gaus`, $|\mathcal{S}| = 256$, with the per-state and the global coefficient. Sharing information across neighboring states improves both the transition estimate and the policy derived from it, with the advantage concentrated early in training. Mean $\pm$ SEM over 20 seeds.

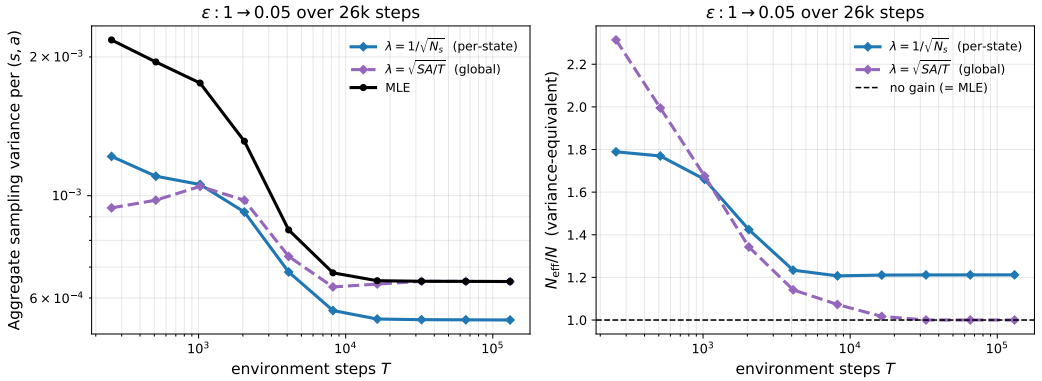


Figure 5: Effective sample size, defined in (6), relative to the true count, against environment steps, mean over 20 seeds, for the per-state and the global coefficient. The gain is largest when data is scarce and falls as counts accumulate. Under the global coefficient it returns to one; under the per-state coefficient it levels off once the policy stops visiting most state and action pairs. This is the self-limiting behavior that (3) predicts.

H DEEP EXPERIMENT DETAILS

Architecture and training. Both arms use DreamerV3 (Hafner et al., 2025) in the `size12m` configuration with unchanged losses, loss weights and optimizer, and the settings of Table 4. GPLD adds only the posterior penalty of (7). The same configuration is used on every task, with exceptions noted where they occur: under pixel observations the penalty starts after 200k environment steps, the warm-start and ManiSkill2 diagnostics use $\lambda_0 = 0.5$, and ManiSkill2 uses the settings given in Appendix J.

Table 4: Settings of both arms. The lower block applies to GPLD only.

Setting	Value
Model size	`size12m`
Deterministic state $\dim(h_t)$	2048
Stochastic state	$K = 32$ categorical variables of $C = 16$ classes
Hidden units (MLPs, RSSM)	256
Categorical mixture with uniform (unimix)	0.01
Free nats	1.0
Loss weights (dynamics, representation)	1.0, 0.1
Optimizer	DreamerV3 default, learning rate 4×10^{-5}, adaptive gradient clipping 0.3
Batch	$B = 16$ sequences of length $L = 64$
Train ratio (replayed steps per agent step)	512
Imagination horizon H	25
Action repeat	2
Parallel environments (train, evaluation)	16, 4
Evaluation	every 10k environment steps, 3 episodes
Replay capacity	5M, uniform sampling
Compute precision	bfloat16
Penalized map	posterior only
Coefficient λ_0	0.25
Decay	$\lambda_{(n)} = \lambda_0/\sqrt{1 + n/1000}$, n = optimizer updates, no floor
Sampled fraction of batch states ρ	0.5
Rows per state and probes per row	all $K = 32$ rows, one Rademacher probe
Penalty start	step 0 (proprioceptive), 200k environment steps (pixels)

Evaluation. Final return is the mean of the evaluation returns over the last 100k environment steps before the task's horizon. nAUC is the trapezoidal area under the evaluation curve from 10k steps to the horizon divided by the elapsed steps, so it is on the return scale. A task is saturated when DreamerV3's nAUC over 10k to 1M steps, divided by 1000, is at least 0.80. Aggregates are the mean and the median over tasks of the normalized return, GPLD's task mean divided by DreamerV3's, with 95% intervals from a stratified bootstrap (10,000 resamples) that resamples seeds within each task for both arms (Agarwal et al., 2021). Learning curves are smoothed with an exponential moving average for display only; no reported number is computed from smoothed curves. The main tables use a fixed number of seeds per task and arm: five on the proprioceptive benchmark (three on `cartpole-balance`, `cartpole-balance-sparse` and `cartpole-swingup`) and six on `humanoid` and `dog`. Where more seeds were completed, a task's table entries use the first of them in a fixed order that both arms completed, a rule set before results were seen; every completed seed enters the per-seed tests of Appendix O. Other tables give their seed counts in the caption or in parentheses. The configurations of the comparison regularizers and the protocol that selected them are given in Appendix N.

Configuration and released DreamerV3 scores. Both arms use the configuration published for this suite: the 12M-parameter model, train ratio 512 and action repeat 2, run to 1M environment steps (Hafner et al., 2025). The one departure is the imagination horizon, 25 rather than 15. It was chosen in preliminary experiments with an earlier Dreamer agent, where 25 outperformed 15 and 30, and fixed before any GPLD run. The code is the official JAX implementation (upstream commit

b65cf81, September 2025). For reference, Table 5 lists the DreamerV3 scores released with that repository, which come from an earlier version of the code (April 2024) run to 500k steps at the default horizon; on several locomotion tasks they are higher than both arms here. The horizon does not explain this gap: DreamerV3 rerun at the default horizon of 15 on the two tasks with the largest gap, one seed each to 500k steps, scores 177.8 on walker-run and 486.0 on finger-spin, below its mean at a horizon of 25 (202.6 and 549.3) and far below the released 632.7 and 937.2, all as the mean over the last 100k steps. GPLD run the same way, at a horizon of 15, scores 235.7 and 732.4, so the gain persists at the default horizon. Every comparison in this paper is between arms that share all settings, so the gain it measures is that of adding the penalty to a fixed agent.

Table 5: Final return at 500k environment steps (mean over the last 100k steps, ± SEM over seeds): the DreamerV3 scores released with the official repository (earlier code, default horizon) and the two arms of this paper at the same budget. Seed counts for our arms are those of Table 1.

Task	Released DreamerV3 $n = 5$	DreamerV3 (ours)	GPLD (ours)
acrobot-swingup	154.5 ±8.6	205.2 ±11.3	232.8 ±17.9
cartpole-balance	990.5 ±1.9	992.5 ±3.5	996.9 ±0.0
cartpole-balance-sparse	996.8 ±3.0	991.2 ±8.6	998.6 ±0.8
cartpole-swingup	850.0 ±6.9	861.2 ±4.4	871.4 ±0.3
cartpole-swingup-sparse	468.1 ±173.7	636.9 ±133.2	746.2 ±11.6
cheetah-run	575.9 ±13.2	501.5 ±24.4	534.8 ±26.6
finger-spin	937.2 ±10.4	549.3 ±30.5	582.3 ±76.4
finger-turn-easy	745.4 ±62.7	858.1 ±18.8	858.6 ±15.9
finger-turn-hard	841.0 ±30.4	705.3 ±51.7	630.6 ±95.1
hopper-hop	111.0 ±14.7	29.3 ±11.9	45.5 ±27.6
hopper-stand	573.2 ±115.3	251.0 ±98.4	373.1 ±133.2
pendulum-swingup	766.0 ±38.4	797.2 ±22.2	836.4 ±13.1
reacher-easy	947.1 ±3.4	957.3 ±7.0	964.9 ±4.1
reacher-hard	936.2 ±8.3	935.5 ±9.8	927.2 ±17.2
walker-run	632.7 ±28.8	202.6 ±33.8	380.3 ±39.3
walker-stand	956.9 ±7.7	907.8 ±9.5	934.3 ±15.1
walker-walk	935.7 ±4.8	744.7 ±106.5	810.9 ±59.1
Mean over 17 tasks	730.5	654.5	689.7

Probes. Each of the K rows is estimated with its own Rademacher probe, since the rows are independent categoricals given u_t; a single probe over all KC outputs would add the cross-row terms $\xi_i^\top J_i J_j^\top \xi_j$ to the estimate. More probes per row help little. On walker-run at 500k steps (three seeds, both arms), probe noise accounts for 14 to 25% of the noise in the penalty with one probe per row; the rest comes from which states are sampled, so the fraction ρ rather than the number of probes sets the precision of the estimate.

Hardware and wall-clock time. Runs used NVIDIA A100 GPUs on a shared cluster. Eleven of the runs behind Table 1 were trained on an NVIDIA RTX 4090 with the same code and configuration: both arms of the same seeds on acrobot-swingup, cartpole-swingup-sparse and walker-walk, and one DreamerV3 seed on walker-stand. Without them, the unsaturated normalized return is 1.186 rather than 1.181. One GPLD seed on each quadruped task also ran on the 4090; without it, GPLD's nAUC gain is 23.0% on run and 49.4% on walk, against 24.1% and 32.9% with it. Part of the regularizer comparison also ran on the 4090. Table 6 gives the wall-clock time of complete runs on the A100 nodes only, five seeds per task and arm. The overhead is set by ρ and the second backward pass of double backpropagation (Drucker & LeCun, 1992) rather than by the number of probes: on the six tasks timed at every ρ, GPLD's time relative to DreamerV3's is 1.11 at $\rho = 0.25$ and 1.68 at $\rho = 1$. Under pixel observations it is 1.19, because the decoder, which the penalty does not touch, dominates the baseline's time. Across the proprioceptive tasks the factor ranges from 1.39 to 1.82.

More gradient updates for DreamerV3. The penalty's second backward pass makes a run 1.50 times as long. Spending the same factor on DreamerV3 as extra gradient updates, a train ratio of

Table 6: Wall-clock time of a complete run in hours, mean over five seeds, and GPLD's time relative to DreamerV3's, for the runs timed on the A100 nodes.

Observations	Task	DreamerV3 (h)	GPLD (h)	Ratio
Proprioceptive	`acrobot-swingup`	1.96	3.07	1.57
	`acrobot-swingup-sparse`	1.97	2.78	1.41
	`cartpole-balance`	1.95	3.11	1.59
	`cartpole-balance-sparse`	1.95	2.75	1.41
	`cartpole-swingup`	1.95	2.82	1.44
	`cartpole-swingup-sparse`	1.98	2.88	1.45
	`cheetah-run`	2.01	3.16	1.57
	`finger-spin`	2.01	2.97	1.48
	`finger-turn-easy`	1.99	2.84	1.42
	`finger-turn-hard`	2.03	2.88	1.42
	`hopper-hop`	2.06	2.95	1.44
	`hopper-stand`	2.08	3.20	1.54
	`pendulum-swingup`	1.96	2.72	1.39
	`quadruped-run`	2.06	2.94	1.42
	`quadruped-walk`	1.77	3.21	1.82
	`reacher-easy`	1.99	2.88	1.45
	`reacher-hard`	2.00	3.02	1.51
	`walker-run`	2.02	3.20	1.58
	`walker-stand`	2.05	3.13	1.53
	`walker-walk`	2.10	3.17	1.51
	Mean (median) over tasks			1.50 (1.47)
Pixels	`cheetah-run`	5.76	6.95	1.21
	`hopper-stand`	5.82	7.07	1.22
	`pendulum-swingup`	5.64	6.77	1.20
	`quadruped-run`	5.93	6.94	1.17
	`walker-run`	5.79	7.03	1.21
	`walker-walk`	5.84	6.72	1.15
	Mean (median) over tasks			1.19 (1.20)

768 instead of 512 with every other setting unchanged, does not improve it on the three tuning tasks (Table 7): its normalized return is 0.935 in final return and 0.958 in nAUC, against 1.277 and 1.353 for GPLD, which is ahead of it by more than the sum of the standard errors on two of the three tasks. GPLD's advantage therefore does not come from the additional computation.

Table 7: DreamerV3 with 1.5 times the gradient updates per environment step (train ratio 768) on the tuning tasks at 1M environment steps, mean $\pm$ SEM over six seeds per arm, the seeds of Table 28. Normalized values are relative to DreamerV3 at the default train ratio of 512; bold marks an entry better than it by more than the sum of the two standard errors.

	Final return			nAUC		
Task	DreamerV3	DreamerV3, 768	GPLD	DreamerV3	DreamerV3, 768	GPLD
`cheetah-run`	638.2 ± 32.8	629.4 ± 9.9	$\mathbf{751.3 \pm 26.7}$	472.1 ± 15.1	460.5 ± 10.1	$\mathbf{512.9 \pm 14.1}$
`walker-run`	415.2 ± 64.9	325.2 ± 63.3	$\mathbf{630.2 \pm 39.9}$	221.4 ± 30.6	180.5 ± 19.0	$\mathbf{393.7 \pm 21.8}$
`hopper-stand`	642.4 ± 79.8	665.7 ± 42.1	729.7 ± 57.8	309.8 ± 66.6	335.2 ± 36.6	369.7 ± 76.5
Normalized, mean over tasks		0.935	1.277		0.958	1.353

I PER-TASK RESULTS

Proprioceptive tasks. Three tasks do not improve: `finger-turn-hard` (0.989) is the only unsaturated regression, and `cartpole-balance-sparse` (0.963) falls from a baseline of 999.8 ± 0.2, a ceiling effect rather than a failure, and `reacher-easy` is level (1.000). The conclusion does not depend on where the saturation line is drawn: sweeping the threshold from 0.70 to 0.90 moves the unsaturated gain only between $+49.7$ and $+61.9$ points, and at every setting GPLD leads on all but at most one task in the set. On nAUC, GPLD's median gain on the unsaturated tasks is 9.8%, its normalized nAUC on the saturated ones is 1.000, and its gain across all 18 is 13.3% (Table 10). The normalized return weights tasks with low returns heavily. In raw final return, averaged over tasks, GPLD's gain is 9.8% on the unsaturated tasks and 5.8% on all 18; the interquartile mean of the per-task normalized returns, which drops the highest and lowest quarter of tasks, is 1.117 on the unsaturated tasks and 1.063 on all 18.

Pixel tasks. Table 8 aggregates the 20 pixel tasks; GPLD improves final return on 12 of them and nAUC on 8.

Table 8: Pixel DMC, all 20 tasks at 1M environment steps, 5 seeds per arm. The first two columns average each arm's task means over tasks, on the return scale. The normalized-return columns take GPLD's task mean over DreamerV3's and aggregate it over tasks, with 95% intervals from a stratified bootstrap that resamples seeds within each task.

	Mean over tasks		Normalized	
	DreamerV3	GPLD	Mean	Median
Final return	729.5	740.8	1.033 [0.985, 1.139]	1.004 [0.995, 1.011]
nAUC	590.3	580.1	0.989 [0.939, 1.082]	0.993 [0.968, 1.007]

Table 9 lists every pixel task with both metrics, on the runs of Table 8.

Table 9: Pixel DMC at 1M environment steps, mean $\pm$ SEM over five seeds per arm. Final return is the mean of evaluations over the last 100k steps; nAUC is on the return scale.

	Final return		nAUC	
Task	DreamerV3	GPLD	DreamerV3	GPLD
`acrobot-swingup`	277.3 ± 26.6	309.3 ± 36.5	180.3 ± 10.7	177.8 ± 17.8
`acrobot-swingup-sparse`	64.5 ± 11.6	67.3 ± 7.4	24.0 ± 6.5	23.3 ± 4.2
`cartpole-balance`	996.8 ± 0.1	996.9 ± 0.1	973.1 ± 2.0	975.0 ± 2.3
`cartpole-balance-sparse`	1000.0 ± 0.0	1000.0 ± 0.0	953.8 ± 4.2	959.9 ± 3.7
`cartpole-swingup`	868.8 ± 3.8	863.5 ± 3.4	797.6 ± 16.3	787.6 ± 14.7
`cartpole-swingup-sparse`	767.8 ± 14.9	763.1 ± 20.4	564.7 ± 50.3	479.8 ± 93.0
`cheetah-run`	854.8 ± 4.3	861.1 ± 7.1	670.3 ± 12.6	652.7 ± 21.0
`finger-spin`	622.4 ± 88.2	669.8 ± 114.1	593.0 ± 82.4	592.5 ± 93.6
`finger-turn-easy`	481.5 ± 83.8	583.3 ± 81.9	409.4 ± 33.3	442.5 ± 29.9
`finger-turn-hard`	641.6 ± 93.9	587.2 ± 64.5	421.5 ± 20.9	299.1 ± 28.9
`hopper-hop`	272.7 ± 85.7	343.8 ± 63.5	195.2 ± 59.1	203.9 ± 53.5
`hopper-stand`	913.3 ± 9.5	924.8 ± 8.2	795.9 ± 20.9	819.8 ± 15.3
`pendulum-swingup`	817.0 ± 22.2	765.9 ± 15.1	713.5 ± 15.0	622.3 ± 79.1
`quadruped-run`	572.7 ± 55.0	576.0 ± 52.1	281.1 ± 30.3	307.9 ± 30.8
`quadruped-walk`	821.9 ± 48.3	$\mathbf{896.7 \pm 14.4}$	368.3 ± 39.3	$\mathbf{472.3 \pm 59.1}$
`reacher-easy`	976.0 ± 2.8	973.2 ± 1.5	836.5 ± 2.6	809.0 ± 19.1
`reacher-hard`	935.5 ± 26.6	953.0 ± 15.8	567.3 ± 38.3	538.2 ± 23.2
`walker-run`	765.9 ± 5.0	748.1 ± 15.4	619.2 ± 11.4	597.9 ± 16.1
`walker-stand`	979.0 ± 5.2	971.3 ± 3.2	944.8 ± 1.2	945.2 ± 2.1
`walker-walk`	960.8 ± 3.0	962.3 ± 5.0	896.9 ± 8.2	895.4 ± 2.4

Table 10 gives the nAUC companion of Table 1: the same tasks, seeds, saturation split and aggregation, with nAUC on the return scale over 10k–1M environment steps.

Table 10: Proprioceptive DMC, nAUC over 10k–1M environment steps on the return scale, mean ± SEM over seeds. Columns and aggregates as in Table 1.

Task	A	DreamerV3	GPLD	Normalized
Unsaturated ($A < 0.80$)				
walker-run	0.233	233.4 ± 34.5	$\mathbf{386.8 \pm 25.3}$	1.657
hopper-hop	0.046	46.1 ± 20.0	69.2 ± 21.1	1.501
acrobot-swingup-sparse	0.034	34.3 ± 2.9	$\mathbf{51.4 \pm 11.3}$	1.496
hopper-stand	0.309	309.5 ± 81.6	381.6 ± 92.5	1.233
acrobot-swingup	0.180	180.2 ± 8.1	$\mathbf{210.7 \pm 4.2}$	1.169
walker-walk	0.656	655.7 ± 70.2	731.0 ± 31.2	1.115
cheetah-run	0.476	476.1 ± 17.8	$\mathbf{514.4 \pm 17.1}$	1.080
finger-spin	0.525	525.2 ± 24.7	567.0 ± 60.6	1.080
cartpole-swingup-sparse	0.572	572.0 ± 51.3	616.7 ± 23.5	1.078
pendulum-swingup	0.746	746.0 ± 9.9	$\mathbf{769.5 \pm 9.6}$	1.032
finger-turn-easy	0.777	776.9 ± 12.3	781.0 ± 13.5	1.005
finger-turn-hard	0.641	641.1 ± 25.1	609.1 ± 36.6	0.950
Unsaturated, mean over tasks				1.200 [1.087, 1.612]
Unsaturated, median over tasks				1.098 [1.025, 1.192]
Saturated ($A \geq 0.80$)				
cartpole-swingup*	0.819	819.3 ± 1.9	$\mathbf{835.6 \pm 1.6}$	1.020
cartpole-balance*	0.968	968.0 ± 5.2	$\mathbf{982.3 \pm 0.7}$	1.015
walker-stand	0.850	850.3 ± 20.9	856.3 ± 12.6	1.007
cartpole-balance-sparse*	0.962	962.0 ± 8.3	962.1 ± 12.1	1.000
reacher-hard	0.865	865.4 ± 4.5	848.6 ± 4.5	0.981
reacher-easy	0.926	925.9 ± 5.8	905.6 ± 5.2	0.978
Saturated, mean over tasks				1.000 [0.991, 1.010]
Saturated, median over tasks				1.004 [0.985, 1.016]
All 18, mean over tasks				1.133 [1.058, 1.423]
All 18, median over tasks				1.055 [1.009, 1.065]

Figure 3 (Section 5.4) shows the benchmark over training. Each task's return is divided by DreamerV3's mean final return on that task and the result is averaged over tasks, so every task carries equal weight and the curves end near the normalized returns of Table 1; a ratio taken at every step would instead blow up where DreamerV3's return is near zero early in training. Figure 6 gives every task.

Table 11: Sensitivity to the saturation threshold on DreamerV3's normalized area A (Section 5.1). For each threshold: the number of unsaturated tasks, GPLD's gain in their mean final return, the normalized return, and the tasks that enter (+) or leave (−) the unsaturated set relative to the threshold of 0.80 used in Table 1.

Threshold on A	Unsaturated tasks	Gain in mean return	Normalized return	Change relative to 0.80
0.70	10	+61.9	1.209	−finger-turn-easy, −pendulum-swingup
0.75	11	+59.6	1.194	−finger-turn-easy
0.80	12	+57.7	1.181	–
0.85	13	+55.7	1.170	+cartpole-swingup
0.90	15	+49.7	1.149	+cartpole-swingup, +reacher-hard, +walker-stand

Figure 7 shows all 20 pixel tasks on the runs of Table 8.

Figure 8 shows every dog and humanoid task in Table 2, on the same seeds. On humanoid-walk, the one task where GPLD does not improve, DreamerV3 begins to climb earlier and the two arms finish level.

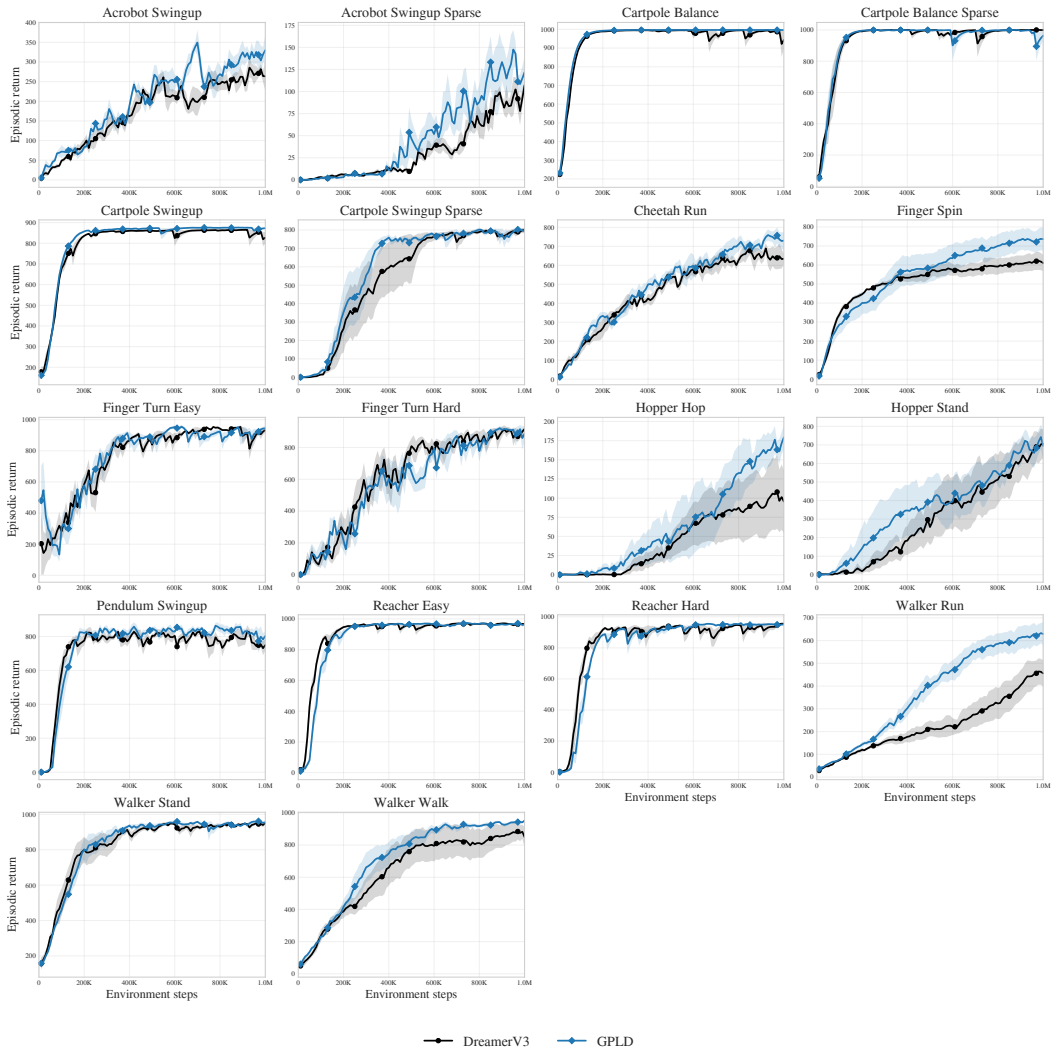


Figure 6: Learning curves on all 18 proprioceptive DMC tasks. Mean ± SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.3$).

`humanoid` and `dog`. The largest nAUC gain, +103.1% on `humanoid-stand`, rests on a small denominator: DreamerV3 stays near zero until late in training, so its area over the horizon is small, and the median over tasks is the more reliable summary. Table 12 gives the gains behind Table 2: GPLD's task mean over DreamerV3's, the per-seed paired gain, and the mean and median over tasks with 95% intervals.

`quadruped`. Table 13 reports each task at 3M and 4M steps, along with nAUC over the full 4M, on the same five seeds per arm in every column. GPLD leads on 4 of 5 seeds on `run` and 3 of 5 on `walk` at 3M, and the medians show the same gap, 12% and 108% higher. On `quadruped-walk`, one DreamerV3 seed never learns stably: its evaluations alternate between near-solved (above 950) and near-zero throughout training, so its final score is 184.0 against 933.8 for the other seeds. The advantage does not depend on this seed. With that seed removed from both arms, GPLD's mean return at 3M is 860.3 against 621.7 for DreamerV3, 38% higher on the mean and 42% on the median; GPLD also reaches 800 first on 3 of the 4 remaining seeds and keeps a 22.3% higher nAUC. `quadruped-run` has no such seed.

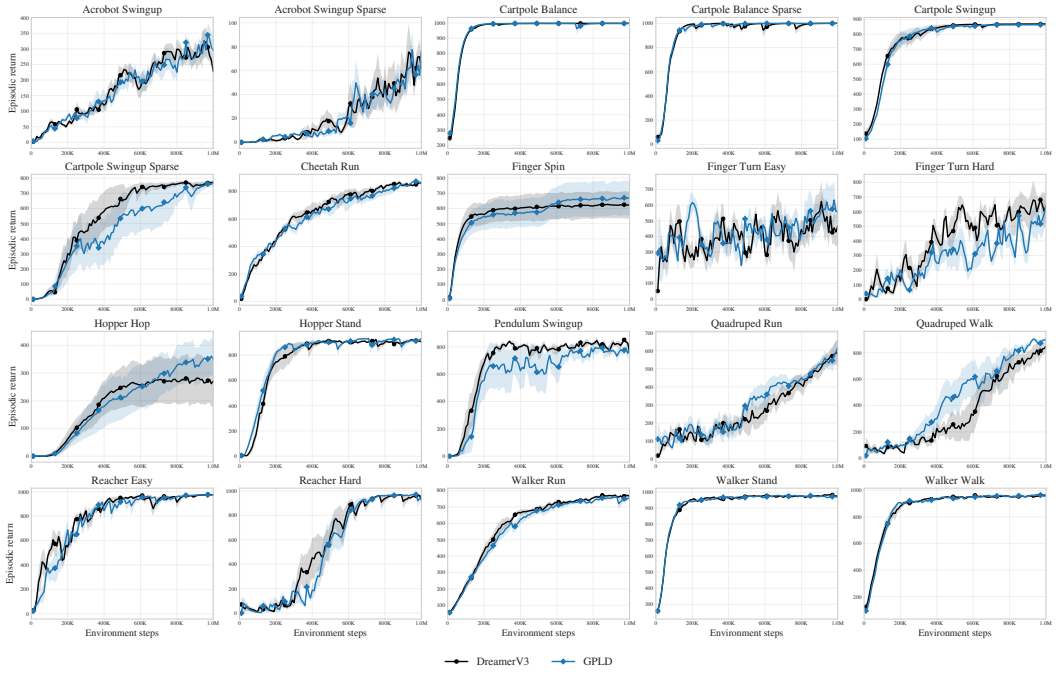


Figure 7: All 20 pixel DMC tasks at 1M environment steps, mean ± SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.3$).

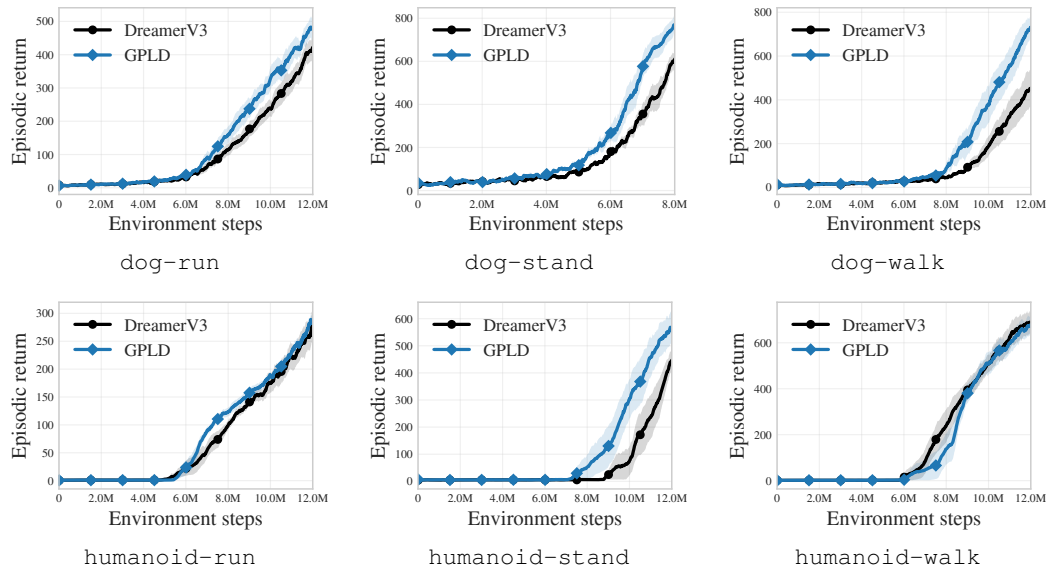


Figure 8: All six `dog` and `humanoid` tasks at $\lambda_0 = 0.25$, mean ± SEM over seeds at each task's horizon, smoothed with an exponential moving average ($\alpha = 0.1$).

Table 12: Gains behind Table 2: `humanoid` and `dog` at each task's horizon, all at $\lambda_0 = 0.25$. Final return is the mean of evaluations over the last 100k steps. Gain is GPLD's task mean over DreamerV3's; Paired is the mean $\pm$ SEM, over the six seeds completed by both arms, of the per-seed relative difference. nAUC is on the return scale (Section 5). Aggregates are the mean and median over tasks, with 95% intervals from a stratified bootstrap that resamples seeds within each task.

		Final return		nAUC	
Task	Horizon	Gain	Paired	Gain	Paired
`dog-run`	12M	$+13.3\%$	$+14.9 \pm 9.2\%$	$+27.1\%$	$+33.9 \pm 20.6\%$
`dog-stand`	8M	$+25.8\%$	$+28.7 \pm 11.6\%$	$+41.4\%$	$+48.0 \pm 21.6\%$
`dog-walk`	12M	$+60.1\%$	$+104.2 \pm 53.8\%$	$+74.6\%$	$+138.2 \pm 81.6\%$
`humanoid-run`	12M	$+0.7\%$	$+6.2 \pm 11.3\%$	$+11.9\%$	$+20.7 \pm 17.3\%$
`humanoid-stand`	12M	$+26.1\%$	$+32.1 \pm 20.2\%$	$+103.1\%$	$+191.3 \pm 98.2\%$
`humanoid-walk`	12M	-1.2%	$+1.2 \pm 9.4\%$	-9.4%	$-3.1 \pm 16.6\%$
Mean over tasks		$+20.8\%$	[+10.6%, +36.6%]	$+41.4\%$	[+20.8%, +78.1%]
Median over tasks		$+19.5\%$	[+5.2%, +30.4%]	$+34.2\%$	[+15.3%, +55.0%]

Table 13: `quadruped` on the seeds completed by both arms. Return at 3M and 4M is the mean of evaluations over the preceding 100k steps, summarized over seeds by the mean ($\pm$ SEM) and the median. nAUC covers 0–4M and is normalized by DreamerV3's mean final return, so DreamerV3's own value is below 1. Steps to reach 800 is the first step at which a seed's windowed return reaches 800, as the median over seeds. †Over the seeds that reach 800 in both arms, since one DreamerV3 seed never does.

	`quadruped-run` ($n = 5$)		`quadruped-walk` ($n = 5$)	
	DreamerV3	GPLD	DreamerV3	GPLD
Mean return, 3M	637.3 ± 134.1	$\mathbf{857.0 \pm 20.2}$	516.5 ± 158.1	$\mathbf{845.9 \pm 26.7}$
Mean return, 4M	899.2 ± 15.5	913.2 ± 7.0	783.8 ± 150.0	912.8 ± 35.0
Median return, 3M	773.9	865.3	422.7	878.8
Median return, 4M	920.5	921.2	931.2	948.7
nAUC, 0–4M	0.453 ± 0.047	$\mathbf{0.562 \pm 0.045}$	0.450 ± 0.067	$\mathbf{0.598 \pm 0.042}$
Steps to reach 800	2.62M	2.23M	3.11M†	2.62M†

J MANIPULATION

Contact-rich manipulation is a natural next setting for GPLD, and a first result is promising. On `PickCube-v0` from ManiSkill2 (Gu et al., 2023), over four seeds, GPLD's mean maximum success rate at 5.95M environment steps is 0.968 ± 0.016, against 0.925 ± 0.018 for DreamerV3 (Figure 9). GPLD also gets there sooner: its smoothed success rate first exceeds 0.9 after $1.89\text{M} \pm 0.21\text{M}$ environment steps on average, against $2.79\text{M} \pm 0.30\text{M}$ for DreamerV3. Both arms use state observations with end-effector position control (`pd_ee_delta_pos`), an imagination horizon of 15, a replay capacity of 1M and 16 evaluation episodes, and train for 6M environment steps; all other settings are those of Table 4.

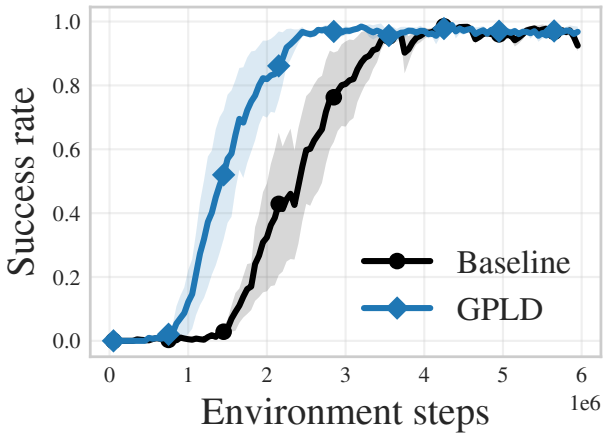


Figure 9: `PickCube-v0` from ManiSkill2: maximum success rate per evaluation episode, mean $\pm$ SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.3$). The GPLD runs here use $\lambda_0 = 0.5$.

K ABLATIONS

The ablations come from two groups of runs. The first uses the configuration of the main experiments (1M environment steps, λ_0 decaying as $\lambda_0/\sqrt{1+n/1000}$) and covers the sampling fraction ρ and the choice of the penalized map. The remaining questions were answered by a second group of runs at 500k environment steps. Each comparison in this group is made against its own DreamerV3 runs, and we report the coefficient of each arm. Every table gives final return, mean $\pm$ SEM over seeds, and the mean over tasks of each arm's normalized return and normalized nAUC.

Sampling fraction. Table 14 varies ρ at $\lambda_0 = 0.25$. The configuration used throughout the paper, $\rho = 0.5$, has the highest mean ratio on final return (1.224) and on nAUC (1.233); the other values give 1.131 to 1.161 on final return. Every value of ρ improves on DreamerV3. The $\rho = 0.5$ column is the GPLD arm of Table 1, so its seeds differ from the other columns.

Table 14: Sampling fraction ρ at $\lambda_0 = 0.25$, final return at 1M environment steps, mean $\pm$ SEM over five seeds for DreamerV3 and $\rho = 0.5$ (the seeds of Table 1) and six for the other columns.

Task	DreamerV3	$\rho = 0.25$	$\rho = 0.5$	$\rho = 0.75$	$\rho = 1$
`acrobot-swingup`	268.7 ± 6.1	318.9 ± 14.7	319.9 ± 20.7	293.4 ± 21.1	324.6 ± 9.9
`cheetah-run`	638.8 ± 40.2	665.4 ± 7.4	741.8 ± 30.6	693.6 ± 23.6	657.2 ± 35.5
`finger-spin`	614.9 ± 44.3	584.6 ± 44.4	732.9 ± 57.3	696.5 ± 73.5	649.6 ± 55.4
`hopper-hop`	99.0 ± 43.1	169.6 ± 34.6	171.1 ± 16.2	123.7 ± 43.7	139.9 ± 36.2
`hopper-stand`	681.5 ± 85.2	638.6 ± 133.8	702.4 ± 62.4	657.3 ± 84.2	630.0 ± 53.6
`pendulum-swingup`	762.5 ± 22.8	790.8 ± 15.5	798.5 ± 32.0	827.4 ± 12.9	789.0 ± 11.9
`walker-run`	456.3 ± 61.5	617.2 ± 65.0	625.4 ± 48.5	659.7 ± 18.9	604.0 ± 50.1
`walker-walk`	877.6 ± 50.0	935.4 ± 5.8	937.8 ± 9.4	927.0 ± 7.4	926.5 ± 11.7
Normalized final return		1.161	1.224	1.139	1.131
Normalized nAUC		1.168	1.233	1.204	1.115

Which map to penalize. Table 15 applies the same penalty to the posterior or to the prior, on the same seeds. Penalizing the posterior gives normalized values of 1.106 on final return and 1.100 on nAUC, against 1.008 and 0.997 for the prior. Penalizing both heads with a shared coefficient is also worse than penalizing the posterior alone (Table 16): the posterior alone reaches 1.303, while the shared coefficient gives 1.233, 1.195 and 1.049 as it grows.

Table 15: The penalty applied to the posterior or to the prior, $\lambda_0 = 0.25$, $\rho = 0.5$, final return at 1M environment steps, mean $\pm$ SEM over five seeds for DreamerV3 and three for the posterior and the prior.

Task	DreamerV3	Posterior	Prior
`acrobot-swingup`	268.7 ± 6.1	304.1 ± 35.4	303.0 ± 25.9
`cheetah-run`	638.8 ± 40.2	708.4 ± 48.9	712.9 ± 21.8
`finger-spin`	614.9 ± 44.3	748.2 ± 102.3	607.7 ± 88.2
`hopper-hop`	99.0 ± 43.1	131.7 ± 42.0	87.1 ± 57.5
`hopper-stand`	681.5 ± 85.2	576.5 ± 104.4	551.7 ± 275.1
`pendulum-swingup`	762.5 ± 22.8	798.7 ± 6.4	782.4 ± 9.3
`walker-run`	456.3 ± 61.5	499.5 ± 117.6	485.6 ± 62.0
`walker-walk`	877.6 ± 50.0	938.6 ± 2.1	923.0 ± 5.8
Normalized final return		1.106	1.008
Normalized nAUC		1.100	0.997

Coefficient and schedule. In the 500k-step runs, a coefficient of 0.25 on the posterior gives normalized values of 1.303 on final return and 1.362 on nAUC, slightly above the 1.271 and 1.319 of 0.125 (Table 17). The decay matters more than the exact coefficient (Table 18): across 18 tasks, the decaying coefficient improves on DreamerV3 (1.133) while a constant one falls below it (0.936). That constant arm uses a larger coefficient, so a smaller constant coefficient of 0.05 was also run on

Table 16: 500k-step runs: the posterior alone or both heads with a shared coefficient λ, $\rho = 0.5$, final return at 500k environment steps, mean $\pm$ SEM over five seeds.

Task	DreamerV3	Posterior, $\lambda = 0.25$	Both, $\lambda = 0.125$	Both, $\lambda = 0.25$	Both, $\lambda = 0.375$
cartpole-swingup-sparse	598.7 ± 113.8	748.3 ± 31.1	779.1 ± 4.5	725.7 ± 27.7	593.9 ± 125.1
cheetah-run	467.9 ± 25.4	537.5 ± 31.1	527.1 ± 29.5	519.0 ± 21.0	511.6 ± 16.0
hopper-stand	357.6 ± 100.8	684.6 ± 83.8	583.2 ± 131.2	441.0 ± 166.5	405.4 ± 144.4
pendulum-swingup	789.1 ± 15.6	799.7 ± 19.5	845.9 ± 17.2	817.6 ± 13.0	808.4 ± 25.1
walker-run	240.0 ± 55.0	309.6 ± 40.7	309.2 ± 58.8	342.0 ± 66.9	271.5 ± 38.6
walker-walk	743.8 ± 78.2	893.6 ± 17.7	728.8 ± 62.1	856.4 ± 23.7	682.8 ± 76.4
Normalized final return		1.303	1.233	1.195	1.049
Normalized nAUC		1.362	1.332	1.179	1.102

five tasks (Table 19). It recovers part of the gain but not all of it: normalized values of 1.112 in final return and 1.044 in nAUC, against 1.360 and 1.360 for the decaying coefficient.

Table 17: 500k-step runs: posterior coefficient, $\rho = 0.5$, final return at 500k environment steps, mean $\pm$ SEM over five seeds.

Task	DreamerV3	$\lambda = 0.125$	$\lambda = 0.25$
cartpole-swingup-sparse	598.7 ± 113.8	766.3 ± 15.4	748.3 ± 31.1
cheetah-run	467.9 ± 25.4	522.4 ± 28.7	537.5 ± 31.1
hopper-stand	357.6 ± 100.8	547.9 ± 158.0	684.6 ± 83.8
pendulum-swingup	789.1 ± 15.6	816.3 ± 15.1	799.7 ± 19.5
walker-run	240.0 ± 55.0	370.6 ± 41.5	309.6 ± 40.7
walker-walk	743.8 ± 78.2	833.4 ± 48.9	893.6 ± 17.7
Normalized final return		1.271	1.303
Normalized nAUC		1.319	1.362

Table 18: 500k-step runs: decaying against constant coefficient, $\rho = 0.5$, final return at 500k environment steps, mean $\pm$ SEM over five seeds.

Task	DreamerV3	Decayed, $\lambda_0 = 0.25$	Constant, $\lambda = 0.4$
acrobot-swingup	160.2 ± 15.5	242.7 ± 8.1	237.0 ± 10.3
acrobot-swingup-sparse	22.8 ± 10.9	31.0 ± 10.8	10.1 ± 4.3
cartpole-balance	977.6 ± 12.4	996.7 ± 0.0	996.5 ± 0.3
cartpole-balance-sparse	970.9 ± 26.0	995.5 ± 3.4	998.1 ± 0.8
cartpole-swingup	859.5 ± 3.9	870.9 ± 0.8	865.7 ± 3.6
cartpole-swingup-sparse	598.7 ± 113.8	748.3 ± 31.1	568.7 ± 150.2
cheetah-run	467.9 ± 25.4	537.5 ± 31.1	489.9 ± 32.2
finger-spin	667.7 ± 59.1	607.8 ± 47.8	508.3 ± 16.7
finger-turn-easy	856.5 ± 37.2	883.1 ± 16.9	877.6 ± 25.5
finger-turn-hard	805.4 ± 42.2	848.7 ± 25.6	563.8 ± 107.2
hopper-hop	60.3 ± 22.3	42.8 ± 30.3	28.9 ± 20.0
hopper-stand	357.6 ± 100.8	684.6 ± 83.8	182.8 ± 92.2
pendulum-swingup	789.1 ± 15.6	799.7 ± 19.5	810.9 ± 28.9
reacher-easy	966.6 ± 3.7	950.4 ± 11.9	946.6 ± 12.5
reacher-hard	927.7 ± 29.2	881.5 ± 34.7	855.7 ± 36.9
walker-run	240.0 ± 55.0	309.6 ± 40.7	336.4 ± 59.3
walker-stand	928.5 ± 5.1	932.2 ± 4.3	928.8 ± 16.3
walker-walk	743.8 ± 78.2	893.6 ± 17.7	789.4 ± 43.5
Normalized final return		1.133	0.936
Normalized nAUC		1.115	0.894

Table 19: 500k-step runs: decaying coefficient against a small and a large constant coefficient, $\rho = 0.5$, final return at 500k environment steps, mean $\pm$ SEM over five seeds.

Task	DreamerV3	Decayed, $\lambda_0 = 0.25$	Constant, $\lambda = 0.05$	Constant, $\lambda = 0.4$
`acrobot-swingup`	160.2 ± 15.5	242.7 ± 8.1	246.7 ± 11.5	237.0 ± 10.3
`cartpole-balance`	977.6 ± 12.4	996.7 ± 0.0	996.8 ± 0.0	996.5 ± 0.3
`cheetah-run`	467.9 ± 25.4	537.5 ± 31.1	500.5 ± 39.8	489.9 ± 32.2
`hopper-stand`	357.6 ± 100.8	684.6 ± 83.8	274.2 ± 137.8	182.8 ± 92.2
`walker-walk`	743.8 ± 78.2	893.6 ± 17.7	867.6 ± 37.8	789.4 ± 43.5
Normalized final return		1.360	1.112	1.024
Normalized nAUC		1.360	1.044	0.911

K.1 Warm-starting the encoder under pixel observations

Table 20 and Figure 10 give the diagnostic of Section 5.2: pixel `quadruped-walk` at 1M environment steps, 5 seeds per arm. In the warm-started runs the encoder and decoder are initialized from a pretrained model and every other component is trained from scratch.

Table 20: Warm-start diagnostic on pixel `quadruped-walk` at 1M environment steps, mean $\pm$ SEM over 5 seeds. nAUC is on the return scale; final return is the mean of evaluations over the last 100k steps.

	nAUC		Final return	
Encoder and decoder	DreamerV3	GPLD	DreamerV3	GPLD
From scratch	449.5 ± 54.5	424.9 ± 33.9	830.2 ± 33.3	864.9 ± 16.8
Warm-started	547.3 ± 33.0	$\mathbf{646.7 \pm 23.4}$	883.1 ± 27.6	$\mathbf{921.5 \pm 8.9}$

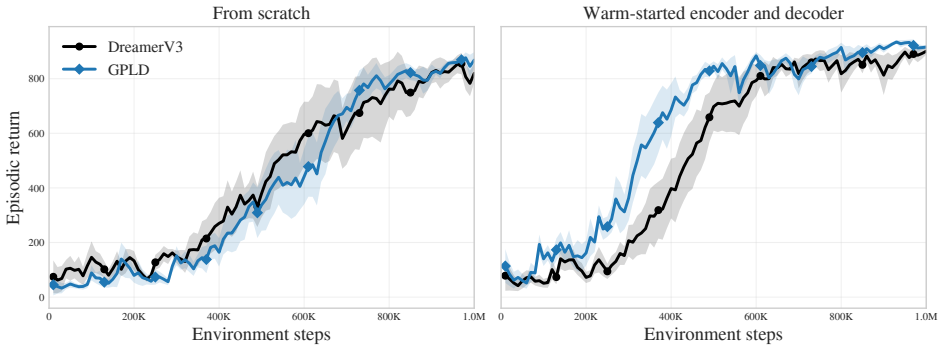


Figure 10: Pixel `quadruped-walk` from scratch (left) and with a warm-started encoder and decoder (right), mean $\pm$ SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.3$). The GPLD runs in this diagnostic use $\lambda_0 = 0.5$ with the penalty active from the first step, rather than the $\lambda_0 = 0.25$ used throughout the paper and the 200k-step delay of the other pixel runs.

L SMOOTHNESS DIAGNOSTICS

The measurements behind Section 5.3 use the 1M-step checkpoints of `cheetah-run`, `walker-run` and `hopper-stand`, with 16 seeds per arm for DreamerV3 and GPLD on `cheetah-run` and `walker-run` and six on `hopper-stand`, and five seeds for VAT ($\lambda = 0.02$, constant coefficient), for SN ($c = 30$ and $c = 120$) and for noise injection and noise consistency, configured as in Appendix N. States come from a fixed corpus of evaluation episodes generated by four policies (DreamerV3, GPLD and two VAT configurations), so no model is scored only on the trajectories it generated. Every state follows a 25-step burn-in on real observations.

Entropy. For each of the K categorical rows, the normalized entropy is $-\sum_c p_c \log p_c / \log C$, which is 0 for a one-hot row and 1 for a uniform one. A row is saturated when its largest probability exceeds 0.9. Table 21 averages both quantities over rows and states.

Table 21: Posterior sharpness at the 1M-step checkpoint, mean ± SEM over 16 seeds for DreamerV3 and GPLD on `cheetah-run` and `walker-run`, six on `hopper-stand`, and five for each comparator. Sharpening would lower the entropy and raise the saturated fraction.

Arm	`cheetah-run`	`walker-run`	`hopper-stand`
Normalized entropy			
DreamerV3	0.503 ± 0.005	0.273 ± 0.003	0.437 ± 0.006
GPLD	0.555 ± 0.004	0.326 ± 0.004	0.461 ± 0.010
VAT	0.316 ± 0.012	0.317 ± 0.005	0.356 ± 0.010
SN, $c = 30$	0.506 ± 0.009	0.290 ± 0.007	0.446 ± 0.006
SN, $c = 120$	0.515 ± 0.010	0.284 ± 0.002	0.432 ± 0.017
Noise injection	0.488 ± 0.014	0.283 ± 0.006	0.439 ± 0.009
Noise consistency	0.366 ± 0.011	0.259 ± 0.004	0.363 ± 0.015
Saturated rows (%)			
DreamerV3	1.6 ± 0.1	24.4 ± 0.5	9.1 ± 0.7
GPLD	1.0 ± 0.0	15.3 ± 0.5	9.0 ± 1.4
VAT	8.6 ± 1.7	14.7 ± 0.8	10.7 ± 1.0
SN, $c = 30$	1.3 ± 0.2	20.7 ± 1.5	8.5 ± 0.6
SN, $c = 120$	1.4 ± 0.1	21.9 ± 0.4	9.9 ± 1.8
Noise injection	1.7 ± 0.6	22.9 ± 1.3	8.0 ± 0.8
Noise consistency	6.0 ± 0.7	25.9 ± 1.0	13.3 ± 2.0

Jacobian energy. Energies are Hutchinson estimates of the squared Frobenius norm of the posterior Jacobian with respect to the head input $u_t = [h_t, e_t]$, with one Rademacher probe per row as in training, evaluated on 128 states per model (32 from each source policy). Every model sees the same states and the same probes. The raw energy is not comparable across models, because the head maps u_t through a linear layer W_0: scaling one block of u_t by a and the matching columns of W_0 by $1/a$ leaves the posterior unchanged and divides that block's energy by a^2. The models do differ in input scale. GPLD's encoder features are larger than DreamerV3's by a factor of 1.6 to 1.8 in root-mean-square and its recurrent state by about 1.2, while no comparator enlarges its inputs (encoder features 0.67 to 1.03 times DreamerV3's, recurrent state 0.93 to 1.10 times). We therefore report the scale-invariant energy $\mathcal{R}_h \,\mathrm{rms}(h_t)^2 + \mathcal{R}_e \,\mathrm{rms}(e_t)^2$ per state, where $\mathcal{R}_h$ and $\mathcal{R}_e$ are the energies with respect to each block. It measures the change in the posterior per change of the input relative to its own typical size. Probability space differentiates the posterior probabilities, which is the quantity GPLD penalizes. Logit space differentiates the pre-softmax logits, which removes the softmax Jacobian from the chain. The two spaces respond oppositely to sharpening. Under a pure temperature change of the logits, $\ell \mapsto a\ell$ with $a > 1$, the logit-space energy grows by a^2, whereas the probability-space energy falls because the softmax Jacobian decays as the rows sharpen. Genuine smoothing lowers both. A model can smooth and sharpen at once, however, since the two lower the probability-space energy together, so each energy is read alongside the entropy and saturated-row fraction of Table 21. Those are the only measures of sharpening used here.

GPLD lowers the energy in both spaces on all three tasks: by 42% to 50% in probability space and by 27% to 35% in logit space. Its entropy rises at the same time (Table 21), so the softmax factor is, if anything, larger under GPLD, and the reduction must come from the logit Jacobian itself.

Table 22: Scale-invariant posterior Jacobian energy at the 1M-step checkpoint, mean ± SEM over seeds as in Table 21. Scales differ between the two spaces, so values are comparable only within a panel. Lower is smoother.

Arm	`cheetah-run`	`walker-run`	`hopper-stand`
Probability space, $\|J_\pi\|_F^2$			
DreamerV3	0.146 ± 0.0063	0.415 ± 0.014	0.237 ± 0.012
GPLD	0.0775 ± 0.0035	0.207 ± 0.011	0.137 ± 0.012
VAT	0.0110 ± 0.00024	0.0653 ± 0.0057	0.0249 ± 0.0010
SN, $c = 30$	0.123 ± 0.011	0.395 ± 0.021	0.206 ± 0.015
SN, $c = 120$	0.151 ± 0.0063	0.415 ± 0.035	0.224 ± 0.020
Noise injection	0.0764 ± 0.0042	0.231 ± 0.015	0.111 ± 0.0039
Noise consistency	0.0386 ± 0.0030	0.146 ± 0.0089	0.0778 ± 0.0055
Logit space, $\|J_\ell\|_F^2$			
DreamerV3	33.6 ± 1.9	177 ± 9.1	60.4 ± 4.1
GPLD	22.0 ± 1.2	130 ± 7.7	41.1 ± 3.4
VAT	10.4 ± 3.0	83.4 ± 5.9	30.2 ± 4.8
SN, $c = 30$	8.81 ± 0.71	21.3 ± 0.90	15.0 ± 0.98
SN, $c = 120$	33.4 ± 1.9	101 ± 9.5	51.2 ± 4.3
Noise injection	21.8 ± 2.4	155 ± 7.1	36.7 ± 2.2
Noise consistency	10.6 ± 1.2	121 ± 12	25.5 ± 3.4

VAT lowers both energies further than GPLD, and its logit-space energy is below GPLD's on `cheetah-run`, `walker-run`, and `hopper-stand`. It does so while sharpening the posterior: on `cheetah-run` and `hopper-stand` its entropy falls well below DreamerV3's and more of its rows saturate (Table 21), so its probability-space reduction also draws on the saturation shortcut discussed in Section 5.3. GPLD smooths less but keeps the posterior softer than DreamerV3's. Spectral normalization shows the reverse pattern. Capping the output layer bounds the logit Jacobian directly, and SN's logit-space energy falls to between 0.12 and 0.99 of DreamerV3's across caps and tasks. Its probability-space energy, however, stays between 0.84 and 1.03 of DreamerV3's, within the seed-to-seed spread, and its entropy is close to DreamerV3's. Constraining the logit map therefore does not by itself make the posterior probabilities less sensitive, and the smoothness prior of Section 2 is stated on those probabilities.

Of the regularizers compared, GPLD is the only one whose energy falls significantly in both spaces on every task ($z < -2$ against DreamerV3, with z the difference of means over the combined standard error) while its entropy does not fall significantly on any task.

The two random-perturbation regularizers of Appendix N.3 fall on either side of this line. Noise consistency lowers both energies further than GPLD, to between 0.26 and 0.35 of DreamerV3's in probability space and between 0.31 and 0.68 in logit space, but its entropy falls significantly on all three tasks, so, as with VAT, part of the reduction is sharpening. Noise injection, which to first order applies the same Jacobian penalty implicitly (Bishop, 1995), comes closest to GPLD: its energy falls to between 0.47 and 0.56 of DreamerV3's in probability space and between 0.61 and 0.87 in logit space, and its entropy does not fall significantly on any task. The reduction is significant everywhere except in logit space on `walker-run` ($z = -1.95$), and its return gain is smaller than GPLD's (Table 32).

The prior. The prior $p_\phi(\hat{z}_t \mid h_t)$ is not penalized, but DreamerV3 trains it toward the posterior. Table 23 scores it at the same diagnostic states and checkpoints as the posterior, on the first six seeds of each task. Its Jacobian depends on h_t only, and its energy is scaled by the input's size as above, $R_h \,\mathrm{rms}(h_t)^2$. Under GPLD the prior's scaled energy is lower on all three tasks, significantly on `walker-run` and `hopper-stand`, and its entropy is higher on all three, so the prior becomes less sensitive to its input without becoming sharper. The reduction is smaller than the pos-

terior's, as expected for a map that is smoothed only through its training targets. Noise injection and noise consistency lower the prior's scaled energy by similar factors, to between 0.73 and 0.80 of DreamerV3's (five seeds each); under noise consistency the prior's entropy also falls significantly, on `hopper-stand` and `cheetah-run`.

Table 23: The unpenalized prior at 1M environment steps: input-scaled Jacobian energy and entropy (nats), mean ± SEM over six seeds, with the ratio of GPLD's energy to DreamerV3's and the Welch test p-value. The posterior's ratio on the same seeds is shown for comparison.

	Prior energy			Prior entropy		Posterior
Task	DreamerV3	GPLD	Ratio (p)	DreamerV3	GPLD	ratio
`walker-run`	1.171 ±0.053	0.672 ±0.021	0.57 (< 0.001)	1.010 ±0.013	1.202 ±0.026	0.43
`hopper-stand`	0.229 ±0.018	0.167 ±0.016	0.73 (0.027)	1.472 ±0.021	1.577 ±0.017	0.58
`cheetah-run`	0.090 ±0.010	0.068 ±0.007	0.75 (0.085)	1.506 ±0.030	1.640 ±0.014	0.56

M OPEN-LOOP PROTOCOL AND EXTENDED RESULTS

Training scores the model only on one-step predictions from observation-corrected states, while the policy is trained on rollouts that apply the model to its own predictions for $H = 25$ steps. Open-loop error measures the model in the regime in which the policy uses it.

Protocol. Every model is scored on one shared pool of windows cut from real evaluation episodes: 200 windows from each source policy, 50 steps long. The first 25 steps are observed and fix the posterior state. The model then imagines the next 25 steps from the recorded actions alone, and each decoded quantity is compared with the true one. Error is cumulative, $\mathrm{Cum}(h) = \sum_{h' \le h} \mathrm{MSE}(h')$. KL is the categorical divergence between the posterior on the true future and the open-loop prior, in nats. A model can predict the trajectories of its own policy much better than those of another: on `walker-run`, cumulative error on the windows of its own policy is lower than on those of another by 33% for DreamerV3 and 43% for GPLD, and on `cheetah-run` by 23% and about 0%. Scoring each arm on its own windows would therefore favor it, so each comparison uses only windows from source policies that neither compared arm generated; for DreamerV3 and GPLD these are the two VAT policies. DreamerV3 and GPLD use the seeds of Table 21.

Table 24: Cumulative open-loop error at $h = 25$, mean $\pm$ SEM over 16 seeds per arm on `cheetah-run` and `walker-run` and six on `hopper-stand`. Lower is better; bold marks GPLD's error lower than DreamerV3's by more than the sum of their standard errors.

Task	Quantity	DreamerV3	GPLD
`cheetah-run`	position	1.875 ± 0.049	$\mathbf{1.621 \pm 0.038}$
	velocity	453.9 ± 11	$\mathbf{428.7 \pm 12}$
	reward	32.08 ± 1.8	33.53 ± 2.2
	KL	549.4 ± 6.0	538.1 ± 9.1
`walker-run`	height	8.744 ± 0.53	8.566 ± 0.61
	orientation	15.47 ± 0.59	15.61 ± 0.75
	velocity	1978 ± 33	2015 ± 48
	reward	34.91 ± 0.97	33.54 ± 1.3
	KL	1215 ± 19	1188 ± 16
`hopper-stand`	position	75.65 ± 7.5	90.34 ± 5.4
	velocity	72.46 ± 2.8	$\mathbf{67.98 \pm 1.1}$
	touch	177.5 ± 12	$\mathbf{149.6 \pm 13}$
	reward	58.62 ± 5.8	$\mathbf{43.29 \pm 6.4}$
	KL	721.8 ± 40	658.4 ± 46

GPLD's error at $h = 25$ is lower than DreamerV3's on ten of the 14 task and quantity pairs, significantly ($|z| > 2$, with z the difference of means over the combined standard error) on one, and significantly higher on zero (Table 24). The gains are largest on `cheetah-run` positions and on `hopper-stand` reward and contact, and `walker-run` is level. The only sizeable increase is on `hopper-stand` position.

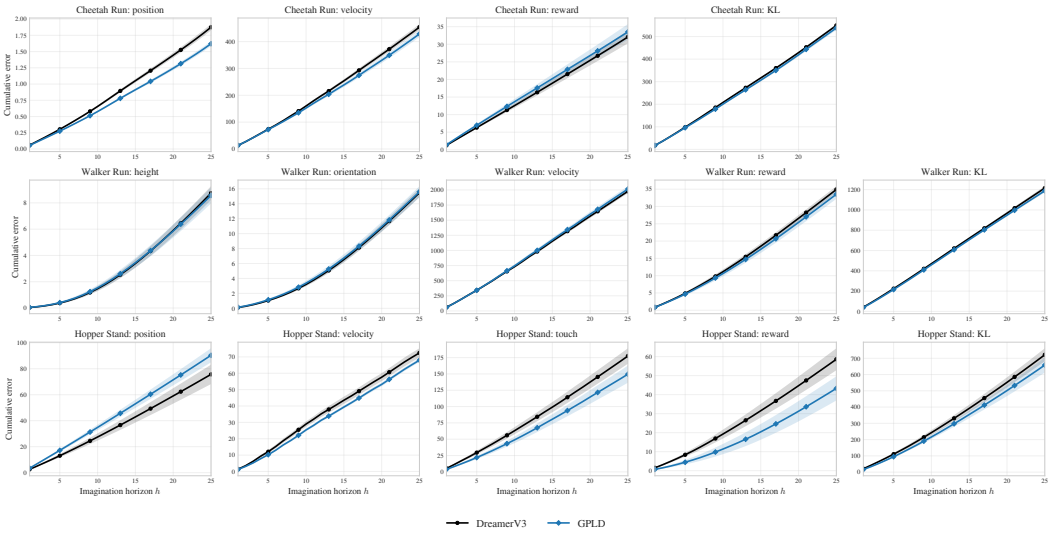


Figure 11: Cumulative open-loop error against the imagination horizon for every task and quantity, mean $\pm$ SEM over seeds.

N COMPARISON WITH OTHER REGULARIZERS

We compare GPLD with four regularizers that also target the input sensitivity of the posterior: spectral normalization (SN), virtual adversarial training (VAT), and the two random-perturbation regularizers of Appendix N.3, noise injection and noise consistency. All act on the posterior head only, as GPLD does, and are otherwise trained with the protocol of Section 5. Each comparator is run in its standard published form. GPLD's per-row probes (Appendix H) are a Monte Carlo device for an unchanged target: a joint probe estimates the same Jacobian energy, with larger variance. A per-row VAT, with one worst-case direction per categorical, would instead change the objective, from the largest curvature of the summed divergence to a sum of per-row largest curvatures, and so would be a different regularizer.

N.1 IMPLEMENTATION DETAILS AND SELECTION

Selection protocol. Each comparator's configuration was chosen by the best mean final return over the tuning tasks `cheetah-run`, `walker-run` and `hopper-stand` at five seeds. Ties ($|z| < 1$) were broken by open-loop error and then in favor of the weaker regularization, except where both tied configurations had also been run on all 12 unsaturated tasks: then the one with the higher normalized return across the 12 is reported. This decides between VAT's two smallest constant coefficients and between SN's two finalist caps, and in both cases it picks the configuration that does better against DreamerV3, the choice that makes the comparison harder for GPLD. GPLD's configuration was fixed earlier, from the ablations of Appendix K, and the comparators' procedure selects it as well. On the three tuning tasks the decaying coefficient is ahead of the constant one (mean final return 510.6 against 336.4), while $\lambda_0 = 0.25$ and 0.125, and $\rho = 0.5$ and 0.75, are tied within one standard error; run on more unsaturated tasks, the tied pairs are separated in favor of the values we use, $\rho = 0.5$ across eight tasks (normalized return 1.224 against 1.139) and $\lambda_0 = 0.25$ across six (1.303 against 1.271). Like the comparators, GPLD then uses a single configuration on every task, with no per-task tuning.

Spectral normalization. The posterior head maps u_t through a linear layer W_0, RMSNorm with gain g, a SiLU and an output layer W_1 to the logits. Every hidden linear layer in this architecture is followed by RMSNorm, which is invariant to the scale of its input, so normalizing the spectral norm of a hidden layer leaves the function unchanged. Halving or doubling the trained kernel of W_0 moves the posterior logits by less than 2×10^{-5} in relative terms. SN is therefore applied to W_1, the only layer whose scale reaches the logits, and its single hyperparameter is the cap c. Three further choices were needed.

- We use the projection $W \mapsto W \min(1, c/\sigma(W))$ (Gouk et al., 2021) rather than the division $W \mapsto cW/\sigma(W)$ of Miyato et al. (2018). At initialization $\sigma(W_1) \approx 2.4$, against roughly 110 after training, so division with a useful cap multiplies the initial weights by 8 to 50. At $c = 20$ the initial dynamics and representation losses are ten times their usual value.
- $\sigma(W_1)$ bounds nothing if the RMSNorm gain upstream of W_1 can grow, and trained DreamerV3 gains lie between 1.06 and 1.23. The gain is therefore clipped above at 1. No lower bound is imposed, because a smaller gain can only reduce sensitivity.
- σ is recomputed by 8 power iterations from a fixed start at every call. This recovers it to within 0.5% and adds no persistent vector that the optimizer would otherwise update.

The cap is absolute and shared across tasks, $c \in \{30, 60, 120\}$. Trained values of $\sigma(W_1)$ lie between about 110 and 140, so $c = 120$ is close to inactive. Under this construction each wrapped matrix has spectral norm at most c, which bounds the sensitivity of the logits to the features entering W_1. It does not make the head c-Lipschitz in u_t, because the normalization layer contributes a factor bounded only by its gain divided by its ϵ. Finally, W_1 is nearly rank one (stable rank about 1.1), so capping its spectral norm mostly rescales all logits together and acts partly as a softmax temperature.

Table 25 gives the three caps on the tuning tasks. Their aggregate final returns lie within one standard error of one another, so the data supports no ordering between them, and the apparent dip at $c = 60$ is within that noise. We report two caps: $c = 120$, which has the highest final return, and $c = 30$,

which has the highest nAUC. Across all 12 unsaturated tasks $c = 120$ has the higher normalized return (1.042 against 1.028), so it is the cap compared in Section 5.4.

Table 25: SN caps on the tuning tasks, five seeds each. Final return is the mean over the three tasks of each task's mean, $\pm$ the combined SEM. Normalized values are relative to DreamerV3 on the same task, averaged over tasks. The caps carried forward are $c = 30$ and $c = 120$.

Cap c	Final return	Normalized final	Normalized nAUC
30	651.6 ± 23.6	1.116	1.332
60	634.8 ± 37.0	1.081	1.210
120	662.2 ± 29.2	1.141	1.151

Virtual adversarial training. VAT penalizes the KL divergence between the posterior at u_t and at $u_t + r$, where r is the most damaging perturbation of radius ϵ found by power iteration (Miyato et al., 2019). We use $\epsilon = 0.1$ relative to the input, a radius of $0.1 \|u_t\|$, and one power iteration. The direction is found by a finite difference with step ξ rather than an exact Hessian-vector product: at a finite radius the KL is not quadratic, and the finite-difference probe finds a more adversarial direction. The best step tracks the radius, $\xi \approx 3\epsilon$, so ξ is tied to ϵ rather than tuned. The penalty is evaluated on the same fraction $\rho = 0.5$ of batch states as GPLD. Published VAT uses a constant coefficient, whereas GPLD decays its coefficient, so VAT was run under both schedules.

Table 26 gives the grid. With a constant coefficient, return peaks at $\lambda = 0.02$ and falls as λ grows to 0.25. With a decaying coefficient it is flat up to $\lambda = 0.1$, within one standard error, and then falls. The constant coefficients $\lambda = 0.01$ and 0.02 are tied on the tuning tasks (within one standard error) and well ahead of every other cell, which was not carried further. Both were run on all 12 unsaturated tasks, where $\lambda = 0.02$ has the higher normalized return (1.118 against 1.069, ahead on seven of the 12 tasks), so it is the VAT configuration we report. Table 27 shows that the schedule also decides whether VAT smooths or saturates. With a decaying coefficient, VAT lowers the probability-space energy far more than the logit-space energy, which stays close to DreamerV3's on `walker-run`. At $\lambda = 0.25$ with a constant coefficient, most posterior rows on `cheetah-run` are saturated.

Table 26: VAT on the tuning tasks, five seeds per cell. Columns as in Table 25. The selected cell is $\lambda = 0.02$ with a constant coefficient.

Coefficient	λ	Final return	Normalized final	Normalized nAUC
constant	0.01	682.6 ± 29.7	1.171	1.325
constant	0.02	693.7 ± 23.7	1.192	1.328
constant	0.05	609.1 ± 35.5	1.039	1.077
constant	0.1	507.3 ± 37.0	0.860	0.873
constant	0.25	401.9 ± 35.7	0.727	0.712
decayed	0.02	632.3 ± 33.5	1.099	1.226
decayed	0.05	633.5 ± 27.9	1.105	1.182
decayed	0.1	641.2 ± 36.4	1.108	1.122
decayed	0.25	543.6 ± 42.5	0.938	0.980

N.2 Results

Table 28 reports final return at 1M environment steps. Across the 12 unsaturated tasks GPLD's normalized return is 1.204 [1.117, 1.711], against 1.118 [1.021, 1.466] for VAT, 1.028 [0.957, 1.288] for SN at $c = 30$ and 1.042 [0.954, 1.307] at $c = 120$. On nAUC the normalized values are 1.207, 1.127, 1.060 and 1.030.

Table 29 gives nAUC per task, and Figure 12 the learning curves of all six arms on the 12 unsaturated tasks.

`quadruped`. SN was also run on both `quadruped` tasks at both caps, five seeds each (Table 30, Figure 13). GPLD has the highest nAUC on both tasks and the highest mean return at 3M; at 4M the arms that learn stably are close to the ceiling.

Table 27: Posterior entropy, saturated rows and Jacobian energy for VAT under both schedules at the 1M-step checkpoint, with DreamerV3 for reference. Measured as in Appendix L; mean ± SEM over seeds as in Table 21, five for each VAT cell.

Arm	cheetah-run	walker-run	hopper-stand
Normalized entropy			
DreamerV3	0.503 ± 0.005	0.273 ± 0.003	0.437 ± 0.006
$\lambda = 0.02$, constant	0.316 ± 0.012	0.317 ± 0.005	0.356 ± 0.010
$\lambda = 0.25$, constant	0.050 ± 0.004	0.248 ± 0.004	0.173 ± 0.008
$\lambda = 0.02$, decayed	0.436 ± 0.007	0.287 ± 0.004	0.421 ± 0.011
$\lambda = 0.1$, decayed	0.283 ± 0.030	0.305 ± 0.006	0.344 ± 0.016
Saturated rows (%)			
DreamerV3	1.6 ± 0.1	24.4 ± 0.5	9.1 ± 0.7
$\lambda = 0.02$, constant	8.6 ± 1.7	14.7 ± 0.8	10.7 ± 1.0
$\lambda = 0.25$, constant	90.8 ± 1.6	24.4 ± 0.8	43.0 ± 3.1
$\lambda = 0.02$, decayed	1.9 ± 0.2	21.3 ± 1.0	9.2 ± 0.9
$\lambda = 0.1$, decayed	9.4 ± 2.4	17.3 ± 1.2	11.5 ± 1.4

Arm	cheetah-run	walker-run	hopper-stand
Probability space, $\|J_\pi\|_F^2$			
DreamerV3	0.146 ± 0.0063	0.415 ± 0.014	0.237 ± 0.012
$\lambda = 0.02$, constant	0.0110 ± 0.00024	0.0653 ± 0.0057	0.0249 ± 0.0010
$\lambda = 0.25$, constant	0.000398 ± 0.000059	0.0184 ± 0.00031	0.00645 ± 0.00098
$\lambda = 0.02$, decayed	0.0447 ± 0.0031	0.221 ± 0.019	0.0818 ± 0.0080
$\lambda = 0.1$, decayed	0.0185 ± 0.00036	0.106 ± 0.0074	0.0380 ± 0.0040
Logit space, $\|J_\ell\|_F^2$			
DreamerV3	33.6 ± 1.9	177 ± 9.1	60.4 ± 4.1
$\lambda = 0.02$, constant	10.4 ± 3.0	83.4 ± 5.9	30.2 ± 4.8
$\lambda = 0.25$, constant	8.96 ± 1.2	99.8 ± 5.5	26.3 ± 7.7
$\lambda = 0.02$, decayed	23.7 ± 4.3	170 ± 10	46.9 ± 3.6
$\lambda = 0.1$, decayed	25.0 ± 9.0	120 ± 11	34.0 ± 4.9

Where final returns cannot separate the regularizers, the diagnostics can. Open-loop prediction favors GPLD over VAT: scored on trajectories that neither arm generated, GPLD's error is lower on ten of 14 task and quantity pairs, significantly on four and significantly higher on one, with the largest gains on cheetah-run and on hopper-stand position (Table 31). Against SN the open-loop errors are close: GPLD is lower on seven of 14 pairs against $c = 30$ (significantly on two, significantly higher on zero) and on seven against $c = 120$ (significantly on one, significantly higher on one). The smoothing diagnostics separate GPLD from both (Appendix L): VAT lowers the probability-space energy partly by sharpening the posterior, SN lowers the logit-space energy without making the probabilities less sensitive, and only GPLD lowers both without sharpening.

N.3 Random-perturbation regularizers

Two further comparators perturb the input of the posterior head with Gaussian noise whose standard deviation is σ times the input's root-mean-square. Noise consistency penalizes the KL divergence between the posterior at u_t and at the perturbed input, which makes it the random-direction counterpart of VAT. Noise injection trains on the perturbed input with no penalty term, which to first order is a Jacobian penalty (Bishop, 1995); the perturbed sample enters the recurrent state, so it also changes the inputs of the other losses. Both were selected by the protocol of Appendix N.1, from noise consistency $\lambda \in \{0.5, 1, 2, 4\}$ under both schedules at $\sigma = 0.1$ and from noise injection

Table 28: Final return at 1M environment steps, mean ± SEM over seeds. Normalized returns are relative to DreamerV3. Tuning tasks (†) use six seeds per arm, so their DreamerV3 and GPLD values differ from Table 1; the other tasks use the seeds of Table 1.

Task	DreamerV3	GPLD	VAT	SN 30	SN 120
cheetah-run†	638.2 ±32.8	751.3 ±26.7	706.1 ±18.0	644.1 ±20.5	682.9 ±20.0
walker-run†	415.2 ±64.9	630.2 ±39.9	592.2 ±38.8	583.0 ±46.4	607.3 ±15.3
hopper-stand†	642.4 ±79.8	729.7 ±57.8	761.9 ±54.5	753.6 ±38.5	718.1 ±70.1
acrobot-swingup	268.7 ±6.1	319.9 ±20.7	297.6 ±13.3	290.0 ±15.0	309.0 ±18.0
acrobot-swingup-sparse	91.9 ±10.0	124.4 ±14.5	97.0 ±19.9	81.9 ±20.4	73.9 ±15.7
cartpole-swingup-sparse	789.6 ±15.9	791.6 ±19.4	773.7 ±11.4	751.9 ±29.2	765.5 ±10.4
finger-spin	614.9 ±44.3	732.9 ±57.3	671.4 ±55.6	555.6 ±8.0	639.4 ±68.2
finger-turn-easy	891.4 ±25.1	928.0 ±17.8	871.4 ±27.0	892.5 ±35.5	882.4 ±37.5
finger-turn-hard	901.5 ±18.3	891.8 ±34.2	808.8 ±37.0	882.3 ±34.5	857.9 ±38.2
hopper-hop	99.0 ±43.1	171.1 ±16.2	143.6 ±48.4	91.1 ±29.6	81.5 ±49.7
pendulum-swingup	762.5 ±22.8	798.5 ±32.0	822.3 ±14.4	789.9 ±25.3	816.1 ±8.6
walker-walk	877.6 ±50.0	937.8 ±9.4	929.1 ±7.1	869.5 ±53.9	922.7 ±8.7
All 12 unsaturated tasks, normalized final return		1.204	1.118	1.028	1.042
All 12 unsaturated tasks, normalized nAUC		1.207	1.127	1.060	1.030

Table 29: nAUC over 10k–1M environment steps on the return scale, mean ± SEM over seeds; seeds, tuning tasks (†) and aggregates as in Table 28.

Task	DreamerV3	GPLD	VAT	SN 30	SN 120
cheetah-run†	472.1 ±15.1	512.9 ±14.1	485.1 ±16.6	463.8 ±16.1	459.3 ±16.8
walker-run†	221.4 ±30.6	393.7 ±21.8	346.8 ±26.4	353.5 ±29.8	364.2 ±17.3
hopper-stand†	309.8 ±66.6	369.7 ±76.5	458.1 ±69.9	451.7 ±21.8	332.7 ±89.6
acrobot-swingup	180.2 ±8.1	210.7 ±4.2	199.7 ±10.8	193.2 ±8.0	189.0 ±6.4
acrobot-swingup-sparse	34.3 ±2.9	51.4 ±11.3	43.3 ±9.2	35.8 ±16.2	25.5 ±4.5
cartpole-swingup-sparse	572.0 ±51.3	616.7 ±23.5	576.5 ±20.3	585.7 ±40.6	532.6 ±27.9
finger-spin	525.2 ±24.7	567.0 ±60.6	593.3 ±43.1	494.1 ±6.2	545.2 ±51.9
finger-turn-easy	776.9 ±12.3	781.0 ±13.5	727.8 ±18.3	764.7 ±21.5	766.1 ±22.0
finger-turn-hard	641.1 ±25.1	609.1 ±36.6	566.7 ±20.8	624.0 ±16.6	628.1 ±21.2
hopper-hop	46.1 ±20.0	69.2 ±21.1	49.7 ±17.7	30.6 ±12.5	46.6 ±28.6
pendulum-swingup	746.0 ±9.9	769.5 ±9.6	698.4 ±44.2	769.3 ±8.4	745.0 ±10.7
walker-walk	655.7 ±70.2	731.0 ±31.2	728.8 ±33.5	619.5 ±65.1	608.1 ±57.0
All 12 unsaturated tasks, normalized final return		1.204	1.118	1.028	1.042
All 12 unsaturated tasks, normalized nAUC		1.207	1.127	1.060	1.030

$\sigma \in \{0.03, 0.1, 0.3\}$; the selected configurations are noise consistency with $\lambda = 1$ and a constant coefficient, and noise injection with $\sigma = 0.1$.

Table 32 reports their final return on the 12 unsaturated tasks, Table 33 their nAUC, and Figure 12 their learning curves. GPLD does better than both. Its normalized return across the 12 tasks is 1.204, against 1.110 for noise consistency and 1.064 for noise injection (nAUC: 1.207, 1.148 and 1.080). GPLD is ahead by more than the sum of the standard errors on four tasks against noise consistency and five against noise injection, and behind on none. Their smoothing is discussed in Appendix L.

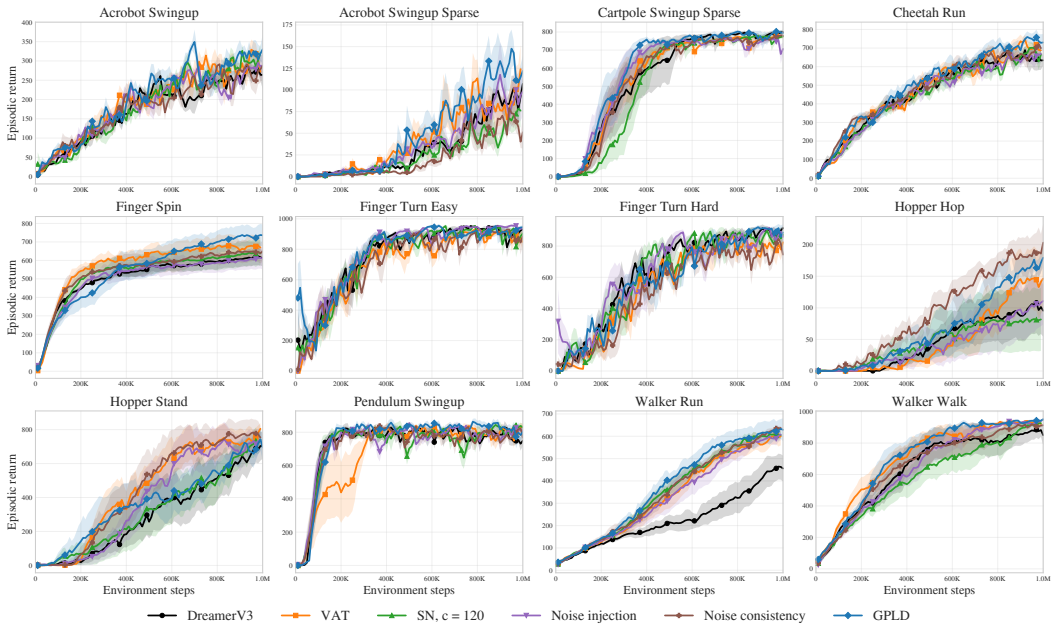


Figure 12: Learning curves of DreamerV3, VAT, SN ($c = 120$), noise injection, noise consistency and GPLD on the 12 unsaturated proprioceptive tasks. Mean $\pm$ SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.3$).

Table 30: `quadruped` with SN at both caps, five seeds per arm. Return at 3M and 4M is the mean of evaluations over the preceding 100k steps; nAUC covers 0–4M and is normalized by DreamerV3's mean final return. Mean $\pm$ SEM over seeds; bold marks an arm better than DreamerV3 by more than the sum of their standard errors.

Task	Metric	DreamerV3	GPLD	SN, $c = 30$	SN, $c = 120$
`quadruped-run`	Mean return, 3M	637.3 ± 134.1	$\mathbf{857.0 \pm 20.2}$	480.8 ± 174.7	729.3 ± 93.5
	Mean return, 4M	899.2 ± 15.5	913.2 ± 7.0	890.1 ± 18.5	899.3 ± 22.4
	nAUC, 0–4M	0.453 ± 0.047	$\mathbf{0.562 \pm 0.045}$	0.397 ± 0.060	0.406 ± 0.027
`quadruped-walk`	Mean return, 3M	516.5 ± 158.1	$\mathbf{845.9 \pm 26.7}$	509.6 ± 175.6	576.1 ± 195.1
	Mean return, 4M	783.8 ± 150.0	912.8 ± 35.0	912.1 ± 20.2	775.4 ± 115.8
	nAUC, 0–4M	0.450 ± 0.067	$\mathbf{0.598 \pm 0.042}$	0.521 ± 0.097	0.513 ± 0.091

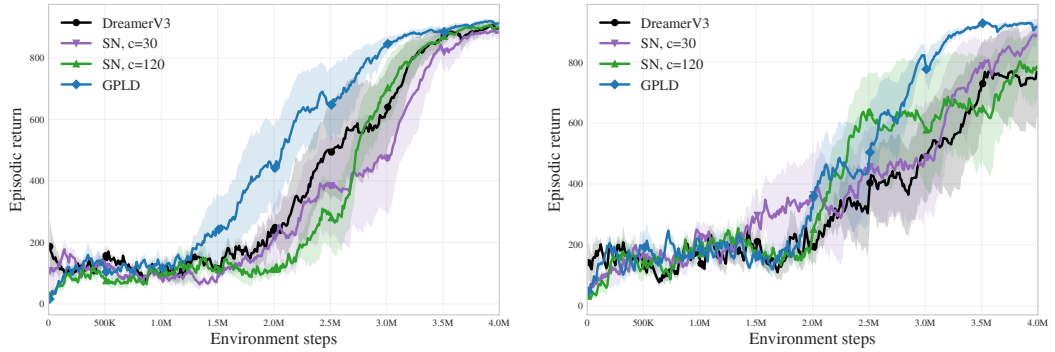


Figure 13: `quadruped-run` (left) and `quadruped-walk` (right) with SN at both caps. Mean $\pm$ SEM over seeds, smoothed with an exponential moving average ($\alpha = 0.1$).

Table 31: Cumulative open-loop error at $h = 25$ for GPLD and each comparator, mean $\pm$ SEM over seeds. Each pair is scored only on windows from source policies that neither arm of the pair generated, so each pair has its own evaluation data: values compare within a pair, not across pairs, and GPLD's values differ between pairs. Lower is better; bold marks the arm of a pair whose error is lower than the other's by more than the sum of their standard errors.

		GPLD vs VAT		GPLD vs SN, $c = 30$		GPLD vs SN, $c = 120$	
Task	Quantity	GPLD	VAT	GPLD	SN	GPLD	SN
cheetah-run	position	$\mathbf{2.735 \pm 0.055}$	3.292 ± 0.14	$\mathbf{2.339 \pm 0.046}$	2.625 ± 0.13	2.339 ± 0.046	2.359 ± 0.067
	velocity	$\mathbf{417.5 \pm 11}$	472.5 ± 14	$\mathbf{416.8 \pm 11}$	492.9 ± 33	416.8 ± 11	435.0 ± 8.8
	reward	$\mathbf{31.53 \pm 2.0}$	39.33 ± 3.4	31.64 ± 2.0	38.58 ± 5.5	31.64 ± 2.0	29.43 ± 1.5
	KL	$\mathbf{529.3 \pm 8.3}$	608.1 ± 22	532.6 ± 8.5	$\mathbf{509.0 \pm 15}$	$\mathbf{532.6 \pm 8.5}$	583.2 ± 11
walker-run	height	8.291 ± 0.51	8.259 ± 0.76	8.474 ± 0.50	$\mathbf{7.235 \pm 0.51}$	8.474 ± 0.50	7.648 ± 0.42
	orientation	14.83 ± 0.58	15.04 ± 1.1	15.26 ± 0.61	13.97 ± 0.91	15.26 ± 0.61	14.22 ± 0.51
	velocity	2018 ± 45	1954 ± 50	2002 ± 41	1971 ± 58	2002 ± 41	1919 ± 64
	reward	28.05 ± 1.2	28.19 ± 1.3	30.54 ± 1.1	28.35 ± 2.3	30.54 ± 1.1	28.77 ± 1.7
	KL	1141 ± 17	$\mathbf{1055 \pm 35}$	1162 ± 15	1193 ± 27	1162 ± 15	1176 ± 19
hopper-stand	position	$\mathbf{240.0 \pm 9.3}$	282.8 ± 14	187.8 ± 6.3	180.4 ± 7.9	187.8 ± 6.3	$\mathbf{159.1 \pm 9.0}$
	velocity	$\mathbf{130.5 \pm 1.6}$	135.7 ± 2.8	102.2 ± 1.4	105.8 ± 2.7	102.2 ± 1.4	100.8 ± 1.8
	touch	155.1 ± 8.5	158.9 ± 12	145.1 ± 11	157.1 ± 13	145.1 ± 11	159.5 ± 16
	reward	41.54 ± 4.3	43.04 ± 5.6	41.42 ± 5.7	47.32 ± 5.9	41.42 ± 5.7	52.22 ± 10
	KL	731.3 ± 37	$\mathbf{639.9 \pm 46}$	678.5 ± 40	626.6 ± 48	678.5 ± 40	762.2 ± 77

Table 32: Random-perturbation regularizers: final return at 1M environment steps, mean $\pm$ SEM over seeds; seeds, tuning tasks (†) and aggregates as in Table 28.

Task	DreamerV3	GPLD	Noise inj.	Noise cons.
cheetah-run†	638.2 $\pm$32.8	751.3 $\pm$26.7	658.1 $\pm$39.5	689.8 $\pm$29.5
walker-run†	415.2 $\pm$64.9	630.2 $\pm$39.9	600.2 $\pm$26.7	631.2 $\pm$9.7
hopper-stand†	642.4 $\pm$79.8	729.7 $\pm$57.8	721.2 $\pm$72.1	769.2 $\pm$78.2
acrobot-swingup	268.7 $\pm$6.1	319.9 $\pm$20.7	282.6 $\pm$13.7	275.9 $\pm$26.6
acrobot-swingup-sparse	91.9 $\pm$10.0	124.4 $\pm$14.5	89.9 $\pm$27.3	58.8 $\pm$9.8
cartpole-swingup-sparse	789.6 $\pm$15.9	791.6 $\pm$19.4	737.1 $\pm$16.7	773.6 $\pm$11.9
finger-spin	614.9 $\pm$44.3	732.9 $\pm$57.3	608.6 $\pm$56.8	645.8 $\pm$54.9
finger-turn-easy	891.4 $\pm$25.1	928.0 $\pm$17.8	944.0 $\pm$5.5	886.8 $\pm$32.1
finger-turn-hard	901.5 $\pm$18.3	891.8 $\pm$34.2	885.1 $\pm$38.1	761.9 $\pm$26.3
hopper-hop	99.0 $\pm$43.1	171.1 $\pm$16.2	103.9 $\pm$38.8	191.5 $\pm$29.1
pendulum-swingup	762.5 $\pm$22.8	798.5 $\pm$32.0	801.4 $\pm$17.6	775.8 $\pm$22.3
walker-walk	877.6 $\pm$50.0	937.8 $\pm$9.4	937.7 $\pm$6.6	909.2 $\pm$12.3
All 12 unsaturated tasks, normalized final return		1.204	1.064	1.110
All 12 unsaturated tasks, normalized nAUC		1.207	1.080	1.148

Table 33: Random-perturbation regularizers: nAUC over 10k–1M environment steps on the return scale, mean $\pm$ SEM over seeds; seeds, tuning tasks (†) and aggregates as in Table 28.

Task	DreamerV3	GPLD	Noise inj.	Noise cons.
cheetah-run†	472.1 $\pm$15.1	512.9 $\pm$14.1	461.0 $\pm$12.5	490.7 $\pm$22.6
walker-run†	221.4 $\pm$30.6	393.7 $\pm$21.8	340.3 $\pm$29.4	362.3 $\pm$19.5
hopper-stand†	309.8 $\pm$66.6	369.7 $\pm$76.5	398.6 $\pm$69.1	462.5 $\pm$81.8
acrobot-swingup	180.2 $\pm$8.1	210.7 $\pm$4.2	190.2 $\pm$6.8	184.8 $\pm$10.5
acrobot-swingup-sparse	34.3 $\pm$2.9	51.4 $\pm$11.3	37.6 $\pm$10.1	22.1 $\pm$3.0
cartpole-swingup-sparse	572.0 $\pm$51.3	616.7 $\pm$23.5	590.1 $\pm$24.3	569.3 $\pm$23.3
finger-spin	525.2 $\pm$24.7	567.0 $\pm$60.6	520.9 $\pm$33.7	559.7 $\pm$50.9
finger-turn-easy	776.9 $\pm$12.3	781.0 $\pm$13.5	793.6 $\pm$16.4	732.2 $\pm$26.3
finger-turn-hard	641.1 $\pm$25.1	609.1 $\pm$36.6	636.9 $\pm$24.6	528.9 $\pm$35.8
hopper-hop	46.1 $\pm$20.0	69.2 $\pm$21.1	43.5 $\pm$22.9	96.0 $\pm$13.7
pendulum-swingup	746.0 $\pm$9.9	769.5 $\pm$9.6	750.9 $\pm$12.8	739.8 $\pm$13.5
walker-walk	655.7 $\pm$70.2	731.0 $\pm$31.2	669.5 $\pm$33.9	677.0 $\pm$52.3
All 12 unsaturated tasks, normalized final return		1.204	1.064	1.110
All 12 unsaturated tasks, normalized nAUC		1.207	1.080	1.148

O SEED RELIABILITY

Aggregates at a handful of seeds establish a direction of effect but cannot separate individual task differences from run-to-run variation, so every task with at least 9 seeds completed by both arms at $\lambda_0 = 0.25$ was also tested per seed (Table 34). The threshold decides which tasks appear, not their results. On the proprioceptive benchmark, `walker-run` at $n = 22$ gives a paired mean difference of $+155.8$ ($t = 3.96$, $p < 0.001$, $d_z = 0.85$, GPLD ahead on 18 of 22 seeds), and `cheetah-run` at $n = 19$ gives $+64.2$ ($t = 3.57$, $p = 0.002$, $d_z = 0.82$, ahead on 16 of 19). Of the five `dog` and `humanoid` tasks with enough seeds, the gain is significant on `dog-stand` and `dog-walk` and not on `dog-run`, `humanoid-run`, and `humanoid-walk`. At these seed counts the large-embodiment result is established in aggregate rather than task by task.

All tests are paired per seed with Student-t intervals, and final return is the mean of evaluations over the last 100k steps, as in Table 1. Seeds are paired by index. The column r is the correlation of per-seed final return between the two arms. Pairing reduces the variance of the difference when $r > 0$ and inflates it when $r < 0$, so on the two tasks with negative r the paired test is the conservative choice.

With 9 to 22 seeds the tests detect the gains on the two proprioceptive tasks and on `dog-stand` and `dog-walk`. On `dog-run`, `humanoid-run`, and `humanoid-walk` they cannot separate a small true gain from none. The main-text tables use a handful of seeds per task, so their per-task entries are estimates with wide intervals; the aggregate over tasks is the claim they support.

Table 34: Paired per-seed tests on every task with at least 9 seeds completed by both arms. Paired diff. is the mean over seeds of GPLD's minus DreamerV3's final return; per-seed gain is the mean $\pm$ SEM of the per-seed relative difference; t and p are from a paired Student-t test; r is the across-arm correlation of per-seed final return.

Task	Horizon	n	Paired diff.	Per-seed gain	t	p	r
`walker-run`	1M	22	$+155.8$	$+56.5 \pm 16.9\%$	3.96	< 0.001	$+0.10$
`cheetah-run`	1M	19	$+64.2$	$+10.6 \pm 3.0\%$	3.57	0.002	$+0.32$
`dog-run`	12M	12	$+19.9$	$+6.7 \pm 6.7\%$	0.70	0.498	$+0.22$
`dog-stand`	8M	11	$+139.7$	$+26.8 \pm 9.6\%$	2.55	0.029	-0.37
`dog-walk`	12M	10	$+291.7$	$+91.0 \pm 32.3\%$	4.11	0.003	-0.20
`humanoid-run`	12M	9	$+19.2$	$+12.5 \pm 8.6\%$	0.89	0.399	$+0.04$
`humanoid-walk`	12M	12	-4.5	$+0.8 \pm 4.9\%$	-0.14	0.889	$+0.25$