RG-TAIL: RELIABILITY-GATED TRUST-AWARE IMAGINATION LEARNING UNDER LIMITED BUDGETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-based reinforcement learning (MBRL) improves sample efficiency by leveraging learned world models to generate synthetic experience. However, model-generated transitions can be harmful when model errors accumulate especially in data-scarce settings, where reliable synthetic experience is essential due to the high cost of real-world interaction. Our large-scale empirical study, *Real-or-Imagined* (RoI), reveals that while synthetic data can substantially improve reinforcement learning, uncertainty estimates do not necessarily align with the long-term utility of synthetic transitions. Based on this observation, we propose *Reliability-Gated Trust-Aware Imagination Learning* (**RG-TAIL**), a framework that combines transition-level trust estimation with a higher-level reliability gate to determine whether synthetic experience could be used in policy optimization. We perform extensive experiments across diverse environments, reinforcement learning algorithms, model configurations, and data budgets. Compared with internal baselines and representative MBRL methods, RG-TAIL achieves the best overall performance under both extremely low- and high-budget settings, improving task success rates by approximately **+19%** on average.

1 INTRODUCTION

Since the emergence of model-based reinforcement learning (MBRL), it has gradually become a widely adopted paradigm for improving sample efficiency in reinforcement learning (Moerland et al., 2023). By learning an approximate environment dynamics model in the form of a state transition function, world models (WM) naturally enable the generation of synthetic experiences for policy optimization (Ha & Schmidhuber, 2018). However, recent studies have shown that model-generated transitions can sometimes harm policy learning due to accumulated model errors. Existing works attempt to incorporate uncertainty estimation from probabilistic world models (PWM) to regulate downstream policy optimization (Chua et al., 2018; Yu et al., 2020; Kidambi et al., 2020).

Nevertheless, uncertainty fundamentally characterize how confidently the world model predicts a transition, rather than how beneficial that transition is for downstream policy optimization (Wei et al., 2024). This introduces an inherent gap between model fidelity and the long-term utility of synthetic experience, which becomes increasingly important in real-world applications where collecting large amounts of interaction data is often expensive, time-consuming, or even unsafe (Zitkovich et al., 2023; Chi et al., 2025). Therefore, developing reliable methods to exploit synthetic experience under limited real-world data budgets is critical. However, systematic investigations into synthetic-data utilization under extremely limited budgets remain relatively underexplored.

Research Questions. Motivated by these observations, we seek to answer the following questions: **(i)** Under low-data-budget settings, can synthetic experience remain reliable and beneficial for reinforcement learning? **(ii)** Is there a supervision signal more aligned with policy optimization than model uncertainty? **(iii)** Can environment-level adaptive utilization of synthetic experience further improve policy optimization?

To answer these questions, we propose *Reliability-Gated Trust-Aware Imagination Learning* (**RG-TAIL**), a framework that estimates transition-level trustworthiness from held-out real-world prediction errors, and introduces a gating mechanism based on world-model agreement and rollout consistency to determine whether synthetic experience should participate in policy optimization.

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.

The **contributions** of this paper are summarized as follows: **(i)** Through a large-scale empirical study, *Real-or-Imagined* (RoI), we demonstrate that synthetic experience can substantially improve reinforcement learning under limited real-data budgets, but uncertainty estimates are not always ideal supervision signals for synthetic transition utilization and do not necessarily reflect the long-term utility of synthetic experience for policy optimization. **(ii)** We propose RG-TAIL, and evaluate it against both internal baselines (RoI and TAIL) and representative model-based reinforcement learning methods (MBPO, TD-MPC2, and DreamerV3), demonstrating that RG-TAIL achieves superior average performance and task success rates under both extremely low- and high-budget settings across diverse control environments and reinforcement-learning algorithms (Janner et al., 2019; Hansen et al., 2024; Hafner et al., 2024). **(iii)** We perform extensive mechanistic analyses, including component-wise validation of RG-TAIL, detailed failure case studies, and budget scaling analysis, providing practical insights into when synthetic experience should or should not be utilized, further improving the practical value of the proposed framework.

2 Related Work

Model-Based Reinforcement Learning. Early MBRL methods established the effectiveness of learning environment dynamics for planning and synthetic experience generation, including probabilistic model-predictive control, latent-space planning, and short-horizon model rollouts (Moerland et al., 2023; Chua et al., 2018; Hafner et al., 2019; Janner et al., 2019). More advanced world-model approaches subsequently shifted toward scalable latent dynamics and imagination-based control, with DreamerV3 and TD-MPC2 demonstrating strong performance across diverse continuous-control domains (Hafner et al., 2024; Hansen et al., 2024). Recent work has further explored richer world-model representations and more scalable policy extraction, including discrete codebook dynamics for continuous control, multi-task world models optimized through first-order policy learning, and event-aware representations for improving generalization across visually diverse environments (Scannell et al., 2025; Georgiev et al., 2025; Peng et al., 2026).

Reliable Synthetic Data Utilization. A parallel line of work has emphasized that model-generated experience can become harmful when prediction errors accumulate or when policies exploit inaccurate regions of the learned dynamics. Existing methods address this issue through uncertainty-aware planning, pessimistic reward or value estimation, conservative policy optimization, and adaptive rollout truncation (Chua et al., 2018; Yu et al., 2020; Kidambi et al., 2020). In particular, COMBO mitigates model bias through conservative value estimation over model-generated out-of-support transitions, without relying on explicit uncertainty estimates (Yu et al., 2021). More recent approaches explicitly control model uncertainty or rollout error, including NUNO (Koprulu et al., 2025), Infoprop (Frauenknecht et al., 2025), and WIMLE (Aghabozorgi et al., 2026). Additionally, FTFL studies model pathologies that degrade synthetic-data quality in MBPO and improves reliability through normalization and prediction-design choices (Barkley & Fridovich-Keil, 2026). Unlike these approaches, RG-TAIL distinguishes local transition-level trust from policy-conditioned global reliability, rather than relying on a single uncertainty signal to regulate synthetic experience.

3 Methodology

3.1 Trust-Aware Design

We first propose *Trust-Aware Imagination Learning* (**TAIL**), which later serves as the foundation of the proposed RG-TAIL framework, and one of the internal baselines during evaluation.

3.1.1 Trust Module

TAIL introduces a supervised trust module trained directly from real-world prediction errors. It receives the current state s_t, action a_t, and predicted next state $\hat{s}_{t+1}$ as input, with the predicted variance σ^2_{t+1} additionally provided for PWM:

$$\tau = \text{sigmoid}\Big(\text{MLP}([s_t, a_t, \hat{s}_{t+1}, \sigma^2_{t+1}\ \text{(PWM only)}])\Big). \tag{1}$$

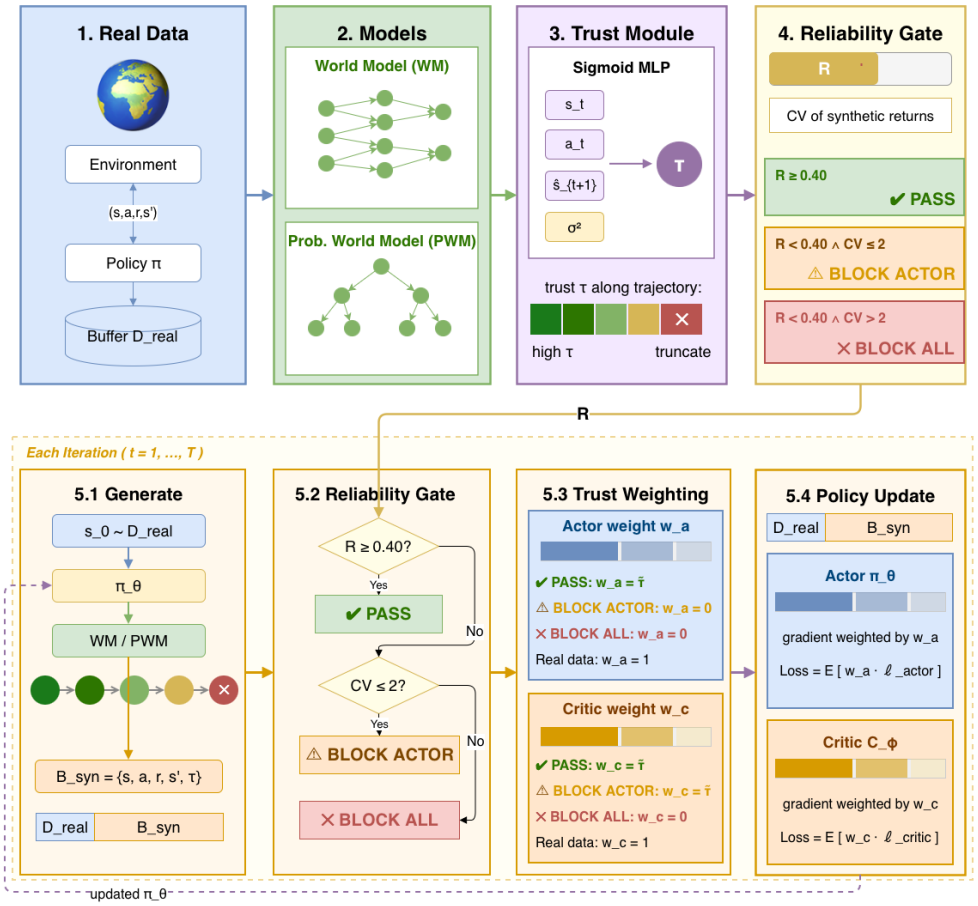


Figure 1: Overall architecture of RG-TAIL.

The supervision target, referred to as trust score, is defined as an exponentially decaying function of the one-step prediction error, where σ_{err} denotes the median measured on held-out real transitions:

$$\tau_{\text{target}} = \exp\left(-\frac{\left\|\hat{s}_{t+1} - s^{\text{real}}_{t+1}\right\|}{\sigma_{\text{err}}}\right). \tag{2}$$

The trust module is trained once on held-out real transitions, frozen thereafter, and applied only as trust weighting for synthetic transitions, with real transitions always assigned unit weight.

3.1.2 Trust Score Utilization

Once trained, the trust score is incorporated into imagined learning through three mechanisms:

• **Trust-stratified start-state sampling.** Imagined rollouts are preferentially initialized from real states with higher trust scores, concentrating synthetic experience in regions where the world model exhibits more reliable one-step predictions.

• **Trust-weighted actor update.** Synthetic transitions are weighted during actor optimization. Given B as the batch size and $\tau_{\min}$ as the minimum weighting coefficient, when the trust-score variance within a batch exceeds a small threshold, the normalized weight for transition i is

$$\tilde{\tau}_i = \text{clip}\left(\frac{\tau_i}{\bar{\tau}}, \tau_{\min}, 1\right), \qquad \bar{\tau} = \frac{1}{B}\sum_{j=1}^{B}\tau_j. \tag{3}$$

Otherwise, uniform weighting is used. The resulting actor objective is

$$L_{\text{actor}} = \mathbb{E}_i\left[\tilde{\tau}_i\, \ell_i^{\text{actor}}\right], \tag{4}$$

where ℓ_i^{actor} denotes the standard actor loss of PPO, SAC, or TD3. Real transitions always receive unit weight, while the critic remains unweighted in TAIL.

• **Trust-gated trajectory termination.** Imagined rollouts terminate when the trust score falls below the calibrated threshold $\tau_{\min}$, preventing further propagation through increasingly unreliable model predictions. For PPO, the termination point is incorporated into advantage estimation; for SAC and TD3, only transitions generated before termination are retained.

3.2 Reliability-Gated Design

Building upon TAIL, we further introduce an additional **Reliability Gate** that operates at a higher level than trust weighting. Trust determines how much a synthetic transition should influence learning, whereas reliability determines whether synthetic information should participate in learning at all. The complete RG-TAIL procedure is demonstrated in Figure 1 and summarized in Algorithm 1.

3.2.1 Reliability Definition

We define a static global reliability score as

$$\tau_{\text{disagree}} = \exp\left(-\beta \frac{\|s_{\text{WM}} - \mu_{\text{PWM}}\|^2}{d}\right), \tag{5}$$

$$\text{reliability} = \text{clip}\left(\tau_{\text{disagree}} \cdot \frac{\textit{avg_traj_len}}{H}, 0, 1\right). \tag{6}$$

Here s_{WM} denotes the WM prediction, μ_{PWM} denotes the PWM mean prediction, d is the state dimension, avg_traj_len is the mean realized trajectory length, and H denotes the rollout horizon.

The disagreement term measures predictive consistency between the WMs and PWMs. The trajectory-length term captures error accumulation during imagined rollouts. For each environment-algorithm configuration, reliability is computed once during the initial calibration stage and remains fixed throughout training. We intentionally adopt static reliability as the default design, since dynamic recalibration provides only marginal improvement while introducing additional computational overhead, as further explored and compared in Appendix D.

3.2.2 Reliability Gate

In addition, we compute the coefficient of variation of undiscounted cumulative returns R_{syn} of synthetic episodes:

$$\textit{CV} = \frac{\sigma(R_{\text{syn}})}{|\mu(R_{\text{syn}})| + \epsilon}.$$

Let w_i^a and w_i^c denote the actor and critic trust weights, and the reliability gate operates as follows:

$$(\textit{mode}, w_i^a, w_i^c) = \begin{cases} (\texttt{pass},\ w_i^a,\ w_i^c), & \text{reliability} \geq 0.40, \\ (\texttt{block_actor},\ 0,\ w_i^c), & \text{reliability} < 0.40 \wedge \textit{CV} \leq 2.0, \\ (\texttt{block_all},\ 0,\ 0), & \text{reliability} < 0.40 \wedge \textit{CV} > 2.0. \end{cases} \tag{7}$$

Reliability measures the overall trustworthiness of the learned world model within an environment, whereas CV characterizes the stability of imagined rollouts. When reliability is low, RG-TAIL further examines CV. A low CV suggests that the generated returns remain sufficiently consistent for value learning, leading to the `block_actor` mode, where only trust-weighted critic updates are retained, since policy gradients are generally more sensitive to model bias than value estimation (Moerland et al., 2023). In contrast, a high CV indicates unstable rollouts that may contaminate both actor and critic learning, triggering `block_all`.

4 EXPERIMENTAL SETUP

We define the scope as a unified benchmark spanning 6 environments of varying complexity, 3 reinforcement learning algorithms, 5 synthetic-data utilization strategies, and 6 real-data budgets.

• **Benchmark Environments.** The benchmark consists of six environments of varying complexity: `LunarLander-v3`, `MountainCarContinuous-v0`, `Hopper-v5`, `HalfCheetah-v5`, `Walker2d-v5`, and `Swimmer-v5` (Towers et al., 2025; Todorov et al., 2012).

• **Algorithms and Configurations.** We evaluate three reinforcement learning (RL) algorithms: PPO, SAC, and TD3 (Schulman et al., 2017; Haarnoja et al., 2018; Fujimoto et al., 2018). `LunarLander-v3` is the only discrete-action environment, and is therefore not applicable to TD3. For each algorithm, we consider five learning configurations: **real-only**, **imagined-WM**, **imagined-PWM**, **hybrid-WM**, and **hybrid-PWM**. The imagined variants train exclusively on synthetic experience generated by either a deterministic world model (WM) or a probabilistic world model (PWM), while the hybrid variants combine real and synthetic experience (Ha & Schmidhuber, 2018).

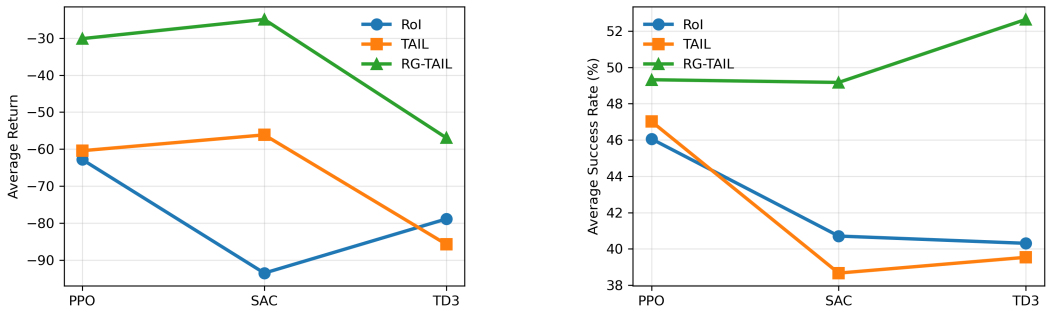
• **Data Collection Protocol.** We define six real-data budgets: **5K**, **10K**, **20K**, **50K**, **100K**, and **500K** transitions, covering from extreme scarcity to rich settings. For each budget, real transitions are collected independently using an online PPO policy and are subsequently used to train the WM and PWM. For each budget, the WM/PWM generates algorithm-specific synthetic rollouts using the corresponding actor, with a fixed **10:1** synthetic-to-real transition ratio. All budgets are evaluated and reported in Appendix H, and all results are averaged over 10 random seeds. The most constrained **5K** budget serves as the primary benchmark in the main paper unless otherwise stated. We additionally include **10K** results for external baseline comparisons for robustness verification.

• **Evaluation Criteria.** Since return alone may not fully reflect task completion, we additionally report success rate (SR). Across all tasks, an episode is considered successful if its return is greater than zero. However, a near-zero average return does not necessarily indicate successful learning, as an agent may converge to a trivial stationary policy while achieving a 0% success rate. Such anomalous cases are annotated with * in the tables. All reported ΔSR are expressed in absolute percentage points rather than relative percentage changes. Accordingly, the highlighted best-performing results are determined by jointly considering both return and SR, rather than return alone.

5 RESULTS AND ANALYSIS

5.1 REAL OR IMAGINED? EMPIRICAL STUDY

We first conduct a preliminary empirical study, namely *Real-or-Imagined* (RoI), with detailed 5K results in Table 10. It reveals that hybrid training outperforms both pure real-data and pure imagined learning, and WMs consistently outperform PWMs despite lacking explicit uncertainty estimation. This observation further validates the research gap that model uncertainty does not necessarily reflect the long-term utility of synthetic transitions for policy optimization (Wei et al., 2024).



(a) Average return.

(b) Average success rate.

Figure 2: Comparison of RoI, TAIL, and RG-TAIL across PPO, SAC, and TD3 under the 5K real-data budget, averaged across all applicable environments, configurations, and 10 random seeds.

5.2 Comparison with Internal Baselines

Comparing RG-TAIL with RoI and TAIL, we first examine each RL algorithm separately, and per-environment results are shown in Table 11 in Appendix. As summarized in Figure 2, RG-TAIL achieves the best average return and success rate across PPO, SAC, and TD3, with detailed per-algorithm results shown in Table 11. Combining three algorithms, Tables 1 and 2 show that under 5K and 10K budget respectively, RG-TAIL outperforms RoI on 6/6 and 5/6 environments, achieving overall improvements of approximately **+44** and **+52**. Compared with TAIL, RG-TAIL wins on 5/6 and 5/6 environments and achieves overall improvements of approximately **+24** and **+54**.

5.3 Comparison with External Baselines

We compare against three external baselines: **MBPO**, a classical MBRL framework based on short model rollouts (Janner et al., 2019); **TD-MPC2**, a recent SOTA method that combines latent predictive modeling with trajectory optimization (Hansen et al., 2024); and **DreamerV3**, a structurally distinct yet widely adopted latent world-model approach (Hafner et al., 2024).

As shown in Table 1, MBPO and DreamerV3 achieve the strongest performance on Hopper and HalfCheetah respectively. On the remaining environments, RG-TAIL consistently demonstrates superior performance under 5K setting. RG-TAIL outperforms all external baselines in average SR under both 5K and 10K budgets, and the advantage is more pronounced under 10K budget as shown in Table 2, achieving an overall improvement of approximately **+10%** compared with the strongest baseline DreamerV3. These results indicate that RG-TAIL remains highly effective under limited-data conditions and generalizes well across diverse environments.

Table 1: Return (SR%) under the 5K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	-324.54 (2.8)	-177.46 (3.8)	-256.19 (3.5)	-865.00 (0.1)	-183.85 (1.4)	-425.64 (0.8)
MountainCar	-28.93 (3.9)	-31.25 (6.1)	-26.88 (10.9)	-0.00 (0.0)*	-0.63 (0.0)*	-0.00 (0.0)*
Hopper	74.79 (90.0)	88.86 (91.7)	115.43 (100.0)	135.88 (100.0)	72.70 (100.0)	90.93 (100.0)
HalfCheetah	-336.30 (1.9)	-370.11 (1.6)	-307.90 (0.0)	-329.29 (0.1)	-2.98 (0.0)*	-0.33 (32.7)
Walker2d	39.97 (53.6)	34.03 (46.0)	161.93 (73.4)	42.89 (40.0)	41.56 (100.0)	40.89 (100.0)
Swimmer	20.23 (89.5)	20.79 (89.3)	25.18 (98.0)	0.24 (43.2)	-0.14 (48.4)	-0.83 (42.2)
Average	**-92.46 (40.3)**	**-72.52 (39.8)**	**-48.07 (47.6)**	**-169.21 (30.6)**	**-12.22 (41.6)**	**-49.16 (45.9)**

Table 2: Return (SR%) under the 10K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	-280.23 (6.3)	-237.13 (13.6)	-107.17 (23.2)	-747.15 (0.0)	-180.68 (1.2)	-521.57 (0.3)
MountainCar	-44.28 (11.2)	-33.98 (14.6)	-4.46 (20.3)	-0.00 (0.0)*	-0.63 (0.0)*	-0.00 (0.0)*
Hopper	131.86 (98.3)	188.14 (96.0)	177.78 (100.0)	330.96 (100.0)	71.34 (100.0)	96.07 (100.0)
HalfCheetah	-329.76 (1.7)	-373.86 (0.0)	-351.46 (0.1)	-207.70 (14.1)	-2.99 (0.0)*	-0.35 (31.3)
Walker2d	123.56 (79.4)	46.96 (56.7)	197.91 (89.6)	91.37 (77.1)	40.88 (100.0)	61.17 (100.0)
Swimmer	19.36 (91.4)	16.25 (85.0)	20.28 (97.3)	-5.28 (34.2)	0.25 (50.6)	-2.05 (40.4)
Average	**-63.25 (48.1)**	**-65.60 (44.3)**	**-11.19 (55.1)**	**-89.64 (37.6)**	**-11.97 (42.0)**	**-61.12 (45.3)**

5.4 Budget Scaling Analysis

Figure 3 summarizes the budget-scaling trends, with detailed results provided in Appendix H.

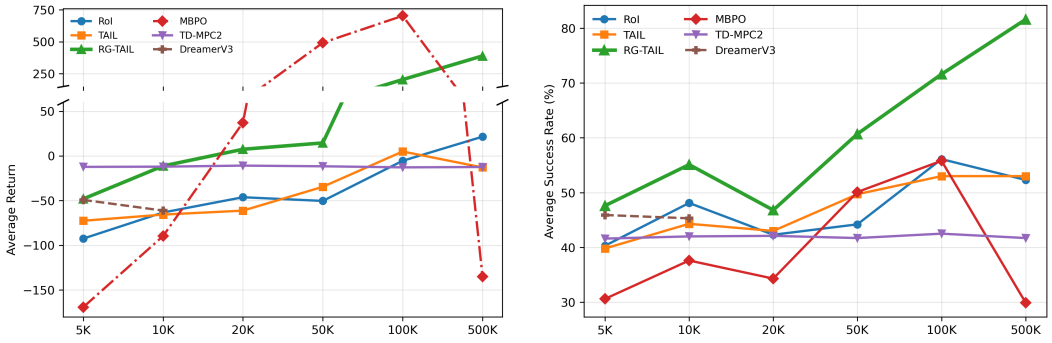
• **Scarce Budget: 5K, 10K.** Under extremely limited real-data budgets, the learned world model remains insufficiently trained, making synthetic experience more susceptible to model bias and there-

fore more likely to harm policy optimization. In this regime, RG-TAIL's conservative routing strategy proves particularly effective. Although trust estimation is relatively noisy at 5K due to limited supervision, it becomes substantially more reliable at 10K, allowing reliability-gated synthetic-data utilization to consistently achieve the best overall performance.

• **Medium Budget: 20K, 50K, 100K.** As the real-data budget increases, the world model becomes substantially more accurate, reducing the benefit of aggressively filtering synthetic experience. Consequently, RG-TAIL no longer consistently outperforms all baselines on every environment, as its conservative routing may occasionally discard useful imagined transitions for certain environments. Nevertheless, trust-aware weighting continues to provide positive guidance, enabling RG-TAIL to maintain the competitive overall average return and task success rate across all three budgets.

• **Sufficient Budget: 500K.** With abundant real experience, the marginal benefit of synthetic data diminishes while its potential risk remains. This is reflected by the observation that both MBPO and TD-MPC2 perform worse than the simple RoI baseline, suggesting that additional imagined experience becomes unnecessary or even detrimental. In contrast, RG-TAIL's conservative routing again becomes advantageous, consistently achieving the best performance across all environments, with approximately **+368** average return and **+29%** in success rate over the strongest baseline.

More importantly, RG-TAIL exhibits substantially more stable performance as the data budget increases, benefiting from its conservative fallback mechanism that can completely exclude synthetic experience. This robustness is particularly evident on HalfCheetah, where MBPO drops from a 96.7% success rate at 100K to only 0.3% at 500K, while RG-TAIL remains successful and continues to improve in return, as demonstrated in Table 18, 19, Figure 6a and 6b in the Appendix.



(a) Average return.

(b) Average success rate.

Figure 3: Comparison of RoI, TAIL, RG-TAIL, MBPO, TD-MPC2, and DreamerV3 across increasing real-data budgets, averaged across all applicable environments, algorithms, configurations.

6 Mechanistic Analysis

6.1 TAIL Failure Analysis

The results in Table 1 reveal that the improvements obtained by TAIL are highly environment-dependent. While TAIL improves over RoI on 3/6 environments, these gains do not generalize consistently across the benchmark suite. To further investigate the effectiveness of the trust module, we conduct a trust-failure ablation study. Four trust variants differ only in how trust values are assigned, with detailed definitions provided in Table 3.

The results reveal that the learned trust module consistently achieves the strongest average performance. This confirms that the trust signal indeed captures meaningful information and contributes positively to synthetic-data utilization. These findings suggest that the trust module itself should be retained, but additional mechanisms are required to determine when synthetic experience should be trusted in the first place. This observation directly inspires the RG-TAIL design.

Table 3: Trust-ablation variants used in the trust-failure analysis.

Variant	Actor Trust	Average
Learned Trust	normalized τ, clamped to $[\tau_{\min}, 1.0]$	**-72.52**
Uniform Trust	$\tau = 1$	-74.91
Random Trust	random weights	-75.15
Binary Trust	thresholded τ	-77.27

6.2 Ablation Analysis

We further conduct ablation studies to evaluate the effectiveness of the two core components of RG-TAIL: trust weighting and reliability gating. Three variants are considered. The *No Trust* variant removes trust weighting by setting all trust weights to 1 while retaining the reliability gate. The *No Gate* variant removes the reliability gate and therefore reduces RG-TAIL to TAIL. Finally, the *No Trust or Gate* variant removes both components, making it equivalent to RoI.

As shown in Table 4, the full RG-TAIL achieves an overall return improvement of approximately **+36** compared with *No Trust*, **+24** compared with *No Gate*, and **+44** compared with *No Trust or Gate*. Taken together, these ablations indicate that neither component alone is sufficient to achieve the full benefit of the synthetic-data utilization strategy.

Table 4: Ablation study of RG-TAIL under the 5K real-data budget. M1 denotes the full RG-TAIL. M2 (*No Trust*)removes trust weighting while retaining the reliability gate. M3 (*No Gate*) removes the reliability gate (equivalent to TAIL). M4 (*No Trust or Gate*) removes both trust weighting and reliability gating (equivalent to RoI).

Environment	M1	M2	M3	M4	M1-M2	M1-M3	M1-M4
LunarLander	-256.19	-341.00	-177.46	-324.54	+84.81 ✓	-78.73 ×	+68.35 ✓
MountainCar	-26.88	-36.41	-31.25	-28.93	+9.53 ✓	+4.37 ✓	+2.05 ✓
Hopper	115.43	96.02	88.86	74.79	+19.41 ✓	+26.57 ✓	+40.64 ✓
HalfCheetah	-307.90	-337.58	-370.11	-336.30	+29.68 ✓	+62.21 ✓	+28.40 ✓
Walker2d	161.93	94.37	34.03	39.97	+67.56 ✓	+127.90 ✓	+121.96 ✓
Swimmer	25.18	22.85	20.79	20.23	+2.33 ✓	+4.39 ✓	+4.95 ✓
Average	**-48.07**	**-83.63**	**-72.52**	**-92.46**	**+35.56** ✓	**+24.45** ✓	**+44.39** ✓

6.3 Threshold Robustness Analysis

The reliability threshold value of 0.40 was selected *a priori* as an environment-agnostic decision boundary based on the bimodal separation of the initial reliability distribution. It was not obtained through threshold sweeping, hyperparameter optimization, or any downstream performance-driven tuning procedure, which avoids tailoring the decision boundary to a particular benchmark suite.

We additionally conducted threshold-robustness experiments using alternative thresholds of 0.35 and 0.45. Both variants achieved overall performance even better than the original 0.40 setting, as shown in Table 12. These results indicate that the effectiveness of RG-TAIL arise primarily from the reliability-gating mechanism itself rather than from a fortunate hyperparameter choice.

By contrast, the CV threshold serves as a conservative heuristic derived from its statistical meaning, and values exceeding 2 indicate that the standard deviation substantially exceeds the mean return magnitude, suggesting high unstable synthetic value targets.

6.4 Decoupled Gating Analysis

Motivated by the signal exploration in Appendix C, we explored a decoupled routing design that applies different gating signals to actor and critic updates. In D1, actor updates are gated by reliability ($\theta = 0.40$), while critic updates are gated by synthetic-return variance ($G_{\text{syn_std}}$, $\theta = 4.143$). When

a gate is inactive, the corresponding trust weight is set to zero, effectively removing synthetic data from that component. We further explored D2, which uses the same actor and critic gates as D1, but assigns a uniform trust weight instead of discarding synthetic data when a gate is inactive.

Detailed results are reported in Table 13. Averaged across environments, RG-TAIL outperforms both D1 and D2. Despite using the strongest critic-side signal and its empirically optimized threshold, the resulting decoupled designs fail to translate these predictive advantages into better overall performance. We hypothesize that the additional actor-critic mismatch introduced by independent routing decisions outweighs the benefit of more specialized gating signals. These results further support the simpler unified reliability-gate design adopted in RG-TAIL.

6.5 Case Study

6.5.1 Policy Dependence of Reliability

Averaged across environments, PPO, SAC and TD3 achieve the reliability scores of 0.612, 0.197 and 0.225 respectively. The routing behavior adapts accordingly: the `block_all` rate is only 5.6% under PPO but increases to 49.1% under SAC, substantially reducing the contribution of synthetic experience to policy optimization. This behavior is consistent with the characteristics of SAC, which induces a broader state distribution through entropy-regularized exploration, making synthetic rollouts more likely to enter unreliable regions. Thus, RG-TAIL's gate is not only model-aware, but also policy-aware, naturally activating more conservative routing decisions under certain policies.

6.5.2 Failure Mode of Gating: LunarLander

LunarLander exposes a false-positive gating failure, as the return drops sharply from TAIL to RG-TAIL under SAC. A low reliability (0.361) causes `block_all` to activate on approximately 90% of routing decisions and reducing the actor trust weight to nearly zero. This case shows that low reliability can occasionally suppress useful synthetic updates under TAIL when the policy-induced reliability estimate is overly pessimistic, a failure mode that can be partially mitigated by the dynamic reliability recalibration explored in Appendix D.

6.5.3 Return-Success Mismatch: MountainCarContinuous

MountainCarContinuous illustrates why return and success rate must be interpreted jointly. External baselines achieve near-zero returns, but their return distributions are almost deterministic and their success rates remain 0%. RG-TAIL has a lower average return (-26.88), but a much broader return distribution (std 40.97), a non-zero success rate of 10.9%, and a maximum return of +84. This indicates genuine task completion in a subset of episodes. The near-zero-return baselines instead converge to a trivial fixed point that avoids action cost but never reaches the goal.

7 Conclusion and Future Work

In this work, we investigated when synthetic experience generated by world models should be trusted for policy optimization under limited real-data budgets. Through the *Real-or-Imagined* (RoI) study, we show that synthetic experience can improve reinforcement learning, while commonly used uncertainty estimates do not necessarily reflect the downstream utility of synthetic transitions. Based on this observation, we propose RG-TAIL, a reliability-gated trust-aware imagination learning framework that combines transition-level trust estimation with a environment-level reliability gate. Extensive experiments across diverse environments, algorithms, and data budgets demonstrate that RG-TAIL improves the robustness and effectiveness of synthetic-data utilization under both extremely low- and high-budget settings. Compared with internal baselines and representative MBRL methods, RG-TAIL improves task success rates by approximately **+19%** on average.

Several directions remain open for future exploration. First, more general reliability estimators beyond model agreement and rollout consistency could be investigated. Second, future work could explore utility-aware trust learning that directly aligns synthetic transition evaluation with downstream policy performance. Finally, our current evaluation is limited to simulated control environments. Validating RG-TAIL on more complex real-world embodied systems and physical robotic tasks remains an important direction toward practical deployment.

8 Reproducibility Statement

Codes are available at https://anonymous.4open.science/r/RG-TAIL-C7F8/. The complete RG-TAIL procedure is summarized in Algorithm 1, and more details regarding reproducibility are discussed in Appendix I.

9 AI Use Disclosure

In this work, we used generative AI tools for: assist with translation. We have **not** used generative AI tools for: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools for: edit a research paper to improve readability, identify relevant literature. We have reviewed all AI-assisted work. All AI-assisted content, including literature suggestions and drafted text, was manually reviewed and revised by the authors. All algorithmic designs, implementations, experiments, and scientific conclusions were developed and validated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Mehran Aghabozorgi, Alireza Moazeni, Yanshu Zhang, and Ke Li. Wimle: Uncertainty-aware world models with imle for sample-efficient continuous control, 2026. URL https://arxiv.org/abs/2602.14351.

Brett Barkley and David Fridovich-Keil. A forensic analysis of synthetic data in rl: Diagnosing and solving algorithmic failures in model-based policy optimization, 2026. URL https://arxiv.org/abs/2510.01457.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf.

Bernd Frauenknecht, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. On rollouts in model-based reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 76464–76489, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/be7a642a92de108a57e4d50866144d73-Paper-Conference.pdf.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Ignat Georgiev, Varun Giridhar, Nick Hansen, and Animesh Garg. Pwm: Policy learning with multi-task world models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 13737–13757, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/240ea1741b205ea295721d55184ac43b-Paper-Conference.pdf.

David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/cf73d57b6dcda32b293df7c2d5341f49-Paper-Conference.pdf.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5faf461eff3099671ad63c6f3f094f7f-Paper.pdf.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf.

Cevahir Koprulu, Franck Djeumou, and ufuk topcu. Neural stochastic differential equations for uncertainty-aware offline rl. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 4889–4913, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0f25eb6e9dc26c933a5d7516abf1eb8c-Paper-Conference.pdf.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 16(1):1–118, 01 2023. ISSN 1935-8237. doi: 10.1561/2200000086. URL https://doi.org/10.1561/2200000086.

Zhao-Han Peng, Shaohui Li, Zhi Li, Shulan Ruan, Yu LIU, and You He. From observations to events: Event-aware world models for reinforcement learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 155763–155805, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fc09b26b85ab3abb2832bd555a2e4215-Paper-Conference.pdf.

Aidan Scannell, Mohammadreza Nakhaeinezhadfard, Kalle Kujanpää, Yi Zhao, Kevin Luck, Arno Solin, and Joni Pajarinen. Discrete codebook world models for continuous control. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 11111–11148, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1dd85d064697ec1a258cd2b8755b8c6d-Paper-Conference.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, J Terry, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4916. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d7ff1795e8527f6443371c3933bdb52b-Paper-Datasets_and_Benchmarks_Track.pdf.

Ran Wei, Nathan Lambert, Anthony McDonald, Alfredo Garcia, and Roberto Calandra. A unified view on solving objective mismatch in model-based reinforcement learning, 2024. URL https://arxiv.org/abs/2310.06253.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28954–28967. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f29a179746902e331572c483c45e5086-Paper.pdf.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Frequently Asked Questions

Q1: Why do we use a fixed reliability-gate threshold across environments and algorithms rather than tuning it for each setting?

Response: We intentionally use a shared reliability-gate threshold across environments and algorithms to evaluate whether the proposed mechanism remains robust without environment- or algorithm-specific tuning. Our goal is to highlight the effectiveness and cross-environment stability of the reliability-gating mechanism itself, rather than to optimize the threshold specifically for the experimental scope considered in this paper. As shown in the threshold-sensitivity analysis in Section 6.3 and Table 12, nearby choices such as 0.35 and 0.45 can lead to improved performance, and it is therefore reasonable to expect that further per-environment or per-algorithm tuning could yield additional gains.

In practical applications, we do not assume that a single universal threshold must always be used. Practitioners may instead perform a small number of held-out or testing rollouts, inspect the resulting reliability statistics, and select a threshold according to the desired reliability–coverage trade-off. When additional validation resources are available, several candidate thresholds may also be evaluated and the best-performing one selected. We therefore view the fixed threshold used in this work as a controlled experimental choice for assessing the generality of the mechanism, rather than as a theoretically optimal or universally applicable constant.

Q2: Does the coefficient of variation become unstable when the mean synthetic return is close to zero?

Response: Our CV definition already uses $|\mu(R_{\text{syn}})| + \epsilon$ in the denominator, which avoids division by zero and removes sign ambiguity. We acknowledge that the resulting CV can still become large when the mean synthetic return is close to zero. However, RG-TAIL does not use CV as a continuous optimization weight; it is only used through a thresholded reliability criterion. Therefore, once the CV exceeds the threshold, whether its numerical value is moderately or extremely large has no additional effect on the actor, critic, gradients, or trust scores.

Moreover, a near-zero mean combined with non-negligible return variance naturally indicates unstable synthetic rollouts, which is precisely the behavior that the CV criterion is intended to detect. We therefore interpret CV as a coarse instability indicator rather than a precisely calibrated continuous statistic in the near-zero-mean regime.

Q3: Is the reliability formulation theoretically grounded, or is it primarily heuristic?

Response: RG-TAIL is intentionally an observation-driven design rather than a theoretically derived optimum. Its formulation emerged from the empirical findings and mechanistic analyses developed throughout this work: RoI first revealed that predictive fidelity alone does not reliably translate into downstream RL utility, while the subsequent analysis of TAIL exposed additional failure modes that motivated the introduction of reliability gating. The transition from TAIL to RG-TAIL is therefore itself driven by mechanistic observations of when and why trust-aware imagination can still fail.

We explicitly recognize that the current reliability formulation, including its functional form and thresholds, is heuristic. To compensate for the absence of a formal optimality guarantee, we provide extensive mechanistic and empirical analysis in Section 6, including ablations, sensitivity studies, cross-environment evaluations, and analysis of the underlying reliability signals. The heuristic nature of the formulation does not preclude the consistent empirical effectiveness observed across environments, algorithms, and real-data budgets; rather, it reflects the empirical scope of the present study. Developing a more principled theoretical or utility-grounded formulation of reliability remains an important direction for future work.

Q4: Could the trust module and reliability gate be learned end-to-end with the downstream RL objective?

Response: We agree that an end-to-end formulation is a particularly promising direction for further bridging predictive uncertainty and downstream RL utility. In the current design, the trust module is trained through a separate supervised objective, and the resulting trust score is therefore only indirectly connected to policy improvement. Although this separation provides a simple and inter-

pretable mechanism for controlling imagined experience, it does not fully close the gap between model-level reliability and task-level utility.

A natural extension would be to couple trust estimation and reliability gating directly with the RL objective, such that the learned trust signal is optimized according to its actual contribution to downstream policy improvement. We believe that such an end-to-end formulation represents a highly promising direction for future work.

B Pseudo Code

The complete RG-TAIL procedure is summarized in Algorithm 1. The same reliability-gating mechanism is applied uniformly across PPO, SAC, and TD3.

Algorithm 1 RG-TAIL: Reliability-Gated Trust-Aware Imagination Learning

Require: Real dataset $\mathcal{D}_{\text{real}}$, policy π_θ, critic C_ϕ, deterministic world model WM, probabilistic world model PWM, trust module T_ψ, rollout horizon H
Require: Reliability threshold $\rho_{\min} = 0.40$, CV threshold $c_{\max} = 2.0$
Ensure: Updated policy π_θ
Initial calibration:
Roll out short synthetic trajectories from states in $\mathcal{D}_{\text{real}}$
Compute world-model disagreement τ_{disagree}
Compute global reliability ρ
for training iteration $t = 1, \ldots, T$ **do**
    Generate synthetic batch $\mathcal{B}_{\text{syn}}$ using the TAIL rollout procedure
    Compute synthetic episode-return variation *CV*
    Compute actor and critic trust weights (w_i^a, w_i^c) for synthetic transitions
    if $\rho \geq \rho_{\min}$ **then**
        `mode` $\leftarrow$ `pass`
        Keep (w_i^a, w_i^c) unchanged
    else if *CV* $\leq c_{\max}$ **then**
        `mode` $\leftarrow$ `block_actor`
        Set $w_i^a \leftarrow 0$ for all synthetic transitions
    else
        `mode` $\leftarrow$ `block_all`
        Set $w_i^a \leftarrow 0$ and $w_i^c \leftarrow 0$ for all synthetic transitions
    end if
    Combine real transitions with permitted synthetic transitions
    Real transitions always use $w_i^a = w_i^c = 1$
    Update π_θ and C_ϕ using the selected RL algorithm
end for
return π_θ

C Gating Signal Exploration

We conduct an S1-S3 signal study, which independently enables or disables synthetic data for actor and critic updates. S1 disables synthetic data for both, S2 enables only for the critic, S3 enables only for the actor. The performance difference $\Delta_{\text{critic}} = S2 - S1$ and $\Delta_{\text{actor}} = S3 - S1$ are used to construct both the beneficial and harmful labels for critic and actor.

Using these labels, we search for optimal signals that can be used for actor and critic gating. Experimental results identify reliability as the strongest predictor of actor synthetic utility, while synthetic-return variance emerges as the strongest predictor of critic synthetic utility. These labels are used for the Decoupled RG-TAIL design as explained in Section 6.4

D DYNAMIC RELIABILITY EXPLORATION

RG-TAIL computes reliability once during the initial calibration stage and keeps it fixed throughout training. Since synthetic trajectories are regenerated at every iteration using the current actor, the induced rollout distribution may change as the policy evolves. We therefore investigate whether periodically recomputing reliability provides additional benefit over the static design.

We compare the original *static* RG-TAIL against a *dynamic* variant that recomputes reliability at every TAIL iteration using the newly generated synthetic rollouts from the current actor. All other components, including the WM/PWM, trust module, reliability thresholds, synthetic-data ratio, and optimization settings, remain unchanged. The experiment is conducted under the 5K real-data budget with SAC across all six environments and 10 random seeds.

As shown in Table 5, dynamic recalibration produces identical performance to static calibration in five of the six environments. The only difference occurs on LunarLander, where dynamic recalibration improves the mean return from -460.87 to -382.22, while the success rate remains unchanged. Overall, the results indicate that periodically updating reliability does not provide a systematic advantage over the original one-time calibration.

Table 5: Static versus dynamically recalibrated reliability under the 5K real-data budget with SAC. Dynamic RG-TAIL recomputes reliability at every TAIL iteration using the current actor. All values are averaged over 10 random seeds.

Environment	**Static Return**	**Static SR**	**Dynamic Return**	**Dynamic SR**	**Δ Return**
LunarLander	-476.03 ± 103.2	0.1	-397.38 ± 212.0	0.1	$+78.65$
MountainCar	-28.61 ± 2.0	0.3	-28.61 ± 2.0	0.3	0.00
Hopper	123.42 ± 32.7	100.0	123.42 ± 32.7	100.0	0.00
HalfCheetah	-172.60 ± 16.5	0.0	-172.60 ± 16.5	0.0	0.00
Walker2d	355.48 ± 217.6	85.2	355.48 ± 217.6	85.2	0.00
Swimmer	23.61 ± 1.9	100.0	23.61 ± 1.9	100.0	0.00

The observed behavior can be explained by the thresholded nature of the reliability mechanism. Reliability affects RG-TAIL only through comparisons with the fixed reliability threshold of 0.40, rather than entering policy optimization as a continuous weight. Table 6 shows that the recalibrated reliability remains on the same side of the threshold throughout training for five environments. LunarLander is the only case in which reliability crosses the threshold, increasing from an initial value of 0.361 to a maximum of 0.443. Consequently, dynamic recalibration changes the gating decision only in this environment.

Table 6: Initial and maximum dynamically recalibrated reliability values under the 5K setting. The reliability threshold is fixed at 0.40.

Environment	**Initial Reliability**	**Maximum Reliability**	**Crosses 0.40?**
LunarLander	0.361	0.443	✓
Swimmer	0.227	0.332	×
MountainCar	0.164	0.206	×
Hopper	0.169	0.335	×
HalfCheetah	0.119	0.137	×
Walker2d	0.098	0.128	×

These results suggest that one-time reliability calibration is generally sufficient under the evaluated setting. Dynamic recalibration only affects a small subset of cases, typically when the reliability estimate approaches the gating boundary after several policy-update iterations, and the resulting improvement is marginal rather than systematic. Moreover, periodically refreshing reliability introduces repeated reliability-score computation and gating overhead throughout training. Considering this limited empirical benefit together with the additional computational cost, we retain one-time static calibration as the default design of RG-TAIL, while viewing dynamic recalibration as an optional extension for threshold-adjacent cases.

E Statistical Significance Analysis

We further assess the statistical consistency of RG-TAIL under the **5K** real-data budget using success rate (SR), which provides a unified measure of task completion across environments. To preserve the hierarchical structure of the evaluation, we employ a hierarchical bootstrap. For internal comparisons with RoI and TAIL, environment–algorithm pairs are resampled as top-level units, followed by paired resampling over the 10 random seeds within each unit. For external comparisons with MBPO, TD-MPC2, and DreamerV3, environments are resampled as top-level units, followed by seed-level resampling within each environment. Holm correction is applied across the five pairwise comparisons.

As shown in Table 7, RG-TAIL achieves positive mean SR improvements over all five baselines. The gains over both internal baselines are statistically significant after Holm correction, with improvements of **+7.7 pp** over RoI ($p = 0.0208$) and **+8.3 pp** over TAIL ($p = 0.0396$), providing statistical support for the contribution of the proposed trust- and reliability-aware design. RG-TAIL also significantly improves over MBPO by **+17.3 pp** ($p = 0.0140$). Improvements over TD-MPC2 and DreamerV3 remain positive but do not reach statistical significance at the 5K budget, largely due to a small number of environment-specific deficits. In particular, RG-TAIL underperforms TD-MPC2 on Walker2d, and underperforms DreamerV3 on HalfCheetah; on the remaining environments, RG-TAIL is generally competitive or superior.

Table 7: Hierarchical-bootstrap significance analysis under the 5K real-data budget. ΔSR denotes the mean success-rate improvement of RG-TAIL over each baseline. Confidence intervals are 95% bootstrap intervals, and p-values are Holm-adjusted across all five comparisons.

Baseline	**ΔSR (pp)**	**95% CI (pp)**	**Holm p**
RoI	+7.7	$[+1.7, +15.0]$	**0.0208**
TAIL	+8.3	$[+1.3, +17.7]$	**0.0396**
MBPO	+17.3	$[+2.3, +37.3]$	**0.0140**
TD-MPC2	+6.1	$[-11.3, +26.3]$	1.0000
DreamerV3	+1.5	$[-19.3, +27.2]$	1.0000

F Runtime Overhead

We additionally profile the computational cost of RG-TAIL using Walker2d with SAC as a representative configuration. Table 8 reports measured wall-clock costs at the 5K and 500K real-data budgets, covering the two extremes of our experimental range. All measurements use the actual training data and checkpoints; no values are extrapolated.

The dominant cost arises from training the WM and PWM, requiring 30.7 s at 5K and approximately 50.5 min at 500K. Importantly, this cost is shared identically by RoI, TAIL, and RG-TAIL, and therefore does not constitute additional overhead introduced by RG-TAIL. The trust-module training and calibration required by TAIL and RG-TAIL adds only 1.92 s at 5K and approximately 2.15 min at 500K. In contrast, the reliability mechanism introduced specifically by RG-TAIL is extremely lightweight: reliability computation, CV estimation, and gating together require only 2.75 ms per training run at 5K and 313.7 ms at 500K. Even across all 10 random seeds, this corresponds to only approximately 3.1 s at the largest budget.

Overall, the additional cost of RG-TAIL relative to TAIL is therefore negligible compared with world-model training and downstream RL optimization. The measured costs also scale approximately with the amount of real or synthetic data for the data-dependent components, while the reliability gate itself remains effectively constant-cost.

G More Detailed Results

Table 9 compares the concepts of the two main signal, trust score and reliability. Table 10 collects detailed RoI results for each configuration and each environment. Table 11 compares RoI, TAIL

and RG-TAIL within each reinforcement learning algorithm. Table 12 collects detailed results of the threshold robustness analysis. Table 13 collects detailed results of the decoupled RG-TAIL experiments. Figure 4a, 4b, 5a, 5b, 6a and 6b compare the task success rate of RG-TAIL, MBPO, TD-MPC2 and DreamerV3 under each budget.

Table 8: Measured computational cost of the main RG-TAIL components on Walker2d with SAC at the 5K and 500K real-data budgets. WM/PWM and trust-module costs are incurred once per environment–budget configuration, while reliability and gating are measured per training run (seed).

Component	**5K**	**500K**	**Frequency**
WM training	14.0 s	1499.2 s	Once
PWM training	16.7 s	1532.4 s	Once
WM/PWM subtotal	**30.7 s**	**3031.6 s**	Once
WM trust training/calibration	1.27 s	63.6 s	Once
PWM trust training/calibration	0.64 s	65.3 s	Once
Trust subtotal	**1.92 s**	**129.0 s**	Once
Reliability computation	18.5 μs	2744.9 μs	Once/run
CV computation	272.6 μs/call	31.09 ms/call	10×/run
Reliability gate	0.94 μs/call	1.27 μs/call	10×/run
Reliability/gating subtotal	**2.75 ms**	**313.7 ms**	Per run

Table 9: Conceptual difference between trust weighting and reliability gating.

Mechanism	**Granularity**	**Decision Type**	**Control Target**
Trust weighting	Transition-level	Continuous	Gradient magnitude
Reliability gate	Environment-level	Binary	Synthetic learning

Table 10: RoI performance under the 5K real-data budget. Results are averaged across all applicable algorithms and 10 random seeds. Values are reported as mean ± standard deviation. Green represents the best performance.

Environment	**Real**	**Imagined-WM**	**Imagined-PWM**	**Hybrid-WM**	**Hybrid-PWM**
LunarLander	-353.4±216.9	-311.8±12.4	-388.1±169.9	-214.6±138.4	-383.8±285.2
MountainCar	-32.8±27.7	-24.0±10.4	-33.1±34.4	-23.2±17.4	-35.4±35.7
Hopper	+54.2±25.3	+80.8±65.4	+53.5±41.0	+86.0±68.0	+78.8±58.2
HalfCheetah	-373.4±60.7	-365.0±136.8	-355.3±151.8	-310.6±151.1	-314.3±122.6
Walker2d	+29.3±16.6	+9.3±22.4	+26.6±13.4	+50.1±46.2	+73.9±39.1
Swimmer	+20.1±6.9	+23.3±3.5	+14.3±4.1	+29.6±4.0	+13.7±9.6
Average	**-109.3**	**-97.9**	**-113.7**	**-63.8**	**-94.5**
MountainCar Success (%)	0.0±0.0	0.1±0.1	0.4±0.4	13.1±18.2	2.0±2.7

H ALL-BUDGET COMPLETE RESULTS

RoI, TAIL, RG-TAIL, MBPO and TD-MPC2 are evaluated across all budgets, with DreamerV3 evaluated on 5K and 10K settings, as the scarce budget setting is our primary focus. Each budget is represented by a complete comparison table. Table 14, 15, 16, 17, 18 and 19 collect detailed results for 5K, 10K, 20K, 50K, 100K and 500K real-data budgets respectively.

To account for the substantially different return scales across environments, we additionally report per-environment min–max normalized returns, where normalization is performed jointly across all available methods within each environment and budget. RG-TAIL achieves the highest average normalized return at every budget, with averages of 0.645, 0.728, 0.563, 0.610, 0.719, and 1.000 for 5K, 10K, 20K, 50K, 100K, and 500K, respectively. Tables 20, 21, 22, 23, 24, and 25 provide the complete normalized-return results for the six real-data budgets.

Table 11: Comparison of RoI, TAIL, and RG-TAIL (RGT) under the 5K real-data budget across PPO, SAC, and TD3. Values are averaged across applicable configurations and 10 random seeds. TD3 does not support LunarLander due to the discrete action space.

	PPO			SAC			TD3		
Environment	**RoI**	**TAIL**	**RGT**	**RoI**	**TAIL**	**RGT**	**RoI**	**TAIL**	**RGT**
LunarLander	-173.05	-155.41	-62.63	-476.03	-199.51	-450.85	–	–	–
MountainCar	-55.41	-47.42	-44.39	-22.24	-26.03	-28.61	-1.66	-20.31	-1.30
Hopper	156.06	159.43	164.52	42.88	79.90	123.42	25.43	27.26	54.29
HalfCheetah	-386.36	-391.80	-301.33	-144.90	-232.76	-172.60	-477.63	-485.78	-453.70
Walker2d	58.87	50.50	38.79	19.83	13.52	355.48	41.22	38.09	89.36
Swimmer	22.98	22.19	24.33	19.42	28.06	23.61	18.28	12.12	26.65
Average	**-62.82**	**-60.42**	**-30.12**	**-93.51**	**-56.14**	**-24.93**	**-78.87**	**-85.72**	**-56.94**

Table 12: Threshold robustness study of RG-TAIL (RGT) under the 5K real-data budget. The default threshold is 0.40. We additionally evaluate thresholds of 0.35 and 0.45. Δ denotes the improvement over RoI.

Environment	**RoI**	**RGT-0.35**	**RGT-0.40**	**RGT-0.45**	**Δ(0.35)**	**Δ(0.40)**	**Δ(0.45)**
LunarLander	-324.54	-157.60	-256.19	-256.19	+166.94	+68.35	+68.35
MountainCar	-28.93	-26.88	-26.88	-26.88	+2.05	+2.05	+2.05
Hopper	+74.79	+115.43	+115.43	+115.43	+40.64	+40.64	+40.64
HalfCheetah	-336.30	-312.42	-307.90	-307.16	+23.88	+28.40	+29.14
Walker2d	+39.97	+161.93	+161.93	+161.93	+121.96	+121.96	+121.96
Swimmer	+20.23	+24.69	+25.18	+26.86	+4.46	+4.95	+6.63
Average	**-92.46**	**-32.14**	**-48.07**	**-47.60**	**+60.32**	**+44.39**	**+44.86**

I Reproducibility Details

I.1 Computing Environment

All experiments were conducted on a server equipped with NVIDIA RTX GPU(s) with 32 GB memory. The core software environment consisted of Ubuntu 22.04.5 LTS, Python 3.11.8, PyTorch 2.8.0, Gymnasium 1.2.3, and NumPy 1.26.4.

I.2 Hyper-parameter Configuration

RL algorithm hyper-parameters. Table 26 lists the core hyper-parameters for each algorithm.

World model. The dynamics ensemble uses a 3-layer MLP, hidden dimension 256, with log-variance heads clamped to $[-20, 2]$ for numerical stability. Trained with Adam, learning rate 1×10^{-3}, batch size 256, weight decay 1×10^{-5}, for 300 epochs on an 80/20 train/validation split of the real-data budget.

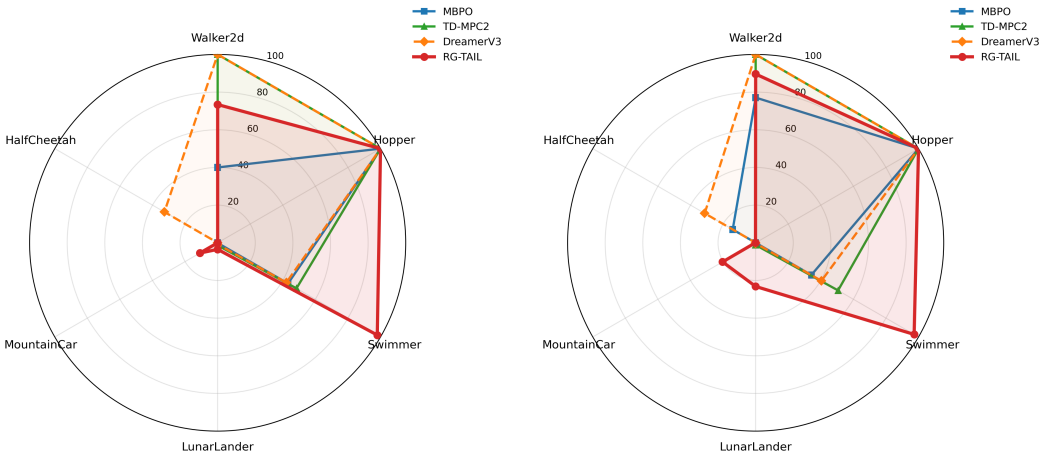
Trust module. A 2-layer MLP (Tanh activations, hidden dim. 64, sigmoid output) trained with Adam, learning rate 1×10^{-3}, batch size 256, weight decay 1×10^{-5}, for 100 epochs. The trust-clip floor τ_{clip} is set to 0.05 for hybrid configurations and 0.10 for imagined-only configurations, uniformly across all three algorithms; the rollout early-stop threshold is $\tau_{\text{var}} = 0.01$.

TAIL / RGTAIL rollout. Synthetic data volume scales linearly with the real-data budget B at a fixed ratio: $10B$ imagined transitions per run, generated over 10 rollout iterations. The imagination rollout horizon is environment- and algorithm-dependent as shown in Table 27.

Baselines. **MBPO** uses a dynamics ensemble of 7 members (5 elites), hidden dim. 200; SAC core with hidden dim. 256, learning rate 3×10^{-4}, $\gamma = 0.99$, $\tau = 0.005$, batch size 256, 20 SAC updates

Table 13: Performance of decoupled RG-TAIL variants. D1 uses independent actor and critic gates with synthetic data removed when a gate is inactive. D2 uses the same gates but retains synthetic data through uniform weighting when a gate is inactive.

Environment	RG-TAIL	D1	D2	D1-RGT	D2-RGT
LunarLander	-256.19	-226.80	-214.55	+29.39	+41.64
MountainCar	-26.88	-25.12	-23.35	+1.76	+3.53
Hopper	115.43	112.72	109.79	-2.71	-5.64
HalfCheetah	-307.90	-340.40	-348.84	-32.50	-40.94
Walker2d	161.93	57.33	75.59	-104.60	-86.34
Swimmer	25.18	26.30	24.17	+1.12	-1.01
Average	**-48.07**	**-66.00**	**-62.86**	**-17.93**	**-14.79**



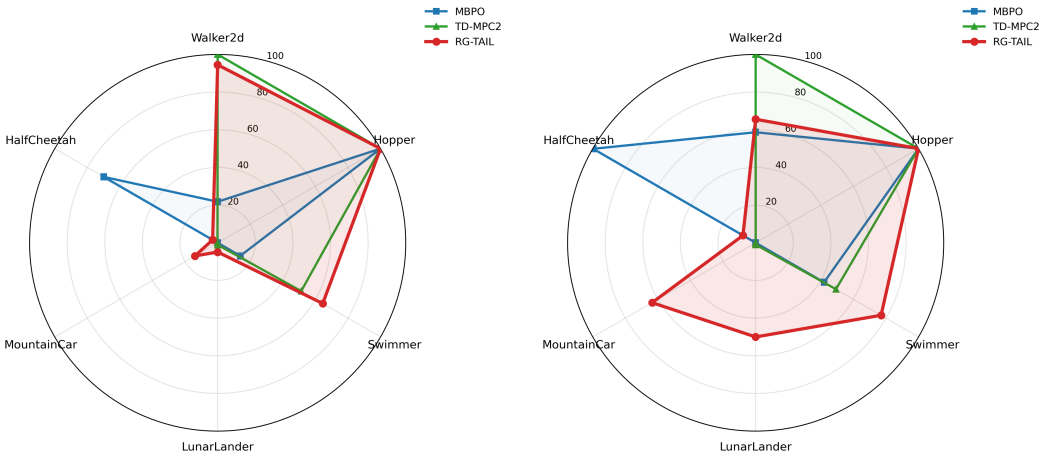
(a) 5K. (b) 10K.

Figure 4: Comparison of RG-TAIL, MBPO, TD-MPC2, and DreamerV3 under the 5K and 10K real-data budgets. Success rates are shown for each environment and averaged across all applicable algorithms, configurations, and 10 random seeds.

per environment step. MBPO-specific settings: real-data ratio 0.05, rollout length 1, model retrained every 250 environment steps for 5 epochs, model replay buffer capacity 10^5.

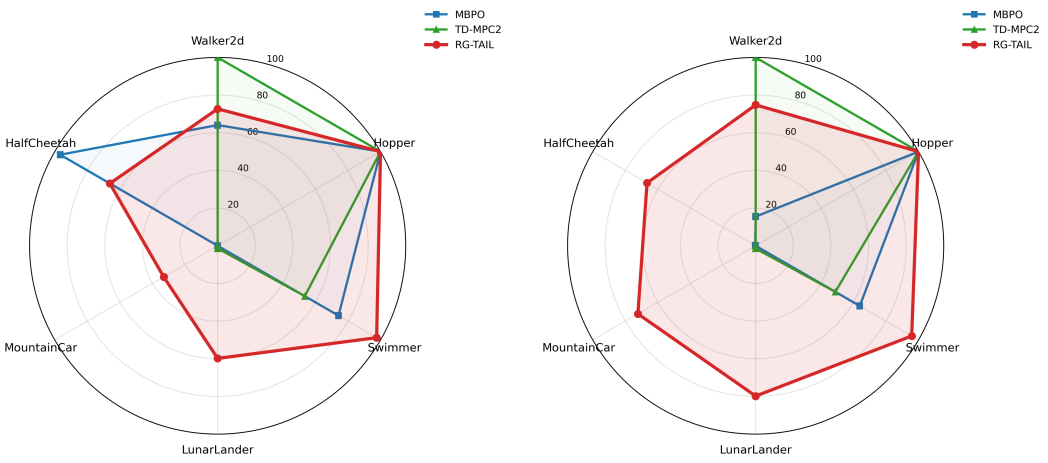
TD-MPC2 uses latent dimension 256, hidden dim. 512, MPPI planning horizon 3, 512 sampled trajectories per planning step, 6 CEM/MPPI refinement iterations, temperature 0.5, $\gamma = 0.99$, target update $\tau = 0.01$, learning rate 3×10^{-4}, batch size 256, replay buffer capacity 5×10^5. A small number of environments use env-specific overrides (e.g. MountainCar: horizon 5, 256 samples, temperature 0.3; Ant: hidden dim. 512, 256 samples, buffer 5×10^5).

DreamerV3 uses an RSSM world model with a 512-unit deterministic GRU state and a 32×32 categorical stochastic state (1024-dim, straight-through gradients), following the standard DreamerV3 architecture. Agent hyper-parameters: learning rate 3×10^{-4}, batch size 128, sequence length 64, train ratio 8, imagination horizon 15, $\gamma = 0.99$, $\lambda_{\text{GAE}} = 0.95$, entropy coefficient 3×10^{-4}.



(a) 20K. (b) 50K.

Figure 5: Comparison of RG-TAIL, MBPO, TD-MPC2, and DreamerV3 under the 20K and 50K real-data budgets. Success rates are shown for each environment and averaged across all applicable algorithms, configurations, and 10 random seeds.



(a) 100K. (b) 500K.

Figure 6: Comparison of RG-TAIL, MBPO, TD-MPC2, and DreamerV3 under the 100K and 500K real-data budgets. Success rates are shown for each environment and averaged across all applicable algorithms, configurations, and 10 random seeds.

Table 14: Return (SR%) under the 5K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	-324.54 (2.8)	-177.46 (3.8)	-256.19 (3.5)	-865.00 (0.1)	-183.85 (1.4)	-425.64 (0.8)
MountainCar	-28.93 (3.9)	-31.25 (6.1)	-26.88 (10.9)	-0.00 (0.0)*	-0.63 (0.0)*	-0.00 (0.0)*
Hopper	74.79 (90.0)	88.86 (91.7)	115.43 (100.0)	135.88 (100.0)	72.70 (100.0)	90.93 (100.0)
HalfCheetah	-336.30 (1.9)	-370.11 (1.6)	-307.90 (0.0)	-329.29 (0.1)	-2.98 (0.0)*	-0.33 (32.7)
Walker2d	39.97 (53.6)	34.03 (46.0)	161.93 (73.4)	42.89 (40.0)	41.56 (100.0)	40.89 (100.0)
Swimmer	20.23 (89.5)	20.79 (89.3)	25.18 (98.0)	0.24 (43.2)	-0.14 (48.4)	-0.83 (42.2)
Average	**-92.46 (40.3)**	**-72.52 (39.8)**	**-48.07 (47.6)**	**-169.21 (30.6)**	**-12.22 (41.6)**	**-49.16 (45.9)**

* A near-zero return does not necessarily indicate successful learning. The algorithm converges to a trivial stationary policy with 0% success rate.

Table 15: Return (SR%) under the 10K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	-280.23 (6.3)	-237.13 (13.6)	-107.17 (23.2)	-747.15 (0.0)	-180.68 (1.2)	-521.57 (0.3)
MountainCar	-44.28 (11.2)	-33.98 (14.6)	-4.46 (20.3)	-0.00 (0.0)*	-0.63 (0.0)*	-0.00 (0.0)*
Hopper	131.86 (98.3)	188.14 (96.0)	177.78 (100.0)	330.96 (100.0)	71.34 (100.0)	96.07 (100.0)
HalfCheetah	-329.76 (1.7)	-373.86 (0.0)	-351.46 (0.1)	-207.70 (14.1)	-2.99 (0.0)*	-0.35 (31.3)
Walker2d	123.56 (79.4)	46.96 (56.7)	197.91 (89.6)	91.37 (77.1)	40.88 (100.0)	61.17 (100.0)
Swimmer	19.36 (91.4)	16.25 (85.0)	20.28 (97.3)	-5.28 (34.2)	0.25 (50.6)	-2.05 (40.4)
Average	**-63.25 (48.1)**	**-65.60 (44.3)**	**-11.19 (55.1)**	**-89.64 (37.6)**	**-11.97 (42.0)**	**-61.12 (45.3)**

Table 16: Return (SR%) under the 20K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	-173.80 (5.4)	-204.34 (6.6)	-155.41 (4.9)	-872.07 (0.0)	-176.53 (1.1)
MountainCar	-17.46 (21.2)	-24.14 (16.5)	-23.24 (14.0)	-0.00 (0.0)*	-0.63 (0.0)*
Hopper	201.61 (100.0)	177.91 (98.1)	226.69 (100.0)	440.53 (100.0)	73.48 (100.0)
HalfCheetah	-377.99 (5.0)	-422.49 (0.8)	-260.05 (3.1)	662.49 (70.0)	-2.99 (0.1)*
Walker2d	83.72 (58.9)	93.41 (62.6)	249.70 (94.5)	3.56 (21.8)	41.46 (100.0)
Swimmer	6.85 (63.1)	12.61 (73.5)	7.19 (64.5)	-11.07 (13.9)	0.33 (51.3)
Average	**-46.18 (42.3)**	**-61.17 (43.0)**	**7.48 (46.8)**	**37.24 (34.3)**	**-10.81 (42.1)**

Table 17: Return (SR%) under the 50K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	-118.67 (22.9)	-123.15 (35.7)	11.03 (50.0)	-841.90 (0.0)	-179.24 (1.0)
MountainCar	-32.97 (17.9)	-23.26 (30.0)	33.00 (63.6)	-0.00 (0.0)*	-0.63 (0.0)*
Hopper	223.82 (100.0)	194.76 (100.0)	278.29 (100.0)	832.61 (100.0)	72.28 (100.0)
HalfCheetah	-440.53 (2.2)	-335.48 (9.0)	-366.02 (7.8)	2730.65 (99.8)	-3.03 (0.0)*
Walker2d	51.06 (43.0)	68.60 (55.3)	112.59 (65.6)	235.31 (58.7)	41.15 (100.0)
Swimmer	15.34 (79.3)	11.49 (68.5)	18.66 (77.0)	-0.15 (42.0)	0.10 (49.3)
Average	**-50.32 (44.2)**	**-34.51 (49.7)**	**14.59 (60.7)**	**492.75 (50.1)**	**-11.56 (41.7)**

Table 18: Return (SR%) under the 100K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	-76.76 (26.5)	-80.28 (26.4)	53.71 (59.8)	-821.67 (0.0)	-186.69 (1.4)
MountainCar	-19.29 (30.6)	-39.13 (16.6)	-13.44 (33.1)	-0.00 (0.0)*	-0.63 (0.0)*
Hopper	200.28 (100.0)	212.08 (100.0)	339.38 (100.0)	578.63 (100.0)	71.92 (100.0)
HalfCheetah	-306.98 (15.4)	-224.84 (23.8)	669.19 (66.2)	4351.49 (96.7)	-2.99 (0.0)*
Walker2d	151.11 (75.1)	142.98 (67.4)	142.46 (72.7)	92.74 (64.1)	41.35 (100.0)
Swimmer	19.20 (89.1)	19.16 (83.6)	35.54 (97.6)	17.36 (74.2)	0.45 (53.5)
Average	**-5.41 (56.1)**	**4.99 (53.0)**	**204.47 (71.6)**	**703.09 (55.8)**	**-12.76 (42.5)**

Table 19: Return (SR%) under the 500K real-data budget. All values are averaged across applicable algorithms and 10 random seeds. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	-74.26 (27.9)	4.31 (46.9)	62.04 (79.8)	-825.63 (0.0)	-184.13 (1.5)
MountainCar	-28.85 (21.4)	-24.61 (26.1)	36.89 (72.3)	-0.00 (0.0)*	-0.63 (0.0)*
Hopper	136.92 (100.0)	287.78 (99.2)	431.33 (100.0)	382.94 (100.0)	72.51 (100.0)
HalfCheetah	39.01 (32.7)	-463.01 (5.0)	1641.14 (66.7)	-374.29 (0.3)	-3.05 (0.0)*
Walker2d	33.69 (37.1)	98.75 (58.1)	129.45 (74.8)	-2.47 (15.5)	41.48 (100.0)
Swimmer	22.96 (94.7)	19.64 (83.0)	36.88 (95.8)	10.33 (63.9)	-0.28 (48.9)
Average	**21.58 (52.3)**	**-12.86 (53.0)**	**389.62 (81.6)**	**-134.85 (29.9)**	**-12.35 (41.7)**

Table 20: Per-environment min–max normalized return under the 5K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	0.786	1.000	0.885	0.000	0.991	0.639
MountainCar	0.154	0.000	0.140	1.000	0.980	1.000
Hopper	0.033	0.256	0.676	1.000	0.000	0.289
HalfCheetah	0.091	0.000	0.168	0.110	0.993	1.000
Walker2d	0.046	0.000	1.000	0.069	0.059	0.054
Swimmer	0.810	0.831	1.000	0.041	0.026	0.000
Average	**0.320**	**0.348**	**0.645**	**0.370**	**0.508**	**0.497**

Table 21: Per-environment min–max normalized return under the 10K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2	DreamerV3
LunarLander	0.730	0.797	1.000	0.000	0.885	0.352
MountainCar	0.000	0.233	0.899	1.000	0.986	1.000
Hopper	0.233	0.450	0.410	1.000	0.000	0.095
HalfCheetah	0.118	0.000	0.060	0.445	0.993	1.000
Walker2d	0.527	0.039	1.000	0.322	0.000	0.129
Swimmer	0.964	0.842	1.000	0.000	0.216	0.126
Average	**0.429**	**0.393**	**0.728**	**0.461**	**0.513**	**0.451**

Table 22: Per-environment min–max normalized return under the 20K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	0.974	0.932	1.000	0.000	0.971
MountainCar	0.277	0.000	0.037	1.000	0.974
Hopper	0.349	0.285	0.417	1.000	0.000
HalfCheetah	0.041	0.000	0.150	1.000	0.387
Walker2d	0.326	0.365	1.000	0.000	0.154
Swimmer	0.757	1.000	0.771	0.000	0.482
Average	**0.454**	**0.430**	**0.563**	**0.500**	**0.494**

Table 23: Per-environment min–max normalized return under the 50K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	0.848	0.843	1.000	0.000	0.777
MountainCar	0.000	0.147	1.000	0.500	0.490
Hopper	0.199	0.161	0.271	1.000	0.000
HalfCheetah	0.000	0.033	0.023	1.000	0.138
Walker2d	0.051	0.141	0.368	1.000	0.000
Swimmer	0.824	0.619	1.000	0.000	0.014
Average	**0.320**	**0.324**	**0.610**	**0.583**	**0.236**

Table 24: Per-environment min–max normalized return under the 100K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	0.851	0.847	1.000	0.000	0.725
MountainCar	0.507	0.000	0.657	1.000	0.984
Hopper	0.253	0.277	0.528	1.000	0.000
HalfCheetah	0.000	0.018	0.210	1.000	0.065
Walker2d	1.000	0.926	0.921	0.468	0.000
Swimmer	0.534	0.533	1.000	0.482	0.000
Average	**0.524**	**0.433**	**0.719**	**0.658**	**0.296**

Table 25: Per-environment min–max normalized return under the 500K real-data budget. Mean returns are normalized jointly across all available methods within each environment. Green represents the best performance.

Environment	RoI	TAIL	RG-TAIL	MBPO	TD-MPC2
LunarLander	0.846	0.935	1.000	0.000	0.723
MountainCar	0.000	0.065	1.000	0.439	0.429
Hopper	0.180	0.600	1.000	0.865	0.000
HalfCheetah	0.239	0.000	1.000	0.042	0.219
Walker2d	0.274	0.767	1.000	0.000	0.333
Swimmer	0.625	0.536	1.000	0.286	0.000
Average	**0.361**	**0.484**	**1.000**	**0.272**	**0.284**

Table 26: Core RL hyper-parameters (identical across all 6 environments).

	PPO	**SAC**	**TD3**
Learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4} (actor & critic)
Hidden dim. (2 layers)	64	256	256
Discount γ	0.99	0.99	0.99
Batch size	64	256	256
Target update τ (Polyak)	–	0.005	0.005
Gradient steps / env step	– (10 PPO epochs/batch)	10	10
GAE λ / clip ϵ	0.95 / 0.2	–	–
Entropy coefficient / α	0.01 (fixed)	0.2 (auto-tuned init.)	–
Policy/target noise	–	–	0.1 / 0.2 (clip 0.5)
Policy delay	–	–	2
Optimizer	Adam	Adam	Adam

Table 27: TAIL/RGTAIL imagination rollout horizon by environment and algorithm.

Environment	**PPO**	**SAC**	**TD3**
LunarLander	30	10	–
MountainCar	30	30	30
Hopper	30	15	30
HalfCheetah	30	15	30
Walker2d	30	15	15
Swimmer	20	15	15