SECOND THOUGHTS: EXPLORING AND PLANNING THROUGH COMPARATIVE IMAGINATION IN WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models let agents compare imagined outcomes before acting, but an agent must still decide which alternatives to explore and when a proposed plan should replace its learned policy. We introduce **SDI**(**S**ibling **D**isagreement in **I**magination), which connects these decisions through a shared actor-based reference. **SDI** compares action sequences imagined from the same model state, using repeated rollouts of the actor's own sequence to measure variation under fixed behavior. This reference scales both the spread across candidate returns, which provides an exploration signal, and a proposed plan's predicted advantage, which governs action acceptance. The agent executes a planned action when its advantage exceeds the reference variability by a fixed factor; otherwise, it follows the actor. **SDI** reuses the base agent's world model, actor, and critic without requiring an additional disagreement ensemble. Exploration and planning guide training data collection, while evaluation uses the learned actor alone. Across Meta-World, Adroit, and DM-Control, **SDI** achieves a higher observed task-average score than every compared baseline in each suite and matches or exceeds the strongest baseline's performance on 15 of 18 tasks. Component ablations and robustness analyses further examine the roles of exploration, planning exposure, and action acceptance. Paired evaluations from identical simulator states show that the gate improves both recovery from actor failures and preservation of successful behavior relative to ungated planning, while substantially reducing return degradation. Together, these results support comparing imagined alternatives against a shared actor reference as a unified approach to exploring and planning with world models.[1]

1 INTRODUCTION

Learned world models support two complementary ways to improve control. Agents such as Dreamer train a fast actor on imagined experience (Hafner et al., 2020; 2025), whereas PlaNet and TD-MPC2 search through predicted futures when an action is needed (Hafner et al., 2019; Hansen et al., 2024). Hybrid approaches use a learned actor or value function to guide online search (Sikchi et al., 2022). These approaches make imagination useful both for learning a policy and for revising its decisions. An actor acts quickly, whereas a planner searches at each step. During online learning, planned actions also change the actor's training data. When learning stalls, when should a searched action replace the actor's choice?

A planner optimizes predictions from an imperfect model. Its highest-scoring sequence can therefore exploit errors in those predictions rather than improve the real outcome (Talvitie, 2017; Lambert et al., 2020; Sims et al., 2024). Searching many alternatives can favor prediction errors, and even an accurately predicted high return offers little reason to intervene if the actor would do as well. Probabilistic ensembles and short model rollouts limit some forms of this risk (Chua et al., 2018; Janner et al., 2019), but a decision still requires comparing a proposed plan with the action the agent would otherwise take. During training, that choice also changes the experience from which the actor

[1]Code and configurations are available in the anonymous repository: `https://anonymous.4open.science/r/sdi-EEDA`.

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.

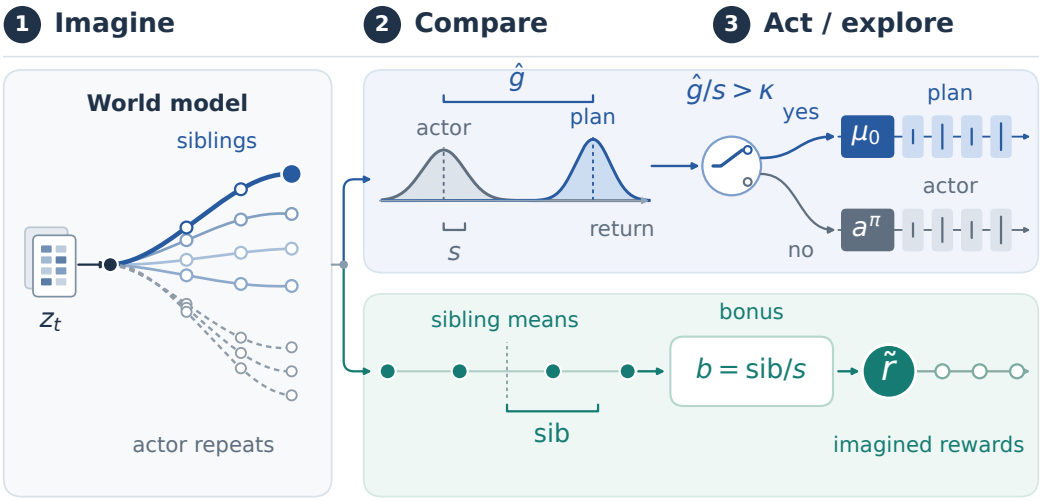


Figure 1: **Overview of SDI. Imagine:** candidate action sequences (siblings, blue) and repeated rollouts of the actor's fixed sequence (dashed gray) start at the same latent state z_t. **Compare:** fresh plan and actor rollouts provide the mean return gap $\hat{g}$ and actor-return standard deviation s; dots represent candidate mean returns, whose spread is measured by sib. **Act/explore:** the upper branch selects the planned action μ_0 or actor fallback a^π; dark cells indicate the executed action. The lower branch normalizes, clips, and weights the exploration bonus b to augment the first imagined reward $\tilde{r}$. The bonus anneals after the first training success. Exploration and gating use separate rollout settings; evaluation uses the actor alone.

learns. It is thus important to distinguish the effect of planning on the eventual policy from the effect of a particular intervention (Hamrick et al., 2021).

Exploration poses a related problem. Prediction error, novelty, and model disagreement can direct an agent toward unfamiliar experience (Pathak et al., 2017; Burda et al., 2019; Pathak et al., 2019; Sekar et al., 2020). These signals are valuable for learning the environment, but familiarity with the dynamics and progress on a particular task are different properties. Even a predictable state may offer actions with very different task outcomes. When a task reward is specified, the agent may therefore seek actions with meaningfully different predicted returns. A large spread across imagined actions is hard to interpret on its own, however: stochastic rollouts can also disagree about the return of one fixed action sequence. A useful comparison needs to account for that variation while remaining relevant to the actor's current behavior.

We introduce **SDI**(**S**ibling **D**isagreement in **I**magination), which compares imagined action sequences with repeated rollouts of the actor's fixed sequence from the same model state. This actor-based reference guides exploration and plan acceptance during data collection; evaluation uses the actor alone. SDI reuses the existing world model and search optimizer. Across Meta-World (Yu et al., 2020a), Adroit (Rajeswaran et al., 2018), and DM-Control (Tassa et al., 2018), SDI matches or exceeds the strongest compared baseline on 15 of 18 tasks. In a replicated, matched-occupancy ablation, it raises mean precision-manipulation success from 32% with ungated planning to 53%.

Our contributions are:

- **A task-directed measure of imagined disagreement.** We define sibling disagreement as variation in predicted task returns across action sequences imagined from the same latent state, measured relative to repeated rollouts of the actor's fixed sequence. This comparison separates action-dependent return variation from stochastic variation under fixed behavior.
- **A unified rule for exploration and selective planning.** We introduce SDI, which uses the same actor-repeat reference to reward informative imagined alternatives and to accept a planned action only when its predicted advantage exceeds local actor variability. SDI

requires no additional disagreement ensemble, guides data collection during training, and retains actor-only evaluation.

- **Broad and intervention-level empirical evidence.** We demonstrate that SDI achieves the highest observed task-average score among compared methods in each of three benchmark suites and matches or exceeds the strongest baseline mean on 15 of 18 tasks. Matched-occupancy ablations and paired-state executions further show improved actor learning, recovery, success preservation, and return retention relative to ungated planning.

2 Related work

World-model learning and planning. Dreamer trains an actor in imagined rollouts and executes it without further search (Hafner et al., 2020; 2021; 2025); R2-Dreamer provides the primary base agent in our experiments (Morihira et al., 2026). Decision-time planners instead search through model-based imagination (Hafner et al., 2019; Schrittwieser et al., 2020; Ye et al., 2021; Hansen et al., 2022; 2024; Nagabandi et al., 2020; Zhou et al., 2025), while hybrid methods combine search with learned policies or values (Sikchi et al., 2022; Anthony et al., 2017). LOOP combines finite-horizon model lookahead with an off-policy terminal value and constrains search relative to an actor (Sikchi et al., 2022). SDI adds an explicit post-search comparison against repeated actor rollouts and uses that same reference construction for exploration. This targets the acceptance of a proposal as well as the experience collected through planning (Hamrick et al., 2021).

Model error and uncertainty. Model error can compound over a rollout (Talvitie, 2017), prediction objectives need not align with control (Lambert et al., 2020), and planning can exploit poorly covered regions (Sims et al., 2024). Short rollouts, ensembles, and uncertainty-aware updates address these problems in different ways (Janner et al., 2019; Chua et al., 2018; Buckman et al., 2018; Yu et al., 2020b; Pan et al., 2020; Frauenknecht et al., 2024). PETS propagates uncertainty through probabilistic ensembles and trajectory sampling (Chua et al., 2018). SDI obtains its reference from stochastic repeats within the base world model. This estimates rollout variability for a fixed actor sequence; model bias remains possible (Berger et al., 2026). Our paired-state experiments test whether a margin scaled by that reference identifies executed improvements.

Exploration and action comparisons. Intrinsic rewards use prediction error, novelty, information gain, or ensemble disagreement to encourage exploration (Pathak et al., 2017; 2019; Burda et al., 2019; Houthooft et al., 2016; Sekar et al., 2020; Kauvar et al., 2023). Other signals measure action influence or reward disagreement (Seitzer et al., 2021; Bierling et al., 2025). Adaptive mechanisms switch exploration modes or balance intrinsic and extrinsic rewards (Pîslar et al., 2022; Chen et al., 2022), while dormant-representation diagnostics provide a different basis for adapting training interventions (Sokar et al., 2023; Xu et al., 2024). Plan2Explore rewards disagreement among predictions of the next observation embedding to seek expected information gain (Sekar et al., 2020). SDI instead compares task-return predictions across action sequences, relative to actor-repeat variability. This makes the signal task-directed and ties it to the reference used for plan acceptance. The construction relates to interventional advantage, counterfactual credit assignment, and twin models (Pan et al., 2022; Pan & Schölkopf, 2024; Mesnard et al., 2021; Balke & Pearl, 1994; Vlontzos et al., 2023); interventional causal representation learning likewise studies information obtained under interventions (Ahuja et al., 2023). We therefore evaluate SDI's predictions using executed alternatives.

3 Method

SDI augments a world-model actor–critic during training while leaving the base learning objectives unchanged. At each latent state, it constructs a local actor reference from repeated imagined rollouts of a fixed actor sequence. The same reference scales a task-directed exploration bonus and the advantage required to execute a planned action. Planning changes the experience collected for learning; evaluation executes the learned actor's mode without planning. Algorithm 1 summarizes the complete training-time interaction.

Algorithm 1 SDI training-time interaction. Exploration modifies imagined rewards, while selective planning modifies environment actions.

Input: latent z_t, actor π, critic V, model f_{img}

Draw fallback action $a_t \sim \pi(\cdot \mid z_t)$
Update bonus weight from the first-training-success schedule
if the exploration bonus is active **then**
    Compute b from sibling and actor-reference rollouts (Eq. 2)
    Weight b and augment the first imagined actor–critic reward
Update planner availability from episode-return history
if the planner is available **then**
    $\mu \leftarrow \text{MPPI}(z_t; N, H, I, R)$
    Fix the actor's mode sequence and re-score it and μ with R_g fresh rollouts
    Compute $\hat{g}$ and s_π (Eq. 1)
    if $\hat{g} > \kappa s_\pi$ **then** $a_t \leftarrow \mu_0$
Execute a_t, append the transition to replay, and update the base agent

3.1 Actor-referenced imagined returns

The base agent provides a recurrent latent dynamics model $z_{\tau+1} = f_{\text{img}}(z_\tau, a_\tau)$, reward head r, continuation head cont, actor π, and critic V. For an H-step action sequence, its continuation-weighted imagined return is

$$G(a_{0:H-1}) = \sum_{\tau<H} d_\tau r(z_{\tau+1}) + d_H V(z_H), \qquad d_0 = 1, \quad d_{\tau+1} = d_\tau \gamma \,\text{cont}(z_{\tau+1}).$$

Model predictive path integral (MPPI) search in latent space (Williams et al., 2016) evaluates candidate action sequences by averaging stochastic model rollouts. After search, SDI re-evaluates the selected plan and the actor's fixed mode sequence with R_g fresh rollouts each. Let $\overline{G}_{\text{plan}}$ and $\overline{G}_\pi$ denote their mean returns, and let s_π denote the standard deviation across actor-sequence rollouts. The imagined advantage and its normalized score are

$$\hat{g}(z) = \overline{G}_{\text{plan}} - \overline{G}_\pi, \qquad \text{IU}(z) = \frac{\hat{g}(z)}{s_\pi(z)}. \tag{1}$$

Both sequences start from the same latent state, so their mean-return gap compares the proposed change in behavior against what the actor would do there. The denominator answers a different question: how much can the predicted return vary when the action sequence is held fixed? A plan whose advantage is small relative to that local variation provides less reason to replace the actor than the same raw gap in a stable state. This scale does not correct model errors shared by the two sequences or turn IU into a calibrated probability of improvement. The actor sequence is generated once and held fixed across its repeated rollouts, so s_π measures return variation under fixed behavior. Exploration and action gating share the actor-referenced comparison, not the sampled trajectories: exploration uses short sibling rollouts and its own actor repeats, whereas the gate freshly re-scores a selected plan and actor sequence over the planning horizon. All imagined rollouts use independent model noise; Appendix A.1 distinguishes this sampling from the paired simulator randomness used only for analysis.

Sibling disagreement for exploration. Before the first successful environment transition, SDI evaluates K action sequences imagined from the same latent state. We call these sequences *siblings*. With $\hat{G}_k$ the estimated mean return of sibling k, their normalized return spread is

$$\text{sib}(z) = \text{std}_k(\hat{G}_k), \qquad b(z) = \frac{\text{sib}(z)}{s_\pi(z)}. \tag{2}$$

The sibling means summarize differences among action choices, whereas s_π summarizes variation from repeating one choice. Their ratio favors states where imagined actions differ in task return relative to that fixed-behavior variation. The bonus affects the actor–critic update in imagination; it is not added to the environment reward or used as the gate's plan advantage. We divide this score by its running mean, clip it to $[0, 3]$, and add it to the first reward of the imagined actor–critic rollout with weight $\beta = 0.5$. Exploration uses $K = 5$ siblings, $K_r = 3$ actor-reference repeats, and horizon 3. The first training transition carrying the environment's success flag starts an anneal over 20k optimizer updates. The environment reward itself is unchanged. The anneal gradually removes the auxiliary signal after discovery, leaving the original task reward to guide later learning.

Table 1: **Cross-domain benchmark performance under actor-only evaluation.** Meta-World (Yu et al., 2020a) and Adroit (Rajeswaran et al., 2018) report success; DM-Control (Tassa et al., 2018) reports episode return. Compared baselines are Plan2Explore (Sekar et al., 2020), Curious Replay (Kauvar et al., 2023), TD-MPC2 (Hansen et al., 2024), DreamerV3 (Hafner et al., 2025), and R2-Dreamer (Morihira et al., 2026). Columns show half and full training budgets, and suite means weight tasks equally.

Task	Plan2Explore		Curious Replay		TD-MPC2		DreamerV3		R2-Dreamer		**SDI**	
	half	full	half	full	half	full	half	full	half	full	half	full
Meta-World												
box-close	.00	.00	.67	.98	.81	.98	.00	.22	.58	.97	**.90**	**.99**
sweep-into	.00	.12	.77	.94	.76	.95	.00	.00	**.90**	.97	.79	**.98**
hammer	.28	.23	.00	.00	1.00	.99	.00	.00	1.00	1.00	.53	1.00
bin-picking	.00	.00	.00	.48	.22	.86	.08	.57	**.48**	.61	.01	**.92**
lever-pull	.08	.12	.00	.45	.24	.37	.24	.57	.40	.90	**.98**	**.98**
peg-insert-side	.00	.00	.04	.96	.05	.96	.05	.59	**.76**	.97	.74	**1.00**
hand-insert	.00	.01	.02	.34	.01	.04	.00	.00	.04	.13	**.11**	**.49**
assembly	.00	.00	.01	.00	.00	.01	.02	.09	.00	.00	**.07**	**.75**
disassemble	.00	.00	.00	.00	.01	.00	.00	.32	.01	.12	.00	**.35**
pick-out-of-hole	.00	.00	.00	.41	.00	.00	.00	.08	.03	.31	**.15**	**.96**
Mean	.04	.05	.15	.46	.31	.52	.04	.24	.42	.60	**.43**	**.84**
Adroit												
hammer-v1	.03	.11	.00	.32	.04	.26	.76	.60	**.93**	.99	.84	.99
door-v1	.58	.85	.45	.49	.00	.00	.00	.00	.00	.00	**.72**	.85
pen-v1	.04	.00	.16	.19	.10	.23	.07	.19	.06	.20	**.24**	**.44**
Mean	.22	.32	.20	.33	.05	.16	.28	.26	.33	.40	**.60**	**.76**
DM-Control												
cartpole-swingup	29	45	9.8	80	0.0	0.0	51	190	26	242	**62**	**279**
acrobot-swingup	**8.3**	20	4.2	23	2.1	3.9	8.0	**69**	2.0	5.8	3.7	16
ball-in-cup-catch	537	**973**	780	445	469	634	**937**	959	750	949	820	961
reacher-hard	51	375	119	370	132	273	21	187	268	819	**732**	**896**
finger-turn-hard	198	304	84	186	**199**	373	163	**616**	163	592	179	579
Mean	164.7	343.4	199.4	220.8	160.4	256.8	236.0	404.2	241.8	521.6	**359.3**	**546.2**

3.2 Search, fresh rescoring, and plan acceptance

SDI separates proposing a plan from deciding whether to execute it. MPPI first searches for a high-return sequence by iteratively sampling candidates and refitting its action distribution to the elites. Because the selected sequence is optimized on these search rollouts, their returns are not reused as evidence for acceptance. Instead, SDI re-evaluates the selected plan and the actor's fixed mode sequence from the same latent state using independent model noise and $R_g = 4$ fresh rollouts per sequence. This post-search comparison supplies $\hat{g}_t$ and $s_\pi(z_t)$ in Eq. 1; the complete MPPI update and initialization appear in Appendix A. Using the actor's repeats to set the threshold also keeps acceptance tied to the current policy, rather than to the spread of candidates that search has already optimized.

Scoring holds the actor's mode sequence fixed, while a rejected plan executes a fresh actor sample. This separates the gate's repeatable reference from stochastic training behavior.

At a planning step, the agent executes the first planned action μ_0 only when the freshly estimated advantage exceeds the actor-reference variability by a factor κ:

$$a_t = \begin{cases} \mu_0, & \hat{g}_t > \kappa s_\pi(z_t), \\ a_t^{\text{actor}} \sim \pi(\cdot \mid z_t), & \text{otherwise}, \end{cases} \qquad \kappa = 2. \tag{3}$$

When a plan is rejected, the agent samples a fresh actor action. Because s_π is a rollout standard deviation rather than a standard error, κ is an empirical operating threshold rather than a statistical confidence level. Only the first planned action is executed; the agent re-plans from its next latent state. These two uses of the actor reference address different decisions. Sibling disagreement identifies states where alternative actions have meaningfully different imagined returns; the gate tests whether a selected plan clears the actor's local return variability. The scores use separate rollout batches and

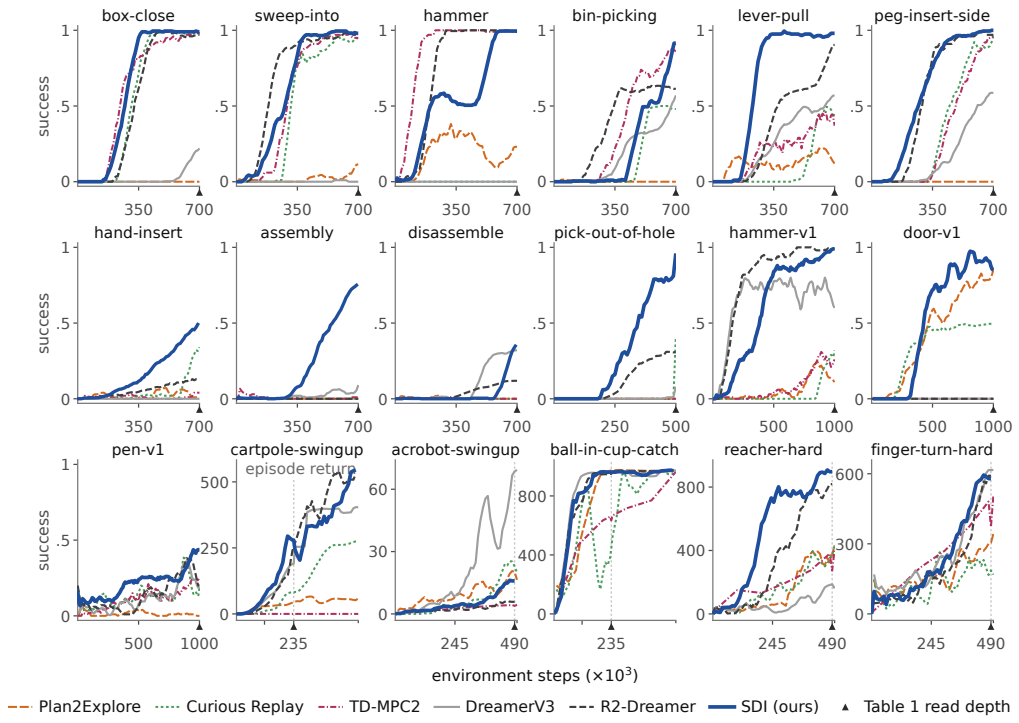


Figure 2: **Cross-domain benchmark learning curves.** Lines show average rolling success on Meta-World (Yu et al., 2020a) and Adroit (Rajeswaran et al., 2018) and average episode return on DM-Control (Tassa et al., 2018). Black triangles mark the Table 1 scoring depths; curves for the latter suite may extend toward the nominal 500k-step budget.

horizons. A state can therefore be useful to explore even when no proposed plan warrants replacing the actor at that step. Exploration changes the actor's imagined learning signal; an accepted plan changes the next environment transition and the experience stored in replay.

3.3 Training schedule and planner configuration

Planning is disabled for the first $t_{\min} = 20$k environment steps to allow initial model learning. Thereafter, a return-based predicate compares an exponential average of episode return with its value $W = 50$k steps earlier and activates planning when relative improvement falls below .05. Once activated, planning remains available. We call the fraction of acting steps with planner availability *occupancy*; this is distinct from the fraction on which a planned action passes the acceptance rule.

The planner uses $N = 256$ candidates, horizon $H = 8$, $I = 4$ refinement iterations, $E = 32$ elites, and $R = 4$ stochastic rollouts per candidate. The selected plan and actor reference are then evaluated with $R_g = 4$ fresh rollouts each. These settings and κ are fixed across domains. Search and re-scoring occur before acceptance, so a rejected proposal still requires the full planning computation. Appendix A provides optimizer updates, initialization, action bounds, and variant configurations.

4 Experimental setup

Benchmarks and budgets. Table 1 covers three pixel-based suites. Meta-World v3 (Yu et al., 2020a) uses 64×64 observations and sparse success rewards. The pre-specified discovery set comprises box-close, sweep-into, and hammer, where vanilla DreamerV3 (Hafner et al., 2025) has no success by 150k steps. The precision-plateau set comprises hand-insert, assembly, and disassemble. The Meta-World budget is 700k, except for pick-out-of-hole, which uses 500k. Adroit (Rajeswaran et al., 2018) includes hammer, door, and pen with dense training rewards and success-based evaluation at

Table 2: **Ablation study on precision-manipulation tasks.** Entries report mean evaluation success at 350k and 700k environment steps; the mean column gives the equally weighted average of the three 700k-step task means. Occupancy is the fraction of acting steps with planner availability, not planned-action execution. Base-agent controls are R2-Dreamer (Morihira et al., 2026) and DreamerV3 (Hafner et al., 2025).

Ablation	Component	hand-insert		assembly		disassemble		*mean*	*occupancy*
		350k	700k	350k	700k	350k	700k	700k	mean
	SDI	.11	.49	**.07**	**.75**	.00	**.35**	**.53**	.97
Base agent	R2-Dreamer	.04	.13	.00	.00	.01	.12	.08	.00
	DreamerV3	.05	.16	.00	.00	.00	.00	.05	.37
Planning	Ungated planner	.20	**.53**	.00	.29	**.03**	.14	.32	.97
	Half occupancy	.14	.47	.00	.00	.00	.01	.16	.47
Exploration	Always-on bonus	.02	.17	.00	.00	.01	.00	.06	.00
	Reset when stuck	.03	.16	.00	.08	.00	.00	.08	.00
Replay	Spread-prioritized replay	**.27**	.40	.00	.00	.00	.00	.13	.47

1M steps. DM-Control (Tassa et al., 2018) includes five tasks at a nominal 500k budget; comparisons use each task's common observed depth.

Baselines and evaluation. We compare with DreamerV3 (Hafner et al., 2025), R2-Dreamer (Morihira et al., 2026), Plan2Explore (Sekar et al., 2020), Curious Replay (Kauvar et al., 2023), and TD-MPC2 (Hansen et al., 2024). All reported baseline results come from our runs. For Meta-World (Yu et al., 2020a) and Adroit (Rajeswaran et al., 2018), the endpoint is mean success over the final 100 evaluation episodes (the last 10 evaluation calls); an episode succeeds if its success flag is set at any step. DM-Control (Tassa et al., 2018) averages episode return over five evaluation calls at a common scoring depth for each task. *Crossing* means reaching a last-10 evaluation success rate of at least 0.5 at the task budget.

5 Results

5.1 Cross-domain benchmark performance

Meta-World. Table 1 shows that across the ten Meta-World tasks (Yu et al., 2020a), SDI achieves the highest observed suite mean at both the half and full training budgets, improving from .43 to .84; the strongest baseline means are .42 and .60, respectively, from R2-Dreamer (Morihira et al., 2026). At the full budget, SDI is highest or tied highest on every task. Its largest gains over the strongest alternative occur on assembly (.75 versus .09), pick-out-of-hole (.96 versus .41), and hand-insert (.49 versus .34).

Adroit. On Adroit (Rajeswaran et al., 2018), SDI also has the highest observed suite mean at both budgets, reaching .60 halfway through training and .76 at the full budget, compared with .33 and .40 for the strongest baseline, R2-Dreamer. At the full budget, it ties the best reported result on hammer (.99) and door (.85), and leads on pen with .44, versus .23 for the strongest alternative. Thus, SDI matches or exceeds the best baseline endpoint on all three tasks.

DM-Control. DM-Control (Tassa et al., 2018) shows a more mixed task-level ranking despite the strongest suite averages. SDI obtains means of 359.3 and 546.2 at the half and full budgets, compared with 241.8 and 521.6 for the strongest baseline, R2-Dreamer. At the full budget, it leads on cartpole-swingup (279) and reacher-hard (896), and remains close to the best result on ball-in-cup-catch (961 versus 973). It trails DreamerV3 (Hafner et al., 2025) on acrobot-swingup (16 versus 69) and finger-turn-hard (579 versus 616). Across all three domains, these results correspond to matching or exceeding the strongest reported baseline mean on 15 of 18 tasks.

5.2 Training dynamics across domains

Figure 2 shows how the endpoint comparisons in Table 1 emerge over training; black triangles mark the half- and full-budget scoring depths. On several Meta-World (Yu et al., 2020a) tasks, endpoint convergence masks different routes to high success. On box-close and sweep-into, SDI enters the near-saturated regime after an initial learning phase and then remains stable. On hammer, TD-MPC2 (Hansen et al., 2024) and R2-Dreamer (Morihira et al., 2026) solve the task earlier, whereas SDI first plateaus near intermediate success and makes a second, late transition to saturation. The ordering is reversed on lever-pull: SDI rises rapidly to near-perfect success and maintains it while the alternatives improve more slowly. On bin-picking and peg-insert-side, R2-Dreamer (Morihira et al., 2026) makes earlier progress, but SDI continues improving and reaches the strongest full-budget performance.

The precision-manipulation tasks show the clearest separation. SDI remains near zero early on assembly and pick-out-of-hole, then undergoes a sustained late rise; hand-insert improves more gradually across the second half of training. Disassemble is different again: DreamerV3 (Hafner et al., 2025) improves first, but SDI's final increase produces the best endpoint. The Adroit (Rajeswaran et al., 2018) curves also distinguish early learning from eventual performance. R2-Dreamer (Morihira et al., 2026) reaches high hammer success fastest and SDI catches up later, while SDI continues to gain on door and pen despite visible fluctuations in the rolling success curves.

The DM-Control (Tassa et al., 2018) trajectories reflect the mixed endpoint pattern. Ball-in-cup-catch is largely an early-saturation task for the strongest methods, whereas cartpole-swingup and reacher-hard continue to separate methods later in training; on reacher-hard, SDI's advantage appears early and is largely maintained. By contrast, DreamerV3 (Hafner et al., 2025) exhibits the strongest late-stage rise on acrobot-swingup and narrowly exceeds SDI at the finger-turn-hard scoring depth. Thus, the suite-level averages summarize task-dependent learning dynamics, including early breakthroughs,delayed transitions, and differences in plateau stability, rather than uniform dominance throughout training.

5.3 Ablation study

Table 2 isolates the contributions of the base agent, planning exposure, action acceptance, exploration schedule, and replay prioritization on the three precision-manipulation tasks. The complete method obtains a final equal-task mean of .53 at .97 planner occupancy, compared with .08 for R2-Dreamer (Morihira et al., 2026) and .05 for the DreamerV3 (Hafner et al., 2025) transfer variant.

At matched .97 occupancy, removing the acceptance gate reduces the three-task mean from .53 to .32. The gate improves assembly from .29 to .75 and disassemble from .14 to .35, while hand-insert changes from .53 to .49. Reducing planner occupancy to .47 lowers the mean to .16 and leaves assembly and disassemble near zero, indicating that sustained planning exposure is important in this setting. The always-on bonus and reset controls reach means of .06 and .08, respectively. Spread-prioritized replay improves hand-insert to .40 but does not transfer to assembly or disassemble, yielding a mean of .13. Overall, no individual control reproduces the full method's cross-task result.

5.4 Paired-state analysis

Gate reliability and selectivity. For Figure 3a, we sample 500 decisions from each of 18 checkpoints across five Meta-World (Yu et al., 2020a) tasks and execute the actor and planned branches from the same state. A decision is counted as beneficial only when the planned branch produces a strictly positive realized return difference. This probability rises monotonically from about .24 in the lowest normalized-margin decile to .42 in the highest, crossing the all-plan rate of .33 in the upper half of the ranking. The imagined margin therefore contains useful ordinal information about realized improvement. Figure 3b shows the corresponding operating points. Always executing uses every proposal, a positive-margin rule uses 91%, the SDI gate uses 47%, and the constant-spread rule uses 14%. SDI thus occupies an intermediate regime, rejecting a slight majority of proposals without becoming as restrictive as the constant-spread rule. These execution shares measure selectivity rather than quality; comparisons among rules must also account for which decisions each rule retains.

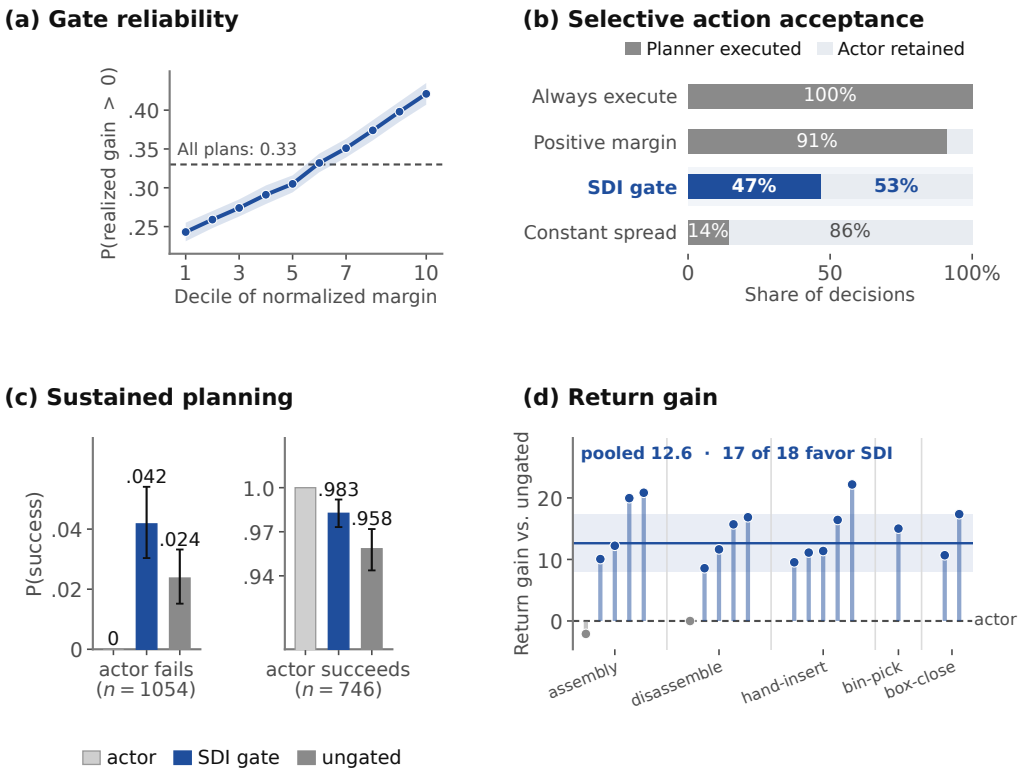


Figure 3: **Paired-state planning experiments.** (a) Realized one-decision improvement over the actor by normalized-margin decile; the dashed line is the all-plan rate. (b) Proposal acceptance under four rules. (c) Recovery and success preservation during 50-step continuations from restored states. (d) Checkpoint mean return difference between gated and ungated continuations. Shading in (a) shows the standard error; intervals in (c,d) are 95%.

Recovery and success preservation. The sustained-planning study in Figure 3c uses 1,800 restored states with planning available throughout each 50-step continuation. Among the 1,054 states where the actor fails, gated planning recovers success in 4.2% of cases, compared with 2.4% for ungated planning: 44 versus 25 recoveries. Among the 746 states where the actor succeeds, the gate preserves 98.3% of successes, compared with 95.8% under ungated planning, corresponding to 13 versus 31 lost successes. Selective execution therefore improves both sides of the intervention tradeoff: it recovers more actor failures while damaging fewer actor successes. The pooled net success change is $+.0172$ for gated planning, compared with $-.0033$ for ungated planning. Figure 3d directly compares the two planning rules at each checkpoint.

6 Conclusion

We introduced Sibling Disagreement in Imagination (SDI), which uses repeated actor rollouts as a shared reference for task-directed exploration and selective planning. The same actor-referenced comparison measures variation across imagined alternatives and determines when a plan should replace the actor, without requiring an additional disagreement ensemble. Planning guides data collection during training, while evaluation retains the learned actor alone. Thus, the benchmark endpoints assess policy improvement, while the paired-state experiments separately test the quality of the planner's training-time interventions. Across 18 tasks in three domains, SDI matches or exceeds the strongest reported baseline on 15 tasks. Overall, the findings support actor-referenced comparison as a practical way to connect exploration and planning in world-model learning.

AI USE STATEMENT

Generative AI tools assisted with methodological and experimental feedback; code scaffolding and experiment orchestration; analysis scripts and interpretation checks; literature search and bibliography verification; scientific-figure drafting; and manuscript organization, title development, and language editing. The tools did not generate synthetic datasets or autonomously collect experimental data, and this work makes no theoretical-proof claims. The authors ran and inspected all AI-assisted code, checked analyses and numerical claims against the retained experiment records, verified citations against primary bibliographic sources, and manually reviewed all generated text and figures. The authors take responsibility for the final content of this work, including text, claims, code, and artifacts produced with AI assistance.

ETHICS STATEMENT

This work studies model-based reinforcement learning in simulated control benchmarks and does not involve human participants, animal subjects, personal or sensitive data, or deployment on physical systems. Although the method is intended for general-purpose control research, improved planning and autonomous decision making could be misapplied in safety-critical or harmful settings. The reported benchmark results do not establish safety, robustness, or suitability for real-world deployment; in particular, world-model bias and planning error remain important limitations. Any transfer to physical or high-stakes domains should therefore include domain-specific risk assessment, human oversight, fail-safe constraints, and testing under distribution shift. We provide code, configurations, and methodological details to support scrutiny and responsible reuse.

REPRODUCIBILITY STATEMENT

Section 3 specifies the return construction, exploration bonus, acceptance rule, planning schedule, and search settings. Section 4 defines the task budgets, endpoints, and replication used for the main claims. Evaluation always uses the learned actor without planning. The supplementary material contains only results with complete reported cohorts. Code, configurations, and instructions are provided in the anonymous repository `https://anonymous.4open.science/r/sdi-EEDA`.

REFERENCES

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2209.11924.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *AAAI Conference on Artificial Intelligence*, pp. 230–237, 1994. URL https://aaai.org/Papers/AAAI/1994/AAAI94-035.pdf.

Julia Berger, Bernd Frauenknecht, Sebastian Trimpe, and Bastian Leibe. Biased dreams: Limitations to epistemic uncertainty quantification in latent dynamics models. *Reinforcement Learning Journal*, 7, 2026. URL https://arxiv.org/abs/2604.25416.

Lukas Bierling, Davide Pasero, Jan Henrik Bertrand, and Kiki Van Gerwen. DreamerV3-XP: Optimizing exploration through uncertainty estimation. *arXiv preprint arXiv:2510.21418*, 2025. URL https://arxiv.org/abs/2510.21418.

Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://arxiv.org/abs/1807.01675.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations (ICLR)*, 2019.

Eric Chen, Zhang-Wei Hong, Joni Pajarinen, and Pulkit Agrawal. Redeeming intrinsic rewards via constrained optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Bernd Frauenknecht, Artur Eisele, Devdutt Subhasish, Friedrich Solowjow, and Sebastian Trimpe. Trust the model where it trusts itself—model-based actor-critic with uncertainty-aware rollout adaption. In *International Conference on Machine Learning (ICML)*, pp. 13973–14005, 2024. URL https://arxiv.org/abs/2405.19014.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021.

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. URL https://doi.org/10.1038/s41586-025-08744-2.

Jessica B. Hamrick, Abram L. Friesen, Feryal Behbahani, Arthur Guez, Fabio Viola, Sims Witherspoon, Thomas Anthony, Lars Buesing, Petar Veličković, and Théophane Weber. On the role of planning in model-based deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=IrM64DGB21.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022.

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

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Isaac Kauvar, Chris Doyle, Linqi Zhou, and Nick Haber. Curious replay for model-based adaptation. In *International Conference on Machine Learning (ICML)*, 2023.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Conference on Learning for Dynamics and Control (L4DC)*, pp. 761–770, 2020.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S. Stepleton, Nicolas Heess, Arthur Guez, Eric Moulines, Marcus Hutter, Lars Buesing, and Remi Munos. Counterfactual credit assignment in model-free reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 7654–7664, 2021. URL https://arxiv.org/abs/2011.09464.

Naoki Morihira, Amal Nahar, Kartik Bharadwaj, Yasuhiro Kato, Akinobu Hayashi, and Tatsuya Harada. R2-Dreamer: Redundancy-reduced world models without decoders or augmentation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=Je2QqXrcQq.

Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning (CoRL)*, pp. 1101–1112, 2020.

Feiyang Pan, Jia He, Dandan Tu, and Qing He. Trust the model when it is confident: Masked model-based actor-critic. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2010.04893.

Hsiao-Ru Pan and Bernhard Schölkopf. Skill or luck? return decomposition via advantage functions. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2402.12874.

Hsiao-Ru Pan, Nico Gürtler, Alexander Neitz, and Bernhard Schölkopf. Direct advantage estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2109.06093.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, pp. 2778–2787, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning (ICML)*, 2019.

Miruna Pîslar, David Szepesvari, Georg Ostrovski, Diana Borsa, and Tom Schaul. When should agents explore? In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=Ewfxf14bca.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018.

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

Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning (ICML)*, 2020.

Harshit Sikchi, Wenxuan Zhou, and David Held. Learning off-policy with online planning. In *Conference on Robot Learning (CoRL)*, pp. 1622–1633, 2022.

Anya Sims, Cong Lu, Jakob Foerster, and Yee Whye Teh. The edge-of-reach problem in offline model-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2402.12527.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2302.12902.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. arXiv:1801.00690, 2018.

Athanasios Vlontzos, Bernhard Kainz, and Ciarán M. Gilligan-Lee. Estimating categorical counterfactuals via deep twin networks. *Nature Machine Intelligence*, 5:159–168, 2023. URL https://arxiv.org/abs/2109.01904.

Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, 2016. doi: 10.1109/ICRA.2016.7487277. URL https://doi.org/10.1109/ICRA.2016.7487277.

Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, Shuzhen Li, Yanjie Ze, Hal Daumé III, Furong Huang, and Huazhe Xu. DrM: Mastering visual reinforcement learning through dormant ratio minimization. In *International Conference on Learning Representations (ICLR)*, 2024.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020a. URL https://arxiv.org/abs/1910.10897.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2411.04983.

LIMITATIONS

What the actor reference measures. Repeated actor rollouts quantify stochastic return variation for one fixed imagined sequence; they do not capture model bias shared by all rollouts. The acceptance threshold κ is an empirical operating point rather than a statistical confidence level. Restored-state experiments show that the normalized margin ranks the frequency of realized improvement, but even its highest decile improves on the actor in fewer than half of paired decisions. The score should therefore not be interpreted as a calibrated probability, despite the better aggregate outcomes of gated than ungated continuations.

Planning cost and simulator access. SDI reuses the base world model, actor, and critic, but search and fresh re-scoring add acting-time computation even when a proposal is rejected. The paired-state analysis also requires simulator-state restoration and common random numbers; these are analysis tools and are not used during training. The experiments measure finite continuations from frozen checkpoints and do not isolate the long-term learning contribution of each accepted or rejected intervention.

A METHOD DETAILS

A.1 SEARCH AND RESCORING

The planner samples $N = 256$ candidate sequences over horizon $H = 8$, with $I = 4$ refinement iterations, $E = 32$ elites, and $R = 4$ independent stochastic rollouts per candidate. Search temperature is 1.0; initial and minimum action standard deviations are .3 and .05. Actions are clipped to the environment's bounds. At each iteration,

$$a^{(n)} = \mathrm{clip}(\mu + \sigma \odot \epsilon^{(n)}, a_{\min}, a_{\max}), \qquad \epsilon^{(n)} \sim \mathcal{N}(0, I), \tag{4}$$

$$\hat{G}^{(n)} = \frac{1}{R}\sum_{j=1}^{R} G_j(a^{(n)}), \qquad \zeta^{(n)} = \frac{\hat{G}^{(n)} - \mathrm{mean}_n \hat{G}^{(n)}}{\mathrm{sd}_n \hat{G}^{(n)}}, \tag{5}$$

$$w^{(e)} = \mathrm{softmax}_{e \in \mathrm{elites}}(\zeta^{(e)}/\lambda), \qquad \mu \leftarrow \sum_e w^{(e)} a^{(e)}, \tag{6}$$

$$\sigma^2 \leftarrow \mathrm{clip}\left(\sum_e w^{(e)}(a^{(e)} - \mu)^2, \sigma^2_{\min}, \sigma^2_{\max}\right). \tag{7}$$

Scores are standardized separately for each environment. The initial mean follows the actor's imagined mode sequence. On subsequent acting steps the previous sequence is shifted forward; episode boundaries reset this initialization.

After search, the plan and the actor's mode sequence receive $R_g = \max(R, 4) = 4$ fresh rollouts each. The gate compares their mean returns using the standard deviation of those actor rollouts. It executes only the first planned action; rejection falls back to an action sampled from the actor. The reference used for scoring is therefore the actor's fixed mode sequence, whereas the training-time fallback is stochastic. Evaluation uses the actor's mode alone.

Independent model noise. Candidate rollouts, repeated actor rollouts, and fresh re-scoring all use independent model noise. Exploration uses a separate batch with its own rollout settings. Sharing the starting latent state does not imply sharing the model's random draws. In contrast, the executed-alternative studies restore simulator state and use common random numbers where supported. These are evaluation procedures, not additional simulator access required by SDI during learning.