THINK FAST, PLAN SELECTIVELY: ADAPTIVE DELIBERATION FOR EFFICIENT DATA-DRIVEN MPC

Anonymous authors
Paper under double-blind review

ABSTRACT

Data-driven model predictive control (MPC) combines learned world models with online trajectory optimization, achieving strong performance in continuous control. However, the per-step cost of sampling and evaluating hundreds of candidate trajectories restricts deployment to control frequencies well below what real-time robotics demands. Motivated by the dual-process theory of human cognition, which distinguishes between fast, intuitive processing (*System 1*) and slower, deliberative reasoning (*System 2*), we ask whether every decision requires the same degree of computational deliberation. We propose **Fast-TD-MPC**, a lightweight framework that adaptively routes between fast policy execution and test-time planning, reserving costly deliberation for states where it is most needed. Fast-TD-MPC delivers competitive task performance across 103 continuous control tasks while achieving up to $\sim 4\times$ faster inference. Under external disturbances, Fast-TD-MPC selectively falls back to planning, maintaining robustness comparable to the original planner.

1 INTRODUCTION

Data-driven Model Predictive Control (MPC) combines learned world models with online trajectory optimization, achieving state-of-the-art performance across diverse domains (Hansen et al., 2024; Hafner et al., 2025; Schrittwieser et al., 2020). By learning to predict environment dynamics, world models (Ha & Schmidhuber, 2018) unlock *test-time planning*: the ability to propose, evaluate, and optimize candidate action sequences before execution, rather than reactively mapping states to actions (Hansen et al., 2022; Schrittwieser et al., 2020; Wang et al., 2025). This planning capability consistently outperforms purely reactive policies and enhances robustness in dynamic environments, driving progress in robotic manipulation and locomotion (Hansen et al., 2024; Hafner et al., 2025; 2020; Nagabandi et al., 2018; Hafner et al., 2019), autonomous driving (Guan et al., 2024; Liu et al., 2026b), and game-playing (Schrittwieser et al., 2020).

Planning brings robustness, but at the cost of substantial per-step computation. At each decision step, a planner samples and evaluates hundreds of candidate trajectories, resulting in per-step latencies that exceed the control frequencies required for real-time robotic deployment and latency-sensitive tasks (Dong et al., 2025; Gangapurwala et al., 2023). When planning is slow relative to changes in the environment, the state for which an action was planned may have already changed by the time it is executed, resulting in *stale-action* errors (Cheng et al., 2026; Lin et al., 2025b).

This raises a natural question: *does every decision require the same amount of deliberation?*

In most control problems, the answer is unlikely to be yes (Kahneman, 2011; Fridovich-Keil et al., 2018; Graves, 2016). Large portions of observations are familiar, and the action trajectories given such observations are predictable. Under such circumstances, a learned, reactive policy with low latency can already provide reliable actions; repeatedly invoking a full planner for these routine observations offers little benefit relative to its cost. Planning becomes more valuable when the agent encounters unfamiliar observations, disturbances, or situations where its learned behavior is less reliable. However, many test-time planners that utilize a learned world model devote the same computational budget to every timestep, ignoring the fact that not all decisions require planning.

A similar distinction arises in human decision-making. Dual-process theory (Kahneman, 2011; Daw et al., 2005; Botvinick et al., 2009) describes two complementary modes of thought: (i) a

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.

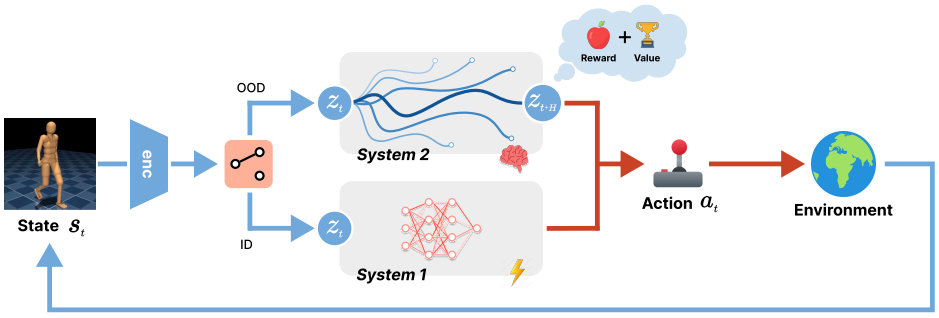


Figure 1: **Overview of Fast-TD-MPC.** Fast-TD-MPC uses TD-MPC2's MPPI planner as System 2 (the slow, robust fallback), a compact amortized MLP trained via planner imitation as System 1 (the fast, reactive policy), and a lightweight OOD detector as the gating mechanism. At each decision step, the OOD detector evaluates the current state: in-distribution (ID) states are routed to the fast and compact System 1, while out-of-distribution (OOD) states fall back to System 2 (TD-MPC2's MPPI planner). Since the detector itself costs negligible computation, the effective per-step latency scales linearly with the fraction of timesteps that require planning.

fast, intuitive *System 1* that handles familiar situations through learned experience, and (ii) a slower, deliberative *System 2* that reasons through complex or novel situations. Humans naturally alternate between both systems for effective decision-making.

Motivated by this view, we propose **Fast-TD-MPC**, a framework that dynamically switches between fast reactive execution and test-time planning. Applied to TD-MPC2 (Hansen et al., 2024), Fast-TD-MPC uses the original Model Predictive Path Integral (MPPI) planner as System 2 (the slow but robust fallback), a compact amortized neural network trained via planner imitation as System 1 (the fast, reflexive policy), and a lightweight out-of-distribution (OOD) detector as the gating mechanism (Figure 1). When the current observation is familiar, System 1 provides a reliable action at low latency; when the observation is OOD, System 2 is triggered to deliberate and plan. Since the detector itself costs negligible computation, the effective per-step latency scales linearly with the fraction of timesteps that require planning.

Our contributions are threefold:

- We introduce **Fast-TD-MPC**, a framework for adaptive test-time planning in data-driven MPC. Applied to TD-MPC2 (Hansen et al., 2024), it combines the original MPPI planner as a robust System 2 with a compact policy trained through planner imitation as a fast System 1, gated by a lightweight OOD detector, without modifying the underlying world model or planner.
- We conduct extensive evaluations across **103 continuous-control tasks** spanning four domains: DMControl (Tassa et al., 2018) (physics simulated locomotion and manipulation), Meta-World (Yu et al., 2019) (multi-object manipulation), ManiSkill2 (Gu et al., 2023) (robotic manipulation), and MyoSuite (Caggiano et al., 2022) (physiologically realistic musculoskeletal motor control). Fast-TD-MPC achieves up to $\sim 4\times$ inference speedup ($\sim 2.7\times$ on average) with $<10\%$ performance drop on 3 of 4 domains, using a single set of hyperparameters across all tasks (Figure 2).
- We show that faster decision-making does not come at the expense of robustness. Under five disturbance types spanning three categories, Fast-TD-MPC selectively falls back to planning when the agent encounters out-of-distribution states, retaining 95% of TD-MPC2's perturbed return on DMControl at half its latency.

2 Preliminaries

Problem formulation. We consider the standard infinite-horizon Markov Decision Process (MDP) (Puterman, 1990) setting, characterized by a tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma, p_0)$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$ are continuous states and actions, $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition (dynamics) function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function associated with a particular task, $\gamma \in [0, 1)$ is a discount factor, and p_0 is the initial state distribution. The goal is to derive a control policy $\pi : \mathcal{S} \to \mathcal{A}$ such

that the expected discounted sum of rewards (return) $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t\right], \quad r_t = \mathcal{R}(s_t, \pi(s_t))$ is maximized. In practice, each step must complete within the environment's control cycle; slow per-step computation incurs stale-action errors and degraded return.

TD-MPC2. Our work builds on TD-MPC2 (Hansen et al., 2024), an algorithm that combines test-time planning in a latent space using MPC (Negenborn et al., 2005) with actor-critic Temporal Difference (TD) learning. TD-MPC2 consists of five components as follows (θ denotes all learnable parameters, d_z the latent state dimensionality, d_a the action dimensionality):

- An encoder $\mathcal{E}_\theta : \mathcal{S} \to \mathbb{R}^{d_z}$ mapping a state s to its latent representation $z = \mathcal{E}_\theta(s)$.
- A latent dynamics model $\mathcal{D}_\theta : \mathbb{R}^{d_z} \times \mathcal{A} \to \mathbb{R}^{d_z}$, predicting the next latent $\hat{z}' = \mathcal{D}_\theta(z, a)$.
- A reward predictor $\mathcal{R}_\theta : \mathbb{R}^{d_z} \times \mathcal{A} \to \mathbb{R}$, computing expected rewards $\hat{r} = \mathcal{R}_\theta(z, a)$.
- A policy prior $\pi_\theta : \mathbb{R}^{d_z} \to \mathcal{A}$, a stochastic maximum-entropy policy that guides planning, from which actions $a \sim \pi_\theta(\cdot \mid z)$ are sampled.
- A critic $Q_\theta : \mathbb{R}^{d_z} \times \mathcal{A} \to \mathbb{R}$, estimating state-action values $Q_\theta(z, a) \approx Q^{\pi_\theta}(s, a)$. TD-MPC2 trains an ensemble of $N_Q = 5$ critics and computes TD-targets as the minimum of two randomly subsampled target critics to mitigate value overestimation.

The world model components $\mathcal{E}_\theta$, $\mathcal{D}_\theta$, $\mathcal{R}_\theta$, and Q_θ are jointly optimized by minimizing a combined objective over sub-trajectories of horizon H sampled from a replay buffer $\mathcal{B}$:

$$\mathcal{L}(\theta) \doteq \mathop{\mathbb{E}}_{(s,a,r,s')_{0:H}\sim\mathcal{B}}\left[\sum_{t=0}^{H}\lambda^t\left(\underbrace{\|\hat{z}'_t - \mathrm{sg}(\mathcal{E}_{\theta^-}(s'_t))\|^2}_{\text{joint-embedding prediction}} + \underbrace{\mathrm{CE}(\hat{r}_t, r_t)}_{\text{reward prediction}} + \underbrace{\mathrm{CE}(\hat{q}_t, q_t)}_{\text{value prediction}}\right)\right], \tag{1}$$

where $\hat{z}_t$ are recurrently predicted latents ($\hat{z}_0 = \mathcal{E}_\theta(s_0)$, $\hat{z}'_t = \mathcal{D}_\theta(\hat{z}_t, a_t)$), sg is the stop-gradient operator, θ^- are target parameters updated as an exponential moving average (EMA) of θ, $\lambda \in (0, 1]$ is a temporal coefficient that down-weights distant predictions, $q_t \doteq r_t + \gamma\bar{Q}(z'_t, \pi_\theta(z'_t))$ is the TD-target with $\bar{Q}$ being an EMA of the critic ensemble, and CE denotes cross-entropy with soft two-hot targets in a symlog-transformed discrete space.

The policy prior π_θ is trained independently by maximizing:

$$\mathcal{L}_\pi(\theta) \doteq \mathop{\mathbb{E}}_{(s,a)_{0:H}\sim\mathcal{B}}\left[\sum_{t=0}^{H}\lambda^t\left[\alpha\, Q_\theta(z_t, \pi_\theta(z_t)) + \beta\,\mathcal{H}(\pi_\theta(\cdot \mid z_t))\right]\right], \tag{2}$$

where $\mathcal{H}$ is the policy entropy and α, β are coefficients balanced via moving statistics to prevent premature entropy collapse. Gradients of $\mathcal{L}_\pi$ are taken only with respect to the parameters of π_θ.

At inference time, TD-MPC2 plans in the latent space using MPPI (Williams et al., 2015). Let $\mu, \sigma \in \mathbb{R}^{H\times d_a}$ be the parameters of a time-dependent multivariate Gaussian with diagonal covariance. Given the current state s_t of a decision step t, the encoder produces $z_t = \mathcal{E}_\theta(s_t)$, and MPPI iteratively refines the parameters (μ, σ) over action sequences of horizon H by (i) sampling N candidate trajectories, including a fraction from the policy prior π_θ; (ii) evaluating each via latent rollout and terminal value bootstrapping:

$$\hat{J} = \gamma^H Q_\theta(z_H, \pi_\theta(z_H)) + \sum_{h=0}^{H-1}\gamma^h\mathcal{R}_\theta(z_h, a_h); \tag{3}$$

and (iii) updating (μ, σ) via an importance weighted average over top-k elite trajectories, yielding refined parameters (μ^*_t, σ^*_t). Only the first action $a_t \sim \mathcal{N}(\mu^*_t, \sigma^{*2}_t)$ is executed, and the process repeats the next decision step, with μ^*_t shifted by one position to warm-start the next iteration's sampling. We refer readers to their paper for exact algorithmic details.

3 Fast-TD-MPC: A Framework for Adaptive Test-Time Planning

We study accelerating TD-MPC2 inference while preserving its robustness to states outside the expert distribution. Given a pre-trained TD-MPC2 world model (with frozen parameters θ), our goal is to route each decision step to the fastest system capable of handling it, *without* significantly degrading return. Fast-TD-MPC consists of three components, all frozen prior to deployment (Figure 1): (i) an amortized policy (System 1), (ii) the MPPI planner (System 2), and (iii) an OOD gate.

System 1: Amortized Policy. *Planner amortization* reduces the per-step cost of MPPI by training a lightweight feed-forward policy π_{S1} to imitate the planner. Given expert demonstrations $\mathcal{D}_{\text{exp}} = \{(z_t, a_t^*)\}$ collected via planner rollouts, we minimize the imitation objective: $\min_\phi \ \mathbb{E}_{(z,a^*)\sim\mathcal{D}_{\text{exp}}}[\ell(\pi_{\text{S1}}(z;\phi),\, a^*)]$, where ϕ are the parameters of π_{S1} and ℓ is an L1 loss, with $|\hat{a} - a^*|$ denoting the element-wise absolute difference summed over d_a. See Appendix **K** for training and architectural details. While this reduces inference to a single forward pass, the amortized policy cannot recover from states outside the expert distribution. We address this via OOD detection.

System 2: MPPI Planner. We use TD-MPC2's world model and MPPI planning algorithm (Section 2) as-is. When the gate switches from System 1 to System 2 after K timesteps, the warm-start buffer is K-step stale. We explicitly zero the buffer on System 1 to System 2 transitions, forcing a cold start from the policy prior π_θ, which provides fresh forward-looking candidates via the world model. The planner resumes normal warm-starting on subsequent consecutive System 2 steps.

OOD Gating. To address the failure mode of the amortized policy, we employ out-of-distribution (OOD) detection (Hendrycks & Gimpel, 2017; Lee et al., 2018) in the latent space. We employ Mahalanobis distance (Mahalanobis, 1936) as the scoring function: $g(z_t) = (z_t - \bar{z})^\top \Sigma_{\text{reg}}^{-1}(z_t - \bar{z})$, where $\bar{z}$ and Σ are the mean and full covariance of in-distribution (ID) latents, fitted offline with ℓ_2 shrinkage ($\Sigma_{\text{reg}} = \Sigma + \lambda\mathbf{I}$, $\lambda = 0.01$). To construct the ID set, we roll out π_{S1} and collect all visited latents $\{z_t\}$. We consider two labeling modes: *(i) theoretical*, where all visited states are treated as ID, and *(ii) reward-gated*, where a state is labeled ID only if its per-step rewards meet or exceed the average per-step counterparts of TD-MPC2 expert rollouts at the same timestep. The gating is then:

$$a_t = \begin{cases} \pi_{\text{S1}}(z_t;\phi) & \text{if } g(z_t) \le \tau, \\ \pi_{\text{S2}}(s_t;\theta) & \text{if } g(z_t) > \tau, \end{cases} \tag{4}$$

where τ controls the speed–robustness trade-off. For *theoretical* mode, τ defaults to the median[1] of all scores on a held-out validation set. For *reward-gated* mode, τ defaults to the midpoint between the median scores of the ID and OOD partitions. We additionally sweep multiple percentile values and report the full Pareto frontier in Section 4.2.

Latency Profile. Since $g(z_t)$ is a single matrix-vector product, the overhead of OOD detection is negligible. The effective per-step latency is: $t_{\text{total}} \approx \mathcal{E}_\theta + g + \rho \cdot \pi_{\text{S2}} + (1-\rho) \cdot \pi_{\text{S1}} + t_{\text{env}}$, where ρ is the S2 invocation rate, $\pi_{\text{S2}} \gg \pi_{\text{S1}}$, and $\mathcal{E}_\theta$, g, and t_{env} are incurred at every step. As $\rho \to 0$, the system approaches the speed of a purely amortized policy; as $\rho \to 1$, it approaches the robustness of full planning. The threshold τ directly controls this trade-off.

Algorithm. Algorithm 1 summarizes the full inference procedure. All components are computed offline and frozen at deployment.

4 Experiments

We seek to address the following research questions:

- ***RQ1:*** *Can Fast-TD-MPC achieve substantial inference speedup over TD-MPC2 without significant degradation in return or success rate across diverse continuous control tasks?*
- ***RQ2:*** *Can Fast-TD-MPC preserve the robustness of the original TD-MPC2 system under various disturbances?*

We additionally analyze the speed-robustness trade-off via τ and ablate the MPPI planner warm-start strategy and OOD distance metric.

4.1 Experimental Setup

We evaluate Fast-TD-MPC across 103 continuous control tasks spanning 4 domains: DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), ManiSkill2 (Gu et al., 2023), and MyoSuite

[1]Under Gaussian assumptions, the squared Mahalanobis distance follows a χ^2 distribution (Mahalanobis, 1936), which is right-skewed; the median is therefore more representative of the typical score than the mean.

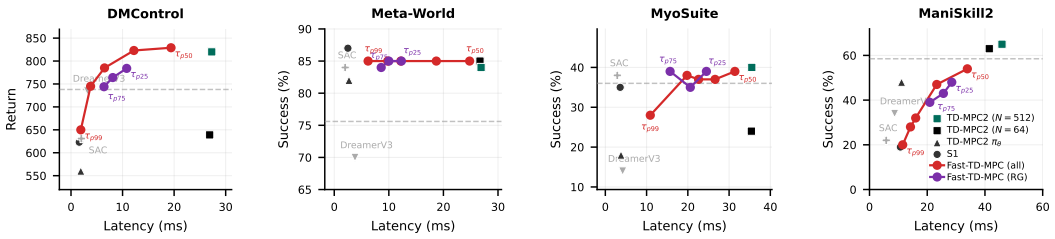


Figure 2: **Pareto frontier.** Return/success rate vs. per-step latency across 4 domains. Fast-TD-MPC (colored) traces the Pareto frontier across varying OOD thresholds, matching or surpassing baselines at comparable latency on DMControl, Meta-World, and ManiSkill2. Dashed line marks 90% of TD-MPC2's performance. DreamerV3 and SAC performance referenced from Hansen et al. (2024).

Table 1: **Benchmark results across 4 domains (103 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. †Return and success rate for DreamerV3 and SAC are referenced from Hansen et al. (2024). *all* denotes the *theoretical* OOD mode; *RG* denotes the *reward-gated* mode. See Appendix **F** for full results including S2 invocation rate and peak GPU memory usage.

Method	DMControl		Meta-World		MyoSuite		ManiSkill2	
	Return	Lat. (ms)	Succ. (%)	Lat. (ms)	Succ. (%)	Lat. (ms)	Succ. (%)	Lat. (ms)
DreamerV3†	$735_{\pm 93}$	$3.2_{\pm .1}$	$70_{\pm 8}$	$3.8_{\pm .1}$	$14_{\pm 17}$	$4.2_{\pm .1}$	$34_{\pm 56}$	$8.7_{\pm .7}$
SAC†	$631_{\pm 108}$	$2.0_{\pm .4}$	$84_{\pm 7}$	$2.0_{\pm .2}$	$38_{\pm 20}$	$2.9_{\pm .5}$	$22_{\pm 54}$	$5.8_{\pm .7}$
S2-alone (TD-MPC2)	$820_{\pm 55}$	$27.3_{\pm 1.1}$	$84_{\pm 9}$	$26.9_{\pm .2}$	$40_{\pm 20}$	$35.5_{\pm .4}$	$65_{\pm 54}$	$45.9_{\pm 4.8}$
S2-alone (N=64)	$639_{\pm 100}$	$26.9_{\pm 1.1}$	$85_{\pm 7}$	$26.7_{\pm .2}$	$24_{\pm 16}$	$35.4_{\pm .4}$	$63_{\pm 53}$	$41.5_{\pm 3.4}$
S2 policy-only	$560_{\pm 105}$	$1.9_{\pm .3}$	$82_{\pm 8}$	$2.7_{\pm .2}$	$18_{\pm 15}$	$3.8_{\pm .4}$	$48_{\pm 56}$	$11.1_{\pm 2.5}$
S1 alone	$623_{\pm 91}$	$1.6_{\pm .2}$	$87_{\pm 8}$	$2.5_{\pm .2}$	$35_{\pm 16}$	$3.6_{\pm .3}$	$19_{\pm 39}$	$10.7_{\pm 3.9}$
Round-robin (S2 50%)	$813_{\pm 57}$	$14.4_{\pm .7}$	$86_{\pm 7}$	$14.7_{\pm .2}$	$45_{\pm 21}$	$19.3_{\pm .3}$	$60_{\pm 44}$	$28.4_{\pm 5.6}$
Round-robin (S2 33%)	$782_{\pm 66}$	$10.1_{\pm .5}$	$86_{\pm 8}$	$10.7_{\pm .2}$	$41_{\pm 18}$	$14.1_{\pm .3}$	$42_{\pm 50}$	$22.3_{\pm 3.8}$
Round-robin (S2 20%)	$743_{\pm 74}$	$6.6_{\pm .4}$	$85_{\pm 8}$	$7.3_{\pm .2}$	$40_{\pm 22}$	$9.8_{\pm .3}$	$38_{\pm 48}$	$17.6_{\pm 3.9}$
Round-robin (S2 10%)	$699_{\pm 83}$	$4.0_{\pm .3}$	$86_{\pm 8}$	$4.9_{\pm .2}$	$40_{\pm 21}$	$6.6_{\pm .3}$	$28_{\pm 46}$	$14.5_{\pm 4.7}$
Fast-TD-MPC (*all*, τ_{p50})	$829_{\pm 55}$	$19.4_{\pm 2.4}$	$85_{\pm 8}$	$24.8_{\pm 1.1}$	$39_{\pm 21}$	$31.4_{\pm 3.6}$	$54_{\pm 55}$	$33.9_{\pm 12.1}$
Fast-TD-MPC (*all*, τ_{p75})	$823_{\pm 55}$	$12.2_{\pm 2.7}$	$85_{\pm 8}$	$18.7_{\pm 1.5}$	$37_{\pm 21}$	$26.6_{\pm 5.8}$	$47_{\pm 54}$	$23.3_{\pm 10.7}$
Fast-TD-MPC (*all*, τ_{p90})	$785_{\pm 58}$	$6.5_{\pm 2.0}$	$85_{\pm 8}$	$12.2_{\pm 1.6}$	$37_{\pm 21}$	$22.6_{\pm 7.6}$	$32_{\pm 42}$	$16.0_{\pm 5.2}$
Fast-TD-MPC (*all*, τ_{p95})	$745_{\pm 74}$	$3.8_{\pm 1.4}$	$85_{\pm 8}$	$9.8_{\pm 1.5}$	$38_{\pm 23}$	$19.8_{\pm 7.6}$	$28_{\pm 46}$	$14.2_{\pm 3.9}$
Fast-TD-MPC (*all*, τ_{p99})	$650_{\pm 92}$	$1.9_{\pm .3}$	$85_{\pm 8}$	$6.2_{\pm 1.3}$	$28_{\pm 17}$	$10.9_{\pm 5.8}$	$20_{\pm 39}$	$11.5_{\pm 3.9}$
Fast-TD-MPC (*RG*, τ_{p25})	$784_{\pm 68}$	$10.8_{\pm 2.3}$	$85_{\pm 8}$	$12.3_{\pm 1.7}$	$39_{\pm 22}$	$24.5_{\pm 4.6}$	$48_{\pm 43}$	$28.5_{\pm 20.9}$
Fast-TD-MPC (*RG*, τ_{p50})	$764_{\pm 70}$	$8.1_{\pm 1.8}$	$85_{\pm 8}$	$10.0_{\pm 1.5}$	$35_{\pm 21}$	$20.6_{\pm 5.2}$	$43_{\pm 43}$	$25.6_{\pm 18.9}$
Fast-TD-MPC (*RG*, τ_{p75})	$744_{\pm 74}$	$6.4_{\pm 1.5}$	$84_{\pm 8}$	$8.6_{\pm 1.5}$	$39_{\pm 21}$	$15.7_{\pm 4.7}$	$39_{\pm 51}$	$20.9_{\pm 10.8}$

(Caggiano et al., 2022), with action spaces up to $\mathcal{A} \in \mathbb{R}^{39}$ and diverse tasks from manipulation to complex locomotion. We consider the single-task setting with 3 seeds per task. All Fast-TD-MPC components are fitted independently for each task-seed pair using a single set of hyperparameters across all tasks; no per-task tuning done. See Appendix **C** and **K** for task and hyperparameter details.

Baselines. We compare against 5 TD-MPC2-derived baselines: (i) **S2-alone** (full TD-MPC2, N=512), (ii) **S2-reduced** (N=64), (iii) **S2 policy-only** (no MPPI), (iv) **S1 alone** (amortized MLP), (v) **Round-robin** (S2 invoked at a fixed rate of 50%, 33%, 20%, or 10%). We evaluate 2 OOD modes (*theoretical* and *reward-gated*) across multiple τ percentiles, and reference DreamerV3 (Hafner et al., 2025) and SAC (Haarnoja et al., 2018) numbers from Hansen et al. (2024). We follow Hansen et al. (2024)'s evaluation setup, and report return/success rate, per-step latency (ms), S2 invocation rate (ρ), as well as peak GPU memory. Refer Appendix **H** for hardware and evaluation details.

4.2 Speedup vs. Performance (***RQ1***)

We present aggregate results in Tab. 1 and Fig. 2, with per-component latency breakdowns in Tab. 2.

Pareto-optimal speed–performance trade-off. At τ_{p50}, Fast-TD-MPC (*all*) matches TD-MPC2 within the 95% CI across all domains (829 vs. 820 on DMControl, 85% vs. 84% on Meta-World)

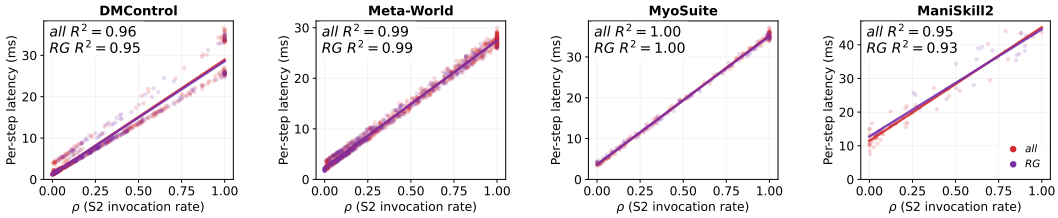


Figure 3: **Per-step latency vs. S2 invocation rate (ρ).** Each point is a task-seed configuration; red denotes the *theoretical* (*all*) OOD mode, purple denotes the *reward-gated* (*RG*) mode. The linear relationship ($R^2 \geq 0.93$) confirms that per-step latency scales linearly with ρ.

Table 2: **Per-component latency breakdown by hardware group and peak GPU memory.** Results are averaged within hardware groups: DMControl, Meta-World, and MyoSuite on H100; ManiSkill2 on A100. π_{S1} and π_{S2} report per-invocation cost (zero entries excluded), ρ is the S2 invocation rate, and $t_{\text{total}} \approx \mathcal{E}\theta + g + \rho\pi\text{S2} + (1-\rho)\pi_{S1} + t_{\text{env}}$.

	Policy components (ms)					Total	ρ	GPU
Method	$\mathcal{E}_\theta$	g	π_{S1}	π_{S2}	**Env**	**(ms)**	(%)	**(MB)**
H100 — DMControl + Meta-World + MyoSuite								
S2-alone (TD-MPC2)	$0.31_{\pm.00}$	—	—	$26.2_{\pm.6}$	$1.4_{\pm.2}$	$27.9_{\pm.7}$	100	$95_{\pm0}$
S1 alone	$0.38_{\pm.00}$	—	$0.49_{\pm.00}$	—	$1.4_{\pm.2}$	$2.3_{\pm.2}$	0	$68_{\pm0}$
Fast-TD-MPC (*all*, τ_{p50})	$0.35_{\pm.00}$	$0.17_{\pm.00}$	$0.40_{\pm.01}$	$26.2_{\pm.5}$	$1.4_{\pm.2}$	$23.3_{\pm1.4}$	$81_{\pm4}$	$90_{\pm2}$
Fast-TD-MPC (*all*, τ_{p90})	$0.35_{\pm.00}$	$0.17_{\pm.00}$	$0.38_{\pm.00}$	$26.3_{\pm.5}$	$1.4_{\pm.2}$	$10.9_{\pm1.6}$	$33_{\pm5}$	$76_{\pm2}$
A100 — ManiSkill2								
S2-alone (TD-MPC2)	$0.35_{\pm.01}$	—	—	$34.2_{\pm.4}$	$11.3_{\pm4.4}$	$45.8_{\pm4.8}$	100	$70_{\pm0}$
S1 alone	$0.37_{\pm.01}$	—	$0.65_{\pm.01}$	—	$9.7_{\pm3.9}$	$10.7_{\pm3.9}$	0	$42_{\pm0}$
Fast-TD-MPC (*all*, τ_{p50})	$0.38_{\pm.01}$	$0.21_{\pm.00}$	$0.59_{\pm.01}$	$34.4_{\pm.5}$	$11.4_{\pm4.8}$	$34.9_{\pm15.1}$	$66_{\pm37}$	$63_{\pm18}$
Fast-TD-MPC (*all*, τ_{p90})	$0.39_{\pm.01}$	$0.21_{\pm.00}$	$0.60_{\pm.01}$	$35.0_{\pm.5}$	$11.3_{\pm2.6}$	$17.0_{\pm4.7}$	$13_{\pm9}$	$46_{\pm8}$

while reducing latency by $\sim$1.3–1.4$\times$ (27.3 ms $\rightarrow$ 19.4 ms on DMControl, 45.9 ms $\rightarrow$ 33.9 ms on ManiSkill2). Varying τ traces a Pareto frontier: at τ_{p75}, speedup reaches $\sim$2$\times$ (27.3 ms $\rightarrow$ 12.2 ms on DMControl, and 45.9 ms $\rightarrow$ 23.3 ms on ManiSkill2); at τ_{p90}, Fast-TD-MPC achieves $\sim$2.7$\times$ average speedup (up to 4.2$\times$ on DMControl) with $<$10% performance drop on 3 of 4 domains, retaining 96% of TD-MPC2's return on DMControl, matching its success rate on Meta-World, and retaining 93% on MyoSuite. Beyond p_{90}, performance degrades sharply: at τ_{p99}, DMControl return drops to 650 ($\sim$20% loss) and ManiSkill2 success to 20%, as the gate routes nearly all steps to System 1, reducing to the S1-alone regime.

State-aware gating vs. fixed scheduling. Round-robin (RR) invokes System 2 at a fixed rate regardless of state. On DMControl, gating dominates every RR operating point at matched latency, e.g., 785 return at 6.5 ms (τ_{p90}) vs. 743 at 6.6 ms (RR 20%), and 745 at 3.8 ms (τ_{p95}) vs. 699 at 4.0 ms (RR 10%). On ManiSkill2, the two are comparable: gating is ahead at intermediate latency, with 47% at 23.3 ms (τ_{p75}) vs. 42% at 22.3 ms (RR 33%), while RR 50% is stronger at the high-latency end (60% at 28.4 ms vs. 54% at 33.9 ms for τ_{p50}). On Meta-World, all scheduling variants lie within 2 points (84–86%), indicating saturation. On MyoSuite, RR (S2 10%) recovers TD-MPC2's success (40%) at 6.6 ms, whereas the gate still invokes S2 on 60% of steps at τ_{p90}. Thus, *which* timesteps are planned matters most when only a small subset of states requires planning; otherwise, fixed scheduling is a simpler alternative.

OOD mode comparison. The *reward-gated* mode matches the *theoretical* mode's performance at lower latency across multiple domains. On Meta-World, *RG* τ_{p50} achieves 85% at 10.0 ms vs. *all* τ_{p90}'s 85% at 12.2 ms. On MyoSuite, *RG* τ_{p75} achieves 39% at 15.7 ms, matching *all* τ_{p50}'s 39% at 31.4 ms and exceeding *all* τ_{p90}'s 37% at 22.6 ms. This suggests reward-gating's tighter ID distribution produces a more precise threshold, reducing unnecessary S2 invocations on states System 1 can reliably handle.

Reference baselines. Both DreamerV3 and SAC are faster than TD-MPC2 but achieve lower returns, particularly on DMControl (735 and 631 vs. 820) and MyoSuite (14% and 38% vs. 40%).

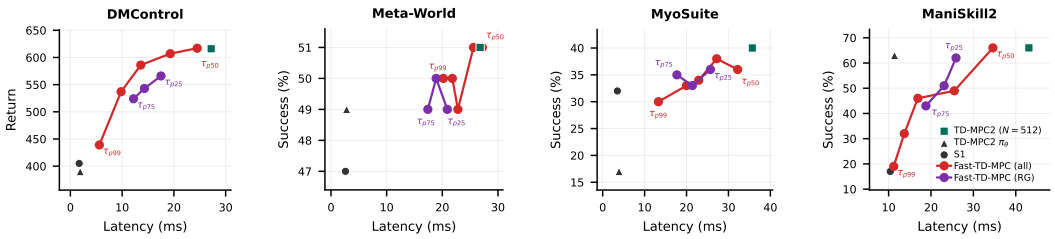


Figure 4: **Pareto frontier under disturbances.** Return/success rate vs. per-step latency across 4 domains. Fast-TD-MPC (colored) matches or surpasses TD-MPC2 at lower latency on DMControl, Meta-World, and ManiSkill2, with a $\sim$13% performance drop on MyoSuite for up to $\sim$2$\times$ speedup.

Table 3: **Results of benchmarks with disturbances across 4 domains (102 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. *all* denotes *theoretical* OOD mode; *RG* denotes *reward-gated* mode. See Appendix **G** for detailed results with all evaluation metrics.

	DMControl		Meta-World		MyoSuite		ManiSkill2	
Method	**Return**	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)
S2-alone (TD-MPC2)	$616_{\pm 104}$	$27.2_{\pm 1.1}$	$51_{\pm 10}$	$26.8_{\pm .2}$	$40_{\pm 22}$	$35.7_{\pm .5}$	$66_{\pm 75}$	$43.1_{\pm 4.7}$
S2 policy-only	$390_{\pm 102}$	$1.9_{\pm .3}$	$49_{\pm 10}$	$2.8_{\pm .2}$	$17_{\pm 15}$	$3.9_{\pm .4}$	$63_{\pm 66}$	$11.4_{\pm 5.7}$
S1 alone	$405_{\pm 98}$	$1.7_{\pm .3}$	$47_{\pm 10}$	$2.6_{\pm .2}$	$32_{\pm 15}$	$3.5_{\pm .3}$	$17_{\pm 52}$	$10.4_{\pm 4.2}$
Round-robin (S2 50%)	$577_{\pm 102}$	$14.5_{\pm .7}$	$49_{\pm 11}$	$14.7_{\pm .2}$	$37_{\pm 20}$	$19.4_{\pm .3}$	$58_{\pm 89}$	$27.1_{\pm 2.0}$
Fast-TD-MPC (*all*, τ_{p50})	$617_{\pm 105}$	$24.5_{\pm 1.9}$	$51_{\pm 11}$	$27.2_{\pm .2}$	$36_{\pm 20}$	$32.2_{\pm 3.9}$	$66_{\pm 80}$	$34.6_{\pm 18.0}$
Fast-TD-MPC (*all*, τ_{p75})	$607_{\pm 103}$	$19.3_{\pm 2.9}$	$51_{\pm 11}$	$25.6_{\pm .9}$	$38_{\pm 22}$	$27.2_{\pm 5.9}$	$49_{\pm 108}$	$25.5_{\pm 22.3}$
Fast-TD-MPC (*all*, τ_{p90})	$586_{\pm 101}$	$13.6_{\pm 3.0}$	$49_{\pm 10}$	$22.8_{\pm 1.6}$	$34_{\pm 22}$	$22.9_{\pm 7.8}$	$46_{\pm 86}$	$16.9_{\pm 10.5}$
Fast-TD-MPC (*all*, τ_{p95})	$537_{\pm 102}$	$9.8_{\pm 2.6}$	$50_{\pm 10}$	$21.8_{\pm 1.8}$	$33_{\pm 20}$	$20.0_{\pm 7.8}$	$32_{\pm 62}$	$13.7_{\pm 5.2}$
Fast-TD-MPC (*all*, τ_{p99})	$439_{\pm 101}$	$5.6_{\pm 2.0}$	$50_{\pm 10}$	$20.2_{\pm 2.0}$	$30_{\pm 18}$	$13.3_{\pm 7.8}$	$19_{\pm 46}$	$11.2_{\pm 3.1}$
Fast-TD-MPC (*RG*, τ_{p25})	$566_{\pm 104}$	$17.5_{\pm 2.6}$	$49_{\pm 11}$	$20.9_{\pm 1.9}$	$36_{\pm 22}$	$25.7_{\pm 4.7}$	$62_{\pm 55}$	$25.9_{\pm 25.6}$
Fast-TD-MPC (*RG*, τ_{p50})	$543_{\pm 102}$	$14.3_{\pm 2.5}$	$50_{\pm 10}$	$18.9_{\pm 2.1}$	$33_{\pm 22}$	$21.4_{\pm 5.6}$	$51_{\pm 61}$	$23.1_{\pm 23.6}$
Fast-TD-MPC (*RG*, τ_{p75})	$524_{\pm 102}$	$12.2_{\pm 2.3}$	$49_{\pm 10}$	$17.4_{\pm 2.2}$	$35_{\pm 24}$	$17.7_{\pm 5.3}$	$43_{\pm 60}$	$18.8_{\pm 14.7}$

Among TD-MPC2-derived baselines, reducing the planner's sample budget (N=512 $\rightarrow$ N=64) degrades return on DMControl by 22% and MyoSuite by 40%, while providing little latency reduction, as latency is dominated by fixed GPU overhead rather than sample count. S1 alone achieves the lowest latency but degrades significantly, particularly on ManiSkill2 (19% vs. 65%) and DMControl (623 vs. 820), confirming that pure amortization without a fallback is insufficient. The policy-only variant (no MPPI) is similarly fast but performs worse than S1 on most domains, confirming that supervised distillation from MPPI yields a stronger reactive policy than the TD-MPC2 actor alone.

Runtime breakdown. Table 2 details the per-component latency breakdown. The OOD gate adds negligible overhead ($\sim$0.2 ms), while System 1 is $\sim$65$\times$ cheaper per invocation than System 2 ($\sim$0.4 ms vs. $\sim$26 ms on H100). At τ_{p50}, ρ adapts to domain difficulty (81% on H100 vs. 66% on A100), and peak GPU memory remains bounded by S2-alone.

Latency scales linearly with planning frequency. Figure 3 confirms that per-step latency scales linearly with ρ across all domains ($R^2 \geq 0.93$), validating the runtime model: since the OOD detector costs negligible computation ($\sim$0.2 ms), total latency is determined by the fraction of timesteps routed to System 2. Both *all* and *RG* modes fall on the same line, confirming that τ controls only the operating point along this line, not the per-step cost structure.

4.3 Robustness Under Disturbances (***RQ2***)

We evaluate 102 of the 103 tasks under five types of externally applied disturbances, organized following the taxonomy of Gu et al. (2025) into three groups: (i) *observation* (observation noise: additive Gaussian on s_t), (ii) *action* (action noise: additive Gaussian on a_t), (iii) *environment* (velocity kick: instantaneous impulse on q_{vel}; gravity shift: sustained rescaling of gravity; sustained force: continuous injection with probability p=0.15). We report results for the *combined* setting, which applies all five disturbances simultaneously. All disturbances are applied post-training at evaluation time with per-episode stochasticity. See Appendix **E** for full details including the excluded task.

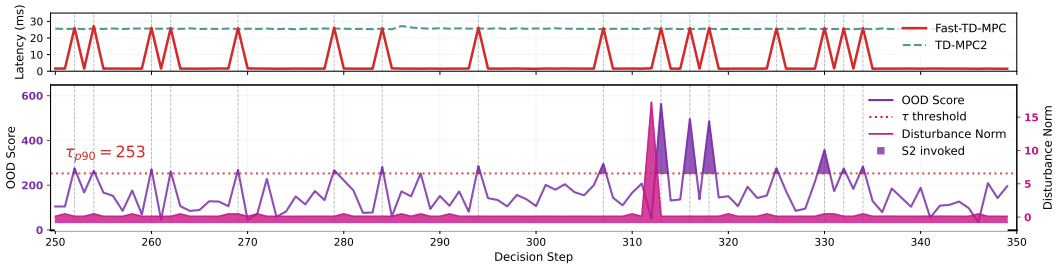


Figure 5: **OOD gating dynamics.** Per-step OOD score and cumulative latency for a representative episode of `walker-walk-backwards` under combined disturbances (DMControl, τ_{p90}). **Top:** Cumulative latency of Fast-TD-MPC (red) vs. TD-MPC2 (teal, dashed). **Bottom:** OOD score (purple, left axis) with τ_{p90} threshold (red dotted). Shaded regions and grey dashed lines mark timesteps where $g(z_t) > \tau$, indicating S2 invocation. Disturbance norm (pink, right axis) shows the total perturbation magnitude per step, with a spike at step 312 from the velocity kick.

We compare against (i) **S2-alone**, (ii) **S2 policy-only**, (iii) **S1 alone**, and (iv) **Round-robin** (50%), measuring absolute performance and degradation relative to the undisturbed setting. We present aggregate results across all 4 domains in Table 3 and Figure 4.

Speed–performance trade-off holds under disturbances. At τ_{p90}, Fast-TD-MPC (*all*) retains 95% of TD-MPC2's perturbed DMControl return (586 vs. 616) at 2× speedup, while on ManiSkill2, τ_{p50} achieves 66% success at 34.6 ms vs. TD-MPC2's 66% at 43.1 ms. On Meta-World, all methods degrade to a similar level (47–51%), leaving little room to separate them.

OOD gating preserves robustness. Perturbations degrade TD-MPC2 substantially (-25% return on DMControl, -39% success on Meta-World), and S1 alone even more (-35% and -46%), indicating that the amortized policy is more sensitive to distribution shift than the planner. At τ_{p50}, where the gate routes most perturbed steps to the planner (88% on DMControl, 99% on Meta-World), Fast-TD-MPC (*all*) tracks TD-MPC2 closely: 617 vs. 616 on DMControl, 51% vs. 51% on Meta-World, 66% vs. 66% on ManiSkill2, and 36% vs. 40% on MyoSuite. This robustness stems from adaptive routing rather than a fixed planning budget: at τ_{p90}, the S2 invocation rate rises from 18% to 45% on DMControl and from 39% to 82% on Meta-World under perturbation (Appendix **G**). Figure 5 illustrates this at the step level: a velocity kick at step 312 pushes $g(z_t)$ above τ, triggering S2 for several steps before the system resumes S1.

State-aware gating vs. fixed scheduling under disturbances. Under perturbation, Fast-TD-MPC matches or slightly exceeds RR 50% at a lower latency on DMControl (586 at 13.6 ms vs. 577 at 14.5 ms, τ_{p90}) and ManiSkill2 (62% at 25.9 ms vs. 58% at 27.1 ms, *RG* τ_{p25}), although differences lie within the 95% CIs. The key difference is adaptivity: the same τ_{p90} configuration runs at 6.5 ms per step on clean DMControl episodes and increases planning only when perturbed, whereas RR 50% spends 14.4 ms per step in both conditions and degrades more ($813 \rightarrow 577$, -29%, vs. $785 \rightarrow 586$, -25%). On Meta-World and MyoSuite, RR is on par with or better than gating at matched latency.

Reward-gated mode under disturbance. The reward-gated mode's efficiency advantage on Meta-World and MyoSuite (Section 4.2) persists under perturbation. On Meta-World, *RG* τ_{p50} and τ_{p75} reach 49-50% at 17.4-18.9 ms, faster than any *all* configuration (the fastest, τ_{p99}, reaches 50% at 20.2 ms). On MyoSuite, *RG* τ_{p75} achieves 35% at 17.7 ms, comparable to *all* τ_{p50} (36% at 32.2 ms) and above *all* τ_{p90} (34% at 22.9 ms). In both cases, gains come from fewer S2 invocations (e.g., 44% vs. 60–89% on MyoSuite), suggesting that reward-gating's tighter ID distribution produces a more precise threshold, reducing unnecessary S2 invocations on states System 1 can reliably handle.

4.4 Ablations

MPPI Planner Warm-Start Strategy. We compare cold starting the MPPI planner against K-shifted warm start, which salvages the first $H-K$ steps of the stale buffer when $K < H-1$ and cold-starts otherwise. A paired t-test across all task-config comparisons on all 4 benchmark domains finds no significant difference ($p > 0.05$) in return (t=1.31, p=0.19) or S2 invocation rate

(t=−0.28, p=0.78). Under disturbance, return (t=−0.78, p=0.44) and S2 rate (t=1.35, p=0.18) remain insignificant. We adopt cold start for its simplicity. See Appendix **I** for full results.

OOD distance metric. We compare Mahalanobis and Tanimoto distance (Rogers & Tanimoto, 1960) for OOD gating. A paired t-test across all task-config comparisons finds no significant difference in nominal return (t=0.85, p=0.40). In the *theoretical* mode, Tanimoto yields equal or lower latency at every τ percentile, but degrades robustness: under perturbation, Mahalanobis achieves significantly higher return (t=2.33, p=0.02) by routing more timesteps to the planner. We therefore adopt Mahalanobis distance as the OOD gating. See Appendix **J** for full results.

5 Limitations and Discussions

We identify three main limitations. First, the OOD gate is *reactive*. It detects atypical states only after they are visited, rather than anticipating them from the learned dynamics model. A predictive gate could pre-route to System 2 before state degradation occurs, further reducing wasted computation on unrecoverable trajectories. Second, while the default τ is set by a fixed rule on held-out OOD scores, the optimal threshold varies across domains (e.g., locomotion vs. manipulation); real-world deployment may require tracing the Pareto frontier to identify a task-appropriate operating point. Finally, all experiments are conducted in simulation, while sim-to-real gaps exist for realistic robotic systems; we defer real-robot validation to future work.

6 Related Work

Data-Driven MPC and TD-MPC2. TD-MPC2 (Hansen et al., 2024) combines latent dynamics modeling with MPPI planning and TD-learning, achieving state-of-the-art performance across 104 continuous control tasks, outperforming SAC (Haarnoja et al., 2018) and DreamerV3 (Hafner et al., 2025). Its ecosystem addresses training efficiency (Evers et al., 2026), humanoid locomotion (Nguyen et al., 2025; 2026b), and world model reliability (Hansen & Wang, 2026). Fast-TD-MPC builds on this framework but targets *inference-time* acceleration.

Accelerating Test-Time Planning. Approaches to reducing the per-step cost of MPPI fall into four categories: (i) *Model reduction* distills the planner into a smaller or quantized model, but MPPI still runs at every step (Kuzmenko & Shvai, 2025; Wang et al., 2025); (ii) *Planner acceleration* speeds up the planning loop itself (e.g., parallelizing latent rollouts (Gao & Xu, 2026)) but does not eliminate it; (iii) *Full amortization* replaces the planner entirely with a feed-forward policy (Nguyen et al., 2026a), achieving large speedups but providing no fallback when the policy encounters out-of-distribution states; (iv) *Conditional replanning* skips planning on some timesteps: Cheng et al. (2026) adapt replanning tolerance based on model mismatch by reusing cached plans, while Lin et al. (2025b) generate action queues from the planner's own latent rollouts with full replanning on mismatch. Fast-TD-MPC differs by combining *separately amortized* full policy replacement with OOD-gated fallback to the complete planner. The amortized policy is distilled from expert planner rollouts, so the planning expertise is retained in the fast path, and only the iterative optimization at inference time is eliminated. Classical event-triggered control (Tabuada, 2007; Heemels et al., 2012) and the dual-process framework (Kahneman, 2011; Daw et al., 2005; Botvinick et al., 2009) provide broader intellectual context. See Appendix **L** for a detailed comparison.

7 Conclusion

We presented Fast-TD-MPC, a framework that accelerates TD-MPC2 inference by routing in-distribution states to a compact amortized policy and falling back to the full MPPI planner only when the OOD detector flags atypical states. Across 103 continuous control tasks spanning four domains, Fast-TD-MPC achieves up to $\sim$4$\times$ inference speedup ($\sim$2.7$\times$ on average) with $<$10% performance drop on 3 of 4 domains, using a single set of hyperparameters. Under five combined external disturbance types, the gate adaptively increases planning on perturbed states, retaining 95% of TD-MPC2's perturbed return on DMControl at half its latency. The framework is conceptually general: any model-based planner can serve as System 2, any amortized policy as System 1, and the gating decision reduces to a single transparent scalar comparison, a desirable property for safety-critical applications where auditability matters.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of the manuscript, including drafting, editing, and formatting assistance. We also used generative AI tools for information retrieval, including searching for and summarizing related work to ensure comprehensive coverage of the literature. We *did not* use generative AI tools for designing the method, conducting experiments, analyzing results, or producing the scientific claims of this work. All experimental results, tables, and figures are from actual experimental data collected by the authors.

REPRODUCIBILITY STATEMENT

Details of hyperparameters are described in the appendix. We will open-source all code and relevant trained model weights upon acceptance.

REFERENCES

Matthew M. Botvinick, Yael Niv, and Andrew G. Barto. Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, 113(3):262–280, 2009.

Gabriel Bravo-Palacios, Jianghan Zhang, Zachary Pestrikov, Brian Plancher, and Thomas Lew. Turbompc: Fast, scalable, and differentiable model predictive control on the GPU. *arXiv preprint arXiv:2606.24039*, 2026.

Lara Brudermüller, Brandon Hung, Xinghao Zhu, Jiuguang Wang, Nick Hawes, Preston Culbertson, and Simon Le Cleac'h. Generative models from and for sampling-based MPC: A bootstrapped approach for adaptive contact-rich manipulation. *IEEE Robotics and Automation Letters*, 2026. doi: 10.1109/LRA.2026.3655193.

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite – a contact-rich simulation suite for musculoskeletal motor control. *arXiv preprint arXiv:2205.13600*, 2022.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Josef Chen. AEGIS: A backup reflex for physical AI. *arXiv preprint arXiv:2606.06660*, 2026.

Yutian Cheng, Xiaojian Ma, Xianhao Wang, Min Yang, Rongpeng Su, Hangxin Liu, Xi Chen, Shuai Li, and Qing Li. AdaReP: Adaptive re-planning under model mismatch for neural world-model predictive control. In *Artificial Neural Networks and Machine Learning – ICANN 2026*, Lecture Notes in Computer Science. Springer, 2026. doi: 10.1007/978-3-032-38404-1_18.

Rohan Chitnis, Yingchen Xu, Bobak Hashemi, Lucas Lehnert, Urun Dogan, Zheqing Zhu, and Olivier Delalleau. IQL-TD-MPC: Implicit Q-learning for hierarchical model predictive control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Inkook Chun, Seungjae Lee, Michael S. Albergo, Saining Xie, and Eric Vanden-Eijnden. Dynamic test-time compute scaling in control policy: Difficulty-aware stochastic interpolant policy. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Nathaniel D. Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature Neuroscience*, 8(12):1704–1711, 2005.

Shengjun Kris Dong, Dima Nikiforov, Widyadewi Soedarmadji, Minh Nguyen, Vikram Jain, Christopher W. Fletcher, and Yakun Sophia Shao. Characterizing and optimizing real-time optimal control for embedded SoCs. In *2025 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 489–503, 2025. doi: 10.1109/iiswc66894.2025.00047.

Thomas Evers, Cristian Meo, Wendelin Böhmer, Justin Dauwels, and Yaniv Oren. EfficientTDMPC: Improved MPC objectives for sample-efficient continuous control. *arXiv preprint arXiv:2605.16692*, 2026.

David Fridovich-Keil, Sylvia L. Herbert, Jaime F. Fisac, Sampada Deglurkar, and Claire J. Tomlin. Planning, fast and slow: A framework for adaptive real-time safe trajectory planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Siddhant Gangapurwala, Luigi Campanaro, and Ioannis Havoutis. Learning low-frequency motion control for robust and dynamic robot locomotion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Yuntian Gao and Xiangyu Xu. Fast LeWorldModel. *arXiv preprint arXiv:2606.26217*, 2026.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

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

Shangding Gu, Laixi Shi, Muning Wen, Ming Jin, Eric Mazumdar, Yuejie Chi, Adam Wierman, and Costas Spanos. Robust gymnasium: A unified modular benchmark for robust reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Yanchen Guan, Haicheng Liao, Zhenning Li, Jia Hu, Runze Yuan, Guohui Zhang, and Chengzhong Xu. World models for autonomous driving: An initial survey. *IEEE Transactions on Intelligent Vehicles*, 2024.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

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, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *Nature*, 640:647–653, 2025.

Nicklas Hansen and Xiaolong Wang. Hallucination in world models is predictable and preventable. *arXiv preprint arXiv:2606.27326*, 2026.

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

Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022.

Kazumune Hashimoto, Shuichi Adachi, and Dimos V. Dimarogonas. Event-triggered intermittent sampling for nonlinear model predictive control. *Automatica*, 81:148–155, 2017.

W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *IEEE Conference on Decision and Control (CDC)*, pp. 3270–3285, 2012.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Hsiang-Wei Huang, Jianxu Shangguan, Junbin Lu, and Jenq-Neng Hwang. Leflow: Generative latent flow planning for world models. *arXiv preprint arXiv:2608.24855*, 2026.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

Ramneet Kaur, Kaustubh Sridhar, Sangdon Park, Yahan Yang, Susmit Jha, Anirban Roy, Oleg Sokolsky, and Insup Lee. CODiT: Conformal out-of-distribution detection in time-series data for cyber-physical systems. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (CPS-IoT Week)*, 2023.

Dmytro Kuzmenko and Nadiya Shvai. TD-MPC-Opt: Distilling model-based multi-task reinforcement learning agents. *arXiv preprint arXiv:2507.01823*, 2025.

Adam Labiosa and Josiah P. Hanna. When to plan: Learning to select between reactive control and deliberative planning. In *Reinforcement Learning Conference (RLC)*, 2026.

Samuel Lavoie, Christos Tsirigotis, Max Schwarzer, Ankit Vani, Michael Noukhovitch, Kenji Kawaguchi, and Aaron Courville. Simplicial embeddings in self-supervised learning and downstream classification. *arXiv preprint arXiv:2204.00616*, 2022.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Yaniv Leviathan, Matan Kalman, and Yossie Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, 2023.

Andrew Zou Li, Gokul Swamy, Yonatan Bisk, and Andrea Bajcsy. ELASTIC: Efficiently learning to adaptively scale test-time compute for generative control policies. *arXiv preprint arXiv:2606.31132*, 2026a.

Wenhao Li, Xiu Su, Yichao Cao, Hongyan Xu, Xiaobo Xia, Shan You, Yi Chen, and Chang Xu. VLA-ATTC: Adaptive test-time compute for VLA models with relative action critic model. *arXiv preprint arXiv:2605.01194*, 2026b.

Haotian Lin, Pengcheng Wang, Jeff Schneider, and Guanya Shi. TD-M(PC)^2: Improving temporal difference MPC through policy constraint. *arXiv preprint arXiv:2502.03550*, 2025a.

Ziyang Lin, Zixuan Sun, Sanhorn Chen, Xiaoyang Chen, and Roy Zhao. Accelerating multimodal LLM gaming performance via input prediction and mishit correction. *arXiv preprint arXiv:2512.17250*, 2025b.

Ning Liu, Sen Shen, Zheng Li, Matthew D'Souza, Jen Jen Chung, and Thomas Braunl. Adaptive reinforcement and model predictive control switching for safe human-robot cooperative navigation. *arXiv preprint arXiv:2601.16686*, 2026a.

Yongzhi Liu, Yang Xiao, Zhong Cao, Zeng Kang, Sunan Zhang, Zhaozhi Dong, Guojun Yu, and Weichao Zhuang. LIDAR-AD: A decoder-free latent-interaction dreamer with action-residual chains for autonomous driving. *arXiv preprint arXiv:2607.11964*, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

Emil Mittag, Richard Dazeley, and Peter Vamplew. OOD-RL-Bench: A benchmark framework for out-of-distribution detection in reinforcement learning. *arXiv preprint arXiv:2607.12523*, 2026.

Aneesh Muppidi, Firas Darwish, Dylan Cope, João F. Henriques, and Jakob Nicolaus Foerster. Finding the time to think: Learning planning budgets in real-time RL. *arXiv preprint arXiv:2606.26463*, 2026.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Rudy R. Negenborn, Bart De Schutter, Marco A. Wiering, and Hans Hellendoorn. Learning-based model predictive control for Markov decision processes. *IFAC Proceedings Volumes*, 38(1):354–359, 2005. 16th IFAC World Congress.

Hoang Nguyen, Xiaohao Xu, and Xiaonan Huang. Latent geometry beyond search: Amortizing planning in world models. *arXiv preprint arXiv:2605.08732*, 2026a.

Khang Nguyen, Khai Nguyen, An T. Le, Jan Peters, Manfred Huber, Ngo Anh Vien, and Minh Nhat Vu. TD-GRPC: Temporal difference learning with group relative policy constraint for humanoid locomotion. *arXiv preprint arXiv:2505.13549*, 2025.

Khang Nguyen, An T. Le, Jan Peters, and Minh Nhat Vu. DoublyAware: Dual planning and policy awareness for temporal difference learning in humanoid locomotion. *IEEE Robotics and Automation Letters*, 2026b.

Mohit Prashant, Arvind Easwaran, Suman Das, and Michael Yuhas. Guaranteeing out-of-distribution detection in deep RL via transition estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Martin L. Puterman. Markov decision processes. In D. P. Heyman and M. J. Sobel (eds.), *Handbooks in Operations Research and Management Science*, volume 2 of *Stochastic Models*, pp. 331–434. North-Holland, Amsterdam, 1990.

David J. Rogers and Taffee T. Tanimoto. A computer program for classifying plants. *Science*, 132 (3434):1115–1118, 1960.

Rohit Kumar Salla, Manoj Saravanan, and Simon Stepputtis. Learning when to stop: Prefix-optimal dynamic diffusion policies for continuous control. *arXiv preprint arXiv:2608.05084*, 2026.

Aidan Scannell, Mohammadreza Nakhaeinezhadfard, Kalle Kujanpää, Yi Zhao, Kevin Sebastian Luck, Arno Solin, and Joni Pajarinen. Discrete codebook world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2025.

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.

Jonathan Spieler and Sven Behnke. Dream-MPC: Gradient-based model predictive control with latent imagination. In *International Conference on Machine Learning (ICML)*, 2026.

Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

Jinhe Tang and Weiming Zhi. AutoIntervene: Calibrated intervention for action-chunking imitation learning policies. *arXiv preprint arXiv:2608.07065*, 2026.

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. Technical report, DeepMind, 2018.

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. IEEE, 2012.

Yuhang Wang, Hanwei Guo, Sizhe Wang, Long Qian, and Xuguang Lan. Bootstrapped model predictive control. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zihua Wang, Zhitao Lin, Ruibo Li, Yu Zhang, Xu Yang, Siya Mi, and Xiu-Shen Wei. Open-loop planning, closed-loop verification: Speculative verification for VLA. *arXiv preprint arXiv:2604.02965*, 2026.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanwen Jiang, Yi Yuan, He Wang, and Li Yi. SAPIEN: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

Jiayi Yan, Francesco Fabiano, and Alessandro Abate. Dual process motion planning. *arXiv preprint arXiv:2609.01260*, 2026.

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)*, 2019.

Hongming Zhang, Ke Sun, Bo Xu, Linglong Kong, and Martin Müller. A distance-based anomaly detection framework for deep reinforcement learning. *Transactions on Machine Learning Research (TMLR)*, 2024.

Zesen Zhao, Minkyoung Cho, Hui Shen, Boyuan Zheng, Kunxiao Gao, Yulong Cao, and Z. Morley Mao. Test-time scaling for world action models via zero-shot geometric evaluation. *arXiv preprint arXiv:2607.17454*, 2026.

APPENDIX

The appendix is organized as follows. Appendix A provides method details including the inference algorithm. Appendix B–E describe the task domains, environment details, and disturbance protocol. Appendices F–G report full results including S2 invocation rate and GPU memory. Appendix H details the evaluation setup and hardware. Appendices I–J present ablations on the MPPI warm-start strategy and OOD distance metric. Appendix K provides implementation details, and Appendix L extends the related work discussion.

A METHOD DETAILS

Algorithm 1 summarizes the full Fast-TD-MPC inference procedure. At each decision step t, the frozen encoder $\mathcal{E}_\theta$ maps the current state s_t to latent z_t, and the OOD detector computes the Mahalanobis distance $g(z_t)$. If $g(z_t) \leq \tau$, the state is deemed in-distribution and the amortized policy π_{S1} produces the action in a single forward pass. Otherwise, the full MPPI planner π_{S2} is invoked. All components are frozen at deployment; the only runtime computation is the encoding, OOD scoring, and the selected policy's forward pass.

Algorithm 1 Fast-TD-MPC inference

Require: Frozen encoder $\mathcal{E}_\theta$, amortized policy $\pi_{\text{S1}}(\cdot\,;\phi)$, MPPI planner $\pi_{\text{S2}}(\cdot\,;\theta)$, OOD detector $g(\cdot\,;\bar{z},\Sigma^{-1})$, threshold τ

$s_0 \sim p_0$
for $t = 0, 1, \ldots, T_{\max} - 1$ **do**
  $z_t \leftarrow \mathcal{E}_\theta(s_t)$ ▷ *Encode state*
  $g_t \leftarrow g(z_t\,;\bar{z},\Sigma^{-1})$ ▷ *OOD score*
  if $g_t \leq \tau$ **then**
    $a_t \leftarrow \pi_{\text{S1}}(z_t;\phi)$ ▷ *System 1: fast policy*
  else
    $a_t \leftarrow \pi_{\text{S2}}(z_t;\theta)$ ▷ *System 2: MPPI planner*
  Execute a_t, observe s_{t+1}

B TASK VISUALIZATIONS

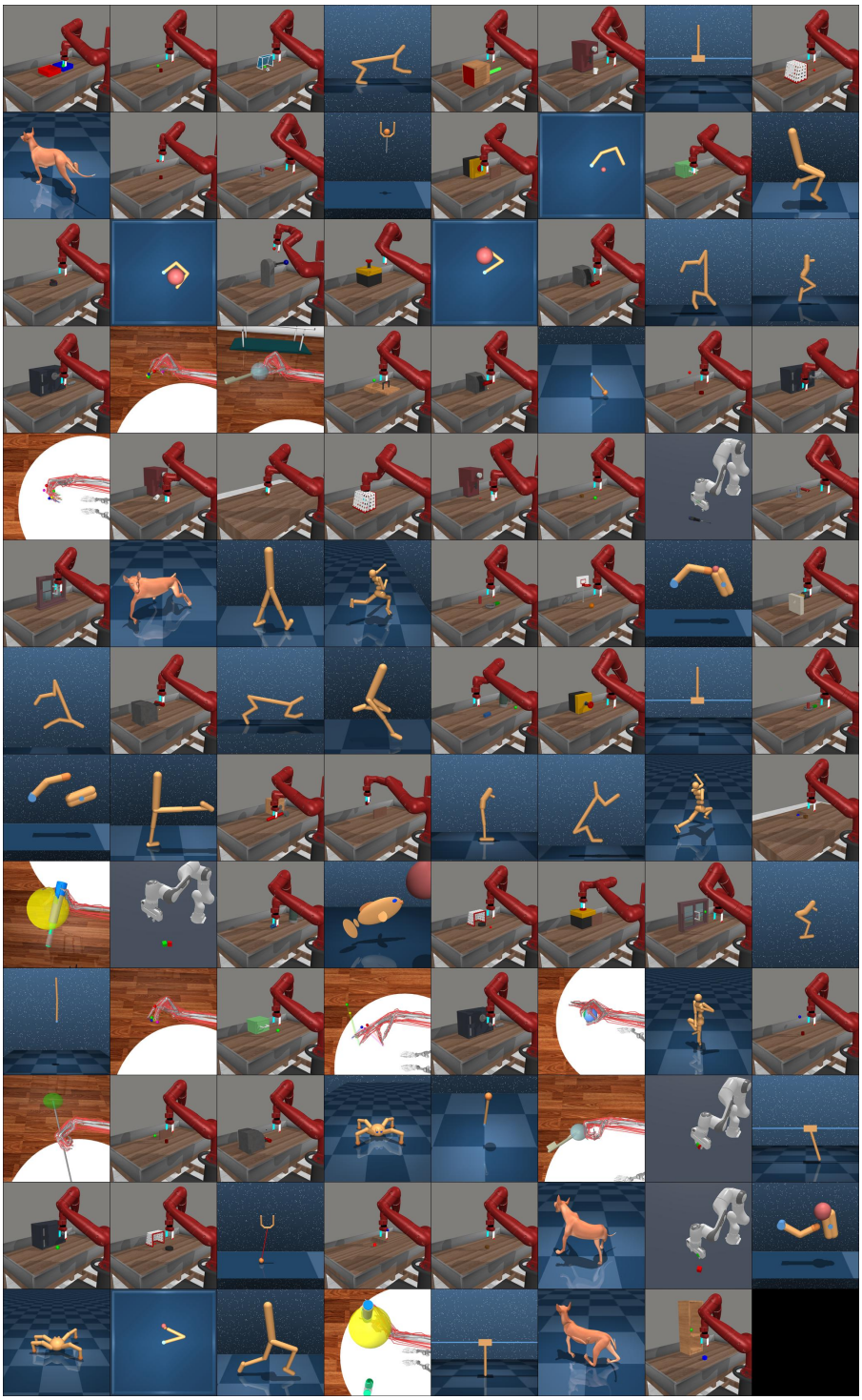


Figure 6: **Task visualizations.** Visualization of random states from all 103 diverse continuous control tasks used in our evaluation, spanning DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), MyoSuite (Caggiano et al., 2022), and ManiSkill2 (Gu et al., 2023). Tasks vary greatly in objective, embodiment, and action space. See Appendix **C** for task details.

C TASK DOMAINS

We consider a total of 103 continuous control tasks from 4 task domains: DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2019), MyoSuite (Caggiano et al., 2022), and ManiSkill2 (Gu et al., 2023). We provide an exhaustive list of all tasks considered, along with their details in Table 4, 5, 6 and 7 below. Visualizations of each task are provided in Appendix **B**. Environment details are provided in Appendix **D** and Appendix **E**.

Table 4: **DMControl.** We consider a total 39 tasks from the DMControl domain. Observation dimension ranges from 3 - 223, with action dimensions ranging from 1 - 38. Metric: Return. Physics engine: MuJoCo.

Task	Obs. dim	Action dim
Acrobot Swingup	6	1
Cartpole Balance	5	1
Cartpole Balance Sparse	5	1
Cartpole Swingup	5	1
Cartpole Swingup Sparse	5	1
Cheetah Jump	17	6
Cheetah Run	17	6
Cheetah Run Back	17	6
Cheetah Run Backwards	17	6
Cheetah Run Front	17	6
Cup Catch	8	2
Cup Spin	8	2
Dog Run	223	38
Dog Stand	223	38
Dog Trot	223	38
Dog Walk	223	38
Finger Spin	9	2
Finger Turn Easy	12	2
Finger Turn Hard	12	2
Fish Swim	24	5
Hopper Hop	15	4
Hopper Hop Backwards	15	4
Hopper Stand	15	4
Humanoid Run	67	21
Humanoid Stand	67	21
Humanoid Walk	67	21
Pendulum Spin	3	1
Pendulum Swingup	3	1
Quadruped Run	78	12
Quadruped Walk	78	12
Reacher Easy	6	2
Reacher Hard	6	2
Reacher Three Easy	8	3
Reacher Three Hard	8	3
Walker Run	24	6
Walker Run Backwards	24	6
Walker Stand	24	6
Walker Walk	24	6
Walker Walk Backwards	24	6

Table 5: **Meta-World.** We consider a total 50 tasks from the Meta-World domain. Observation dimensions are fixed at 39, with action dimensions fixed at 4. Metric: Success rate. Physics engine: MuJoCo.

Task	Obs. dim	Action dim
Assembly	39	4
Basketball	39	4
Bin Picking	39	4
Box Close	39	4
Button Press	39	4
Button Press Topdown	39	4
Button Press Topdown Wall	39	4
Button Press Wall	39	4
Coffee Button	39	4
Coffee Pull	39	4
Coffee Push	39	4
Dial Turn	39	4
Disassemble	39	4
Door Close	39	4
Door Lock	39	4
Door Open	39	4
Door Unlock	39	4
Drawer Close	39	4
Drawer Open	39	4
Faucet Close	39	4
Faucet Open	39	4
Hammer	39	4
...	...	...
Window Open	39	4

Table 6: **MyoSuite.** We consider a total 10 tasks from the MyoSuite domain. Observation dimensions ranges from 83 - 115, with action dimensions fixed at 39. Metric: Success rate. Physics engine: MuJoCo.

Task	Obs. dim	Action dim
Myo Hand Key Turn	93	39
Myo Hand Key Turn Hard	93	39
Myo Hand Obj Hold	91	39
Myo Hand Obj Hold Hard	91	39
Myo Hand Pen Twirl	83	39
Myo Hand Pen Twirl Hard	83	39
Myo Hand Pose	108	39
Myo Hand Pose Hard	108	39
Myo Hand Reach	115	39
Myo Hand Reach Hard	115	39

Table 7: **ManiSkill2.** We consider a total 4 tasks from the ManiSkill2 domain. Observation dimensions ranges from 42 - 55, with action dimensions ranging from 4 - 7. Metric: Success rate. Physics engine: SAPIEN.

Task	Obs. dim	Action dim
Lift Cube	42	4
Pick Cube	51	4
Pick Ycb	51	7
Stack Cube	55	4

D Environment Details

For experiments on ***RQ1***, we follow Hansen et al. (2024)'s setup and benchmark on DMControl, Meta-World, ManiSkill2, and MyoSuite without modification. All four domains are infinite-horizon continuous control environments with fixed episode lengths and no termination conditions. We list episode lengths, action repeats, and the performance metric used for each domain in Table 8. Following Hansen et al. (2024), we adopt a strict success criterion: an episode is successful only if its final step is successful. In object manipulation tasks, this ensures that an episode in which an object is picked up but subsequently dropped is not counted as a success.

Table 8: **Environment details.** We list the episode length and action repeat used for each task domain, as well as the total number of environment steps and performance metrics that we use for benchmarking methods. All methods use the same values for all tasks.

	DMControl	**Meta-World**	**ManiSkill2**	**MyoSuite**
Episode length	1,000	200	200	100
Action repeat	2	2	2	1
Effective length	500	100	100	100
Performance metric	Reward	Success	Success	Success

E Environment with Externally Applied Disturbances

For experiments on ***RQ2***, we introduce five types of disturbances to all four benchmark domains. Table 9 details the five disturbance types, organized following the taxonomy of Gu et al. (2025) into observation, action, and environment disruptors, along with their categories and activation probabilities. All disturbances are applied post-training at evaluation time with per-episode stochasticity (seed = 42); per-episode parameters are drawn from the specified ranges using an isolated `RandomState`.

Table 9: **Environment disturbance configurations.** p denotes the per-step activation probability. †1–2 kicks per episode at random timesteps within 10–90% of episode length. Obs. noise and vel. kick magnitudes are relative (scaled by per-episode std). Tasks: 102 (39 DMControl + 50 MetaWorld + 10 MyoSuite + 3 ManiSkill2; `pick-ycb` excluded).

Disturbance	**Cat.**	p	**Mechanism**	**Parameter**
Obs. noise	Obs.	1.0	Relative $\sigma \cdot \text{std}(s)$ on s_t	$\sigma = .005$
Act. noise	Act.	1.0	Additive $\mathcal{N}(0, \sigma^2)$ on a_t, clamp $[-1, 1]$	$\sigma = .005$
Vel. kick	Env.	$1.0^{\dagger}$	Impulse on q_{vel} at 1–2 random t	$m_{\max} = .3$
Grav. shift	Env.	1.0	Sustained $g \leftarrow g \cdot k$ per episode	$\delta = .05$
Sus. force	Env.	0.15	Per-step force injection when active	$f_{\max} = .3$
Combined	All	—	All five applied simultaneously	—

Excluded Task. The `pick-ycb` task (PickSingleYCB-v0) from ManiSkill2 is excluded from the perturbation evaluation due to a SAPIEN (Xiang et al., 2020) memory management bug that triggers a C-level segfault when physics-based disturbances are applied to articulated YCB mesh objects. This is a known SAPIEN limitation unrelated to our method. All other ManiSkill2 tasks are unaffected.

SAPIEN Backend Adaptation. ManiSkill2 environments are built on SAPIEN (Xiang et al., 2020), which exposes a different physics API than MuJoCo (Todorov et al., 2012). While observation noise and action noise operate on the observation and action arrays respectively (backend-agnostic), three perturbation types require adaptation: (i) *Gravity shift:* SAPIEN's `SceneConfig.gravity` is a direct 3D vector, equivalent to MuJoCo's `model.opt.gravity`. (ii) *Velocity kick:* applied via `set_qvel(get_qvel() +`

`impulse)`. (iii) *Sustained force:* SAPIEN does not expose `qfrc_applied`; we implement this as per-step velocity injection, `set_qvel(get_qvel() + f)`, producing equivalent trajectory disruption with the same magnitude and direction sampling.

F Full Results

Tables 10 – 13 present the full benchmark results for each domain, including per-step latency, S2 invocation rate (ρ), and peak GPU memory, complementing Table 1. Tables 14 – 17 provide per-task breakdowns for four key methods.

Table 10: **DMControl full results.** 39 tasks, 3 seeds each. Mean and 95% CI across all tasks. Return, per-step latency, S2 invocation rate (ρ), and peak GPU memory. †Referenced from Hansen et al. (2024). *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Return**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
DreamerV3†	$735_{\pm 93}$	$3.2_{\pm .1}$	—	—
SAC†	$631_{\pm 108}$	$2.0_{\pm .4}$	—	—
S2-alone (TD-MPC2)	$820_{\pm 55}$	$27.3_{\pm 1.1}$	100	$95_{\pm 0}$
S2-alone (N=64)	$639_{\pm 100}$	$26.9_{\pm 1.1}$	100	$69_{\pm 0}$
S2 policy-only	$560_{\pm 105}$	$1.9_{\pm .3}$	0	$66_{\pm 0}$
S1 alone	$623_{\pm 91}$	$1.6_{\pm .2}$	0	$68_{\pm 0}$
Round-robin (S2 50%)	$813_{\pm 57}$	$14.4_{\pm .7}$	50	$96_{\pm 0}$
Round-robin (S2 33%)	$782_{\pm 66}$	$10.1_{\pm .5}$	33	$68_{\pm 0}$
Round-robin (S2 20%)	$743_{\pm 74}$	$6.6_{\pm .4}$	20	$96_{\pm 0}$
Round-robin (S2 10%)	$699_{\pm 83}$	$4.0_{\pm .3}$	10	$96_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$829_{\pm 55}$	$19.4_{\pm 2.4}$	$68_{\pm 7}$	$85_{\pm 3}$
Fast-TD-MPC (*all*, τ_{p75})	$823_{\pm 55}$	$12.2_{\pm 2.7}$	$40_{\pm 9}$	$77_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p90})	$785_{\pm 58}$	$6.5_{\pm 2.0}$	$18_{\pm 7}$	$73_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p95})	$745_{\pm 74}$	$3.8_{\pm 1.4}$	$8_{\pm 4}$	$70_{\pm 1}$
Fast-TD-MPC (*all*, τ_{p99})	$650_{\pm 92}$	$1.9_{\pm .3}$	$1_{\pm 0}$	$69_{\pm 0}$
Fast-TD-MPC (*RG*, τ_{p25})	$784_{\pm 68}$	$10.8_{\pm 2.3}$	$35_{\pm 8}$	$77_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p50})	$764_{\pm 70}$	$8.1_{\pm 1.8}$	$25_{\pm 7}$	$74_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p75})	$744_{\pm 74}$	$6.4_{\pm 1.5}$	$18_{\pm 5}$	$72_{\pm 2}$

G Full Results under Disturbances

Tables 18 – 21 present the full results under disturbances for each domain, including per-step latency, S2 invocation rate (ρ), and peak GPU memory, complementing Table 3. Tables 22 – 25 provide corresponding per-task breakdowns.

H Evaluation Setup and Hardware Details

Evaluation Metrics. We follow Hansen et al. (2024)'s setup of training on 3 seeds and evaluating on the same 3 seeds, with 1 warmup + 10 evaluation episodes per task per seed. We report return/success rate, per-step latency (ms), S2 invocation rate (ρ), and peak GPU memory.

Hardware. Per-step latency is measured with GPU synchronization at the start and end of each decision step (2 syncs), on a single CPU core with a dedicated empty GPU. DMControl, Meta-World, and MyoSuite run on an H100 node (2× Intel Xeon Gold 6430 CPUs, 2 TB RAM, and 8× NVIDIA H100 80GB GPUs); ManiSkill2 on an A100 workstation (AMD EPYC 7V13 CPU, 216 GB RAM, 1× A100 80GB).

Table 11: **Meta-World full results.** 50 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. †Referenced from Hansen et al. (2024). *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
DreamerV3†	$70_{\pm 8}$	$3.8_{\pm .1}$	—	—
SAC†	$84_{\pm 7}$	$2.0_{\pm .2}$	—	—
S2-alone (TD-MPC2)	$84_{\pm 9}$	$26.9_{\pm .2}$	100	$95_{\pm 0}$
S2-alone (N=64)	$85_{\pm 7}$	$26.7_{\pm .2}$	100	$69_{\pm 0}$
S2 policy-only	$82_{\pm 8}$	$2.7_{\pm .2}$	0	$66_{\pm 0}$
S1 alone	$87_{\pm 8}$	$2.5_{\pm .2}$	0	$67_{\pm 0}$
Round-robin (S2 50%)	$86_{\pm 7}$	$14.7_{\pm .2}$	50	$96_{\pm 0}$
Round-robin (S2 33%)	$86_{\pm 8}$	$10.7_{\pm .2}$	33	$67_{\pm 0}$
Round-robin (S2 20%)	$85_{\pm 8}$	$7.3_{\pm .2}$	20	$96_{\pm 0}$
Round-robin (S2 10%)	$86_{\pm 8}$	$4.9_{\pm .2}$	10	$96_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$85_{\pm 8}$	$24.8_{\pm 1.1}$	$90_{\pm 4}$	$95_{\pm 1}$
Fast-TD-MPC (*all*, τ_{p75})	$85_{\pm 8}$	$18.7_{\pm 1.5}$	$66_{\pm 6}$	$92_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p90})	$85_{\pm 8}$	$12.2_{\pm 1.6}$	$39_{\pm 6}$	$90_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p95})	$85_{\pm 8}$	$9.8_{\pm 1.5}$	$29_{\pm 6}$	$88_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p99})	$85_{\pm 8}$	$6.2_{\pm 1.3}$	$15_{\pm 5}$	$85_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p25})	$85_{\pm 8}$	$12.3_{\pm 1.7}$	$40_{\pm 7}$	$85_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p50})	$85_{\pm 8}$	$10.0_{\pm 1.5}$	$30_{\pm 6}$	$80_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p75})	$84_{\pm 8}$	$8.6_{\pm 1.5}$	$25_{\pm 6}$	$79_{\pm 2}$

Table 12: **MyoSuite full results.** 10 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. †Referenced from Hansen et al. (2024). *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
DreamerV3†	$14_{\pm 17}$	$4.2_{\pm .1}$	—	—
SAC†	$38_{\pm 20}$	$2.9_{\pm .5}$	—	—
S2-alone (TD-MPC2)	$40_{\pm 20}$	$35.5_{\pm .4}$	100	$97_{\pm 0}$
S2-alone (N=64)	$24_{\pm 16}$	$35.4_{\pm .4}$	100	$70_{\pm 0}$
S2 policy-only	$18_{\pm 15}$	$3.8_{\pm .4}$	0	$67_{\pm 0}$
S1 alone	$35_{\pm 16}$	$3.6_{\pm .3}$	0	$69_{\pm 0}$
Round-robin (S2 50%)	$45_{\pm 21}$	$19.3_{\pm .3}$	50	$98_{\pm 0}$
Round-robin (S2 33%)	$41_{\pm 18}$	$14.1_{\pm .3}$	33	$68_{\pm 0}$
Round-robin (S2 20%)	$40_{\pm 22}$	$9.8_{\pm .3}$	20	$98_{\pm 0}$
Round-robin (S2 10%)	$40_{\pm 21}$	$6.6_{\pm .3}$	10	$98_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$39_{\pm 21}$	$31.4_{\pm 3.6}$	$88_{\pm 12}$	$91_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p75})	$37_{\pm 21}$	$26.6_{\pm 5.8}$	$72_{\pm 18}$	$93_{\pm 4}$
Fast-TD-MPC (*all*, τ_{p90})	$37_{\pm 21}$	$22.6_{\pm 7.6}$	$60_{\pm 24}$	$85_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p95})	$38_{\pm 23}$	$19.8_{\pm 7.6}$	$51_{\pm 24}$	$82_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p99})	$28_{\pm 17}$	$10.9_{\pm 5.8}$	$23_{\pm 19}$	$78_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p25})	$39_{\pm 22}$	$24.5_{\pm 4.6}$	$66_{\pm 15}$	$89_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p50})	$35_{\pm 21}$	$20.6_{\pm 5.2}$	$54_{\pm 17}$	$85_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p75})	$39_{\pm 21}$	$15.7_{\pm 4.7}$	$39_{\pm 15}$	$83_{\pm 5}$

I MPPI Planner Warm-Start Strategy Ablation

When the OOD gate triggers a System 1 to System 2 transition after K timesteps, the planner's warm-start buffer (`_prev_mean`) contains a plan that is K steps stale. We consider two initialization strategies:

Table 13: **ManiSkill2 full results.** 4 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. †Referenced from Hansen et al. (2024). *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
DreamerV3†	$34_{\pm 56}$	$8.7_{\pm .7}$	—	—
SAC†	$22_{\pm 54}$	$5.8_{\pm .7}$	—	—
S2-alone (TD-MPC2)	$65_{\pm 54}$	$45.9_{\pm 4.8}$	100	$70_{\pm 0}$
S2-alone (N=64)	$63_{\pm 53}$	$41.5_{\pm 3.4}$	100	$44_{\pm 0}$
S2 policy-only	$48_{\pm 56}$	$11.1_{\pm 2.5}$	0	$41_{\pm 0}$
S1 alone	$19_{\pm 39}$	$10.7_{\pm 3.9}$	0	$42_{\pm 0}$
Round-robin (S2 50%)	$60_{\pm 44}$	$28.4_{\pm 5.6}$	50	$72_{\pm 0}$
Round-robin (S2 33%)	$42_{\pm 50}$	$22.3_{\pm 3.8}$	33	$42_{\pm 0}$
Round-robin (S2 20%)	$38_{\pm 48}$	$17.6_{\pm 3.9}$	20	$72_{\pm 0}$
Round-robin (S2 10%)	$28_{\pm 46}$	$14.5_{\pm 4.7}$	10	$72_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$54_{\pm 55}$	$33.9_{\pm 12.1}$	$67_{\pm 32}$	$63_{\pm 8}$
Fast-TD-MPC (*all*, τ_{p75})	$47_{\pm 54}$	$23.3_{\pm 10.7}$	$34_{\pm 26}$	$46_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p90})	$32_{\pm 42}$	$16.0_{\pm 5.2}$	$13_{\pm 9}$	$46_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p95})	$28_{\pm 46}$	$14.2_{\pm 3.9}$	$8_{\pm 9}$	$46_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p99})	$20_{\pm 39}$	$11.5_{\pm 3.9}$	$1_{\pm 2}$	$46_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p25})	$48_{\pm 43}$	$28.5_{\pm 20.9}$	$51_{\pm 65}$	$58_{\pm 9}$
Fast-TD-MPC (*RG*, τ_{p50})	$43_{\pm 43}$	$25.6_{\pm 18.9}$	$40_{\pm 55}$	$51_{\pm 8}$
Fast-TD-MPC (*RG*, τ_{p75})	$39_{\pm 51}$	$20.9_{\pm 10.8}$	$25_{\pm 29}$	$46_{\pm 5}$

- **Cold start:** Zero the buffer and discard all stale plans. The MPPI planner relies entirely on the policy prior π_θ, which provides $N_\pi = 24$ of $N = 512$ candidate trajectories rolled out from the current latent through the world model.
- **K-shifted warm start:** Shift the buffer left by K positions, salvaging the first $H-K$ future plans when $K < H-1$. If $K \geq H-1$, no valid plan survives and the strategy degenerates to cold start.

We evaluate both strategies across all 103 tasks from the 4 domains and 8 τ configurations (*theoretical*: $p50$, $p75$, $p90$, $p95$, $p99$; *reward-gated*: $p25$, $p50$, $p75$), yielding 824 paired comparisons on the clean benchmark and 816 under disturbance. A paired t-test finds no significant difference ($p > 0.05$) in return or S2 invocation rate between the two strategies on either benchmark, confirming that the policy prior provides sufficient fresh candidates on S1→S2 transitions. We adopt cold start for its simplicity. Tables 26 and 27 report the full results.

J OOD Distance Metric Ablation

We compare Mahalanobis distance against Tanimoto distance (Rogers & Tanimoto, 1960) as the OOD scoring function. Mahalanobis distance captures the full covariance structure of the in-distribution latents:

$$g_{\text{maha}}(z_t) = (z_t - \bar{z})^\top \Sigma_{\text{reg}}^{-1} (z_t - \bar{z}), \tag{5}$$

where $\bar{z} \in \mathbb{R}^{d_z}$ is the mean of the ID latents and $\Sigma_{\text{reg}} = \Sigma + \lambda \mathbf{I}$ is the ℓ_2-regularized covariance (see Section 3), yielding an effective metric that accounts for inter-dimension correlations. Tanimoto distance measures angular and magnitude deviation from the mean without modeling correlations:

$$g_{\text{tan}}(z_t) = 1 - \frac{z_t^\top \bar{z}}{\|z_t\|^2 + \|\bar{z}\|^2 - z_t^\top \bar{z}}, \tag{6}$$

where $z_t, \bar{z} \in \mathbb{R}^{d_z}$ and the denominator $\|z_t\|^2 + \|\bar{z}\|^2 - z_t^\top \bar{z}$ combines Euclidean and inner-product terms into a single similarity ratio. Tanimoto is computationally cheaper ($O(d_z)$ vs. $O(d_z^2)$ per step) and requires no covariance inversion during fitting. Both detectors are fitted on the same System 1 rollout trajectories and evaluated under identical τ percentiles.

Table 14: **DMControl per-task results.** 39 tasks, 3 seeds each. Mean and 95% CI per task. Return and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Return	**Lat.** (ms)	**Return**	**Lat.** (ms)	**Return**	**Lat.** (ms)	**Return**	**Lat.** (ms)
Acrobot Swingup	$642_{\pm 25}$	$26.2_{\pm .8}$	$523_{\pm 163}$	$1.2_{\pm .0}$	$638_{\pm 87}$	$9.8_{\pm 3.1}$	$582_{\pm 155}$	$2.6_{\pm .9}$
Cartpole Balance Sparse	1000	$26.0_{\pm 1.7}$	1000	$1.3_{\pm .0}$	1000	$14.8_{\pm 23.9}$	1000	$4.8_{\pm 5.0}$
Cartpole Balance	$999_{\pm 1}$	$25.8_{\pm 1.4}$	$1000_{\pm 1}$	$1.4_{\pm .0}$	$999_{\pm 1}$	$21.1_{\pm 11.7}$	$999_{\pm 1}$	$3.4_{\pm 1.2}$
Cartpole Swingup Sparse	$297_{\pm 1189}$	$25.6_{\pm 1.3}$	$284_{\pm 1216}$	$1.2_{\pm .0}$	$284_{\pm 1216}$	$17.3_{\pm 13.9}$	$303_{\pm 1177}$	$3.8_{\pm 1.1}$
Cartpole Swingup	$661_{\pm 954}$	$26.2_{\pm 1.1}$	$665_{\pm 938}$	$1.4_{\pm .1}$	$654_{\pm 987}$	$14.4_{\pm 4.1}$	$666_{\pm 934}$	$3.5_{\pm 1.5}$
Cheetah Jump	$833_{\pm 12}$	$26.4_{\pm 1.1}$	$779_{\pm 122}$	$1.3_{\pm .2}$	$833_{\pm 6}$	$12.4_{\pm 1.4}$	$828_{\pm 9}$	$3.4_{\pm 2.2}$
Cheetah Run Back	$854_{\pm 26}$	$26.0_{\pm 2.0}$	$585_{\pm 576}$	$1.2_{\pm .1}$	$854_{\pm 26}$	$13.8_{\pm 14.8}$	$791_{\pm 254}$	$3.4_{\pm 3.5}$
Cheetah Run Backwards	$785_{\pm 230}$	$25.8_{\pm .7}$	$696_{\pm 197}$	$1.1_{\pm .1}$	$787_{\pm 238}$	$22.0_{\pm 17.9}$	$793_{\pm 239}$	$2.9_{\pm 2.0}$
Cheetah Run Front	$762_{\pm 150}$	$25.5_{\pm .4}$	$582_{\pm 346}$	$1.2_{\pm .0}$	$763_{\pm 152}$	$16.0_{\pm 7.9}$	$764_{\pm 147}$	$3.1_{\pm .4}$
Cheetah Run	$887_{\pm 139}$	$25.2_{\pm .5}$	$450_{\pm 605}$	$1.1_{\pm .0}$	$886_{\pm 97}$	$16.3_{\pm 19.1}$	$798_{\pm 240}$	$3.8_{\pm 1.6}$
Cup Catch	$983_{\pm 1}$	$25.7_{\pm 1.0}$	$975_{\pm 20}$	$1.3_{\pm .0}$	$980_{\pm 5}$	$25.1_{\pm .8}$	$981_{\pm 3}$	$13.6_{\pm 17.8}$
Cup Spin	$849_{\pm 1}$	$25.6_{\pm .4}$	$849_{\pm 1}$	$1.4_{\pm .0}$	$849_{\pm 1}$	$17.9_{\pm 2.4}$	$849_{\pm 1}$	$5.4_{\pm 4.6}$
Dog Run	$823_{\pm 128}$	$34.6_{\pm 1.6}$	$171_{\pm 444}$	$3.7_{\pm .9}$	$817_{\pm 101}$	$26.7_{\pm 35.5}$	$603_{\pm 282}$	$11.4_{\pm 20.8}$
Dog Stand	$966_{\pm 76}$	$34.7_{\pm 1.3}$	$484_{\pm 565}$	$3.8_{\pm .7}$	$934_{\pm 119}$	$30.3_{\pm 22.0}$	$821_{\pm 543}$	$8.8_{\pm 9.2}$
Dog Trot	$943_{\pm 51}$	$34.9_{\pm .4}$	$363_{\pm 447}$	$3.7_{\pm .7}$	$947_{\pm 20}$	$34.7_{\pm .7}$	$926_{\pm 164}$	$7.9_{\pm 3.1}$
Dog Walk	$898_{\pm 201}$	$35.8_{\pm .4}$	$775_{\pm 168}$	$3.5_{\pm .5}$	$893_{\pm 224}$	$33.0_{\pm 13.0}$	$826_{\pm 131}$	$6.2_{\pm 1.9}$
Finger Spin	$986_{\pm 2}$	$25.7_{\pm 1.2}$	$982_{\pm 9}$	$1.1_{\pm .0}$	$988_{\pm 5}$	$23.9_{\pm 7.9}$	$989_{\pm 4}$	$17.1_{\pm 32.5}$
Finger Turn Easy	$930_{\pm 208}$	$26.0_{\pm .6}$	$594_{\pm 710}$	$1.3_{\pm .1}$	$949_{\pm 157}$	$14.4_{\pm 9.7}$	$849_{\pm 136}$	$2.5_{\pm 2.6}$
Finger Turn Hard	$909_{\pm 259}$	$25.5_{\pm .6}$	$524_{\pm 510}$	$1.3_{\pm .1}$	$942_{\pm 142}$	$12.2_{\pm 9.7}$	$887_{\pm 185}$	$2.1_{\pm .7}$
Fish Swim	$700_{\pm 90}$	$25.7_{\pm .7}$	$520_{\pm 298}$	$1.3_{\pm .0}$	$767_{\pm 183}$	$14.5_{\pm 5.7}$	$545_{\pm 318}$	$3.7_{\pm 1.1}$
Hopper Hop Backwards	$825_{\pm 136}$	$25.8_{\pm 1.7}$	$789_{\pm 187}$	$1.2_{\pm .0}$	$887_{\pm 98}$	$19.9_{\pm 8.6}$	$906_{\pm 131}$	$2.4_{\pm .7}$
Hopper Hop	$490_{\pm 169}$	$25.6_{\pm .7}$	$431_{\pm 381}$	$1.2_{\pm .1}$	$532_{\pm 192}$	$9.9_{\pm 10.0}$	$505_{\pm 246}$	$2.8_{\pm 1.2}$
Hopper Stand	$863_{\pm 8}$	$26.0_{\pm 1.6}$	$810_{\pm 257}$	$1.3_{\pm .0}$	$915_{\pm 128}$	$22.4_{\pm 14.9}$	$921_{\pm 130}$	$2.9_{\pm 2.0}$
Humanoid Run	$453_{\pm 974}$	$33.4_{\pm .4}$	$13_{\pm 42}$	$1.9_{\pm .1}$	$451_{\pm 971}$	$33.7_{\pm 1.0}$	$451_{\pm 971}$	$33.9_{\pm 1.3}$
Humanoid Stand	$604_{\pm 1290}$	$33.9_{\pm .8}$	$37_{\pm 71}$	$2.1_{\pm .2}$	$609_{\pm 1299}$	$34.1_{\pm 1.7}$	$535_{\pm 1147}$	$15.6_{\pm 31.0}$
Humanoid Walk	$606_{\pm 1302}$	$33.7_{\pm .2}$	$40_{\pm 158}$	$2.0_{\pm .2}$	$607_{\pm 1303}$	$33.3_{\pm .9}$	$583_{\pm 1255}$	$10.4_{\pm 14.8}$
Pendulum Spin	$856_{\pm 4}$	$25.3_{\pm .9}$	$857_{\pm 60}$	$1.2_{\pm .1}$	$857_{\pm 59}$	$12.6_{\pm 2.5}$	$857_{\pm 59}$	$3.5_{\pm 2.1}$
Pendulum Swingup	$570_{\pm 1227}$	$25.3_{\pm .7}$	$505_{\pm 1096}$	$1.1_{\pm .0}$	$559_{\pm 1203}$	$15.5_{\pm 15.9}$	$502_{\pm 1091}$	$3.9_{\pm .6}$
Quadruped Run	$950_{\pm 16}$	$26.6_{\pm 1.8}$	$950_{\pm 7}$	$1.6_{\pm .1}$	$957_{\pm 8}$	$25.7_{\pm 1.5}$	$959_{\pm 8}$	$3.6_{\pm .3}$
Quadruped Walk	$967_{\pm 23}$	$25.9_{\pm 1.4}$	$971_{\pm 18}$	$1.6_{\pm .0}$	$978_{\pm 15}$	$26.3_{\pm 1.2}$	$979_{\pm 15}$	$18.1_{\pm 12.2}$
Reacher Easy	$981_{\pm 7}$	$25.3_{\pm .2}$	$933_{\pm 128}$	$1.2_{\pm .0}$	$982_{\pm 7}$	$11.7_{\pm 10.1}$	$982_{\pm 5}$	$5.8_{\pm 9.0}$
Reacher Hard	$978_{\pm 0}$	$25.7_{\pm 1.8}$	$936_{\pm 132}$	$1.2_{\pm .0}$	$976_{\pm 13}$	$15.3_{\pm 1.3}$	$980_{\pm 5}$	$1.8_{\pm .1}$
Reacher Three Easy	$982_{\pm 1}$	$25.4_{\pm .3}$	$969_{\pm 26}$	$1.2_{\pm .0}$	$984_{\pm 1}$	$13.5_{\pm 6.3}$	$984_{\pm 2}$	$2.8_{\pm 3.3}$
Reacher Three Hard	$652_{\pm 138}$	$25.3_{\pm .4}$	$684_{\pm 238}$	$1.2_{\pm .1}$	$780_{\pm 482}$	$13.1_{\pm 2.2}$	$750_{\pm 500}$	$3.4_{\pm 6.8}$
Walker Run Backwards	$941_{\pm 18}$	$26.3_{\pm .6}$	$100_{\pm 181}$	$1.5_{\pm .0}$	$939_{\pm 37}$	$25.1_{\pm 2.3}$	$637_{\pm 891}$	$16.6_{\pm 27.0}$
Walker Run	$886_{\pm 2}$	$26.0_{\pm 1.8}$	$515_{\pm 118}$	$1.4_{\pm .1}$	$884_{\pm 11}$	$7.7_{\pm 7.5}$	$646_{\pm 294}$	$4.1_{\pm 1.5}$
Walker Stand	$989_{\pm 6}$	$26.2_{\pm 1.9}$	$825_{\pm 346}$	$1.4_{\pm .0}$	$993_{\pm 9}$	$9.5_{\pm 14.6}$	$993_{\pm 10}$	$2.5_{\pm .5}$
Walker Walk Backwards	$682_{\pm 1288}$	$25.9_{\pm 1.2}$	$511_{\pm 980}$	$1.5_{\pm .0}$	$689_{\pm 1274}$	$16.8_{\pm 9.8}$	$680_{\pm 1303}$	$2.9_{\pm 2.1}$
Walker Walk	$982_{\pm 0}$	$25.6_{\pm .9}$	$618_{\pm 467}$	$1.5_{\pm .0}$	$983_{\pm 9}$	$19.5_{\pm 4.8}$	$956_{\pm 122}$	$2.5_{\pm .4}$

Tables 28 and 29 report the full results. Under nominal conditions, a paired t-test across all task-config comparisons finds no significant difference in return (t=0.85, p=0.40), although Mahalanobis achieves substantially higher success on Meta-World (85% vs. 70% at τ_{p50}). In the *theoretical* mode, Tanimoto yields equal or lower latency at every τ percentile, reflecting fewer S2 invocations; in the *reward-gated* mode, it is often slower. Under perturbation, Mahalanobis yields significantly higher return (t=2.33, p=0.02) by invoking S2 more often. We therefore adopt Mahalanobis for its stronger robustness.

K Implementation Details

Single configuration. All Fast-TD-MPC components (architecture, training, and OOD detection) use the same hyperparameters across all 103 tasks. No per-task tuning is performed; τ is set automatically for each task-seed from held-out OOD scores using the same rule for all tasks (the median for *theoretical* mode; the midpoint of the ID and OOD medians for *reward-gated* mode; see Section 3).

K.1 System 1: Amortized Policy Architecture

Model Architecture. All components of TD-MPC2 are parameterized by MLPs with LayerNorm and Mish activations. The encoder $\mathcal{E}_\theta$ and dynamics model $\mathcal{D}_\theta$ additionally use Simplicial Normal-

Table 15: **Meta-World per-task results.** 50 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
Assembly	100	$26.4_{\pm 1.1}$	$67_{\pm 143}$	$2.2_{\pm .2}$	$90_{\pm 25}$	$27.1_{\pm 1.6}$	$87_{\pm 57}$	$19.7_{\pm 24.9}$
Basketball	$97_{\pm 14}$	$26.2_{\pm .9}$	100	$2.0_{\pm .1}$	$90_{\pm 43}$	$26.7_{\pm .1}$	$83_{\pm 72}$	$20.6_{\pm 12.0}$
Bin Picking	100	$26.8_{\pm 1.4}$	100	$2.1_{\pm .2}$	$97_{\pm 14}$	$27.1_{\pm 2.3}$	100	$14.5_{\pm 15.7}$
Box Close	100	$27.6_{\pm .5}$	100	$3.4_{\pm .6}$	100	$27.8_{\pm 1.2}$	100	$9.7_{\pm 7.4}$
Button Press Topdown Wall	100	$26.9_{\pm .7}$	100	$2.7_{\pm .2}$	100	$17.0_{\pm 9.5}$	100	$5.9_{\pm 2.0}$
Button Press Topdown	100	$26.9_{\pm 2.0}$	100	$2.7_{\pm .3}$	100	$15.6_{\pm 6.1}$	100	$6.3_{\pm 1.2}$
Button Press Wall	$93_{\pm 29}$	$27.4_{\pm 1.3}$	100	$3.0_{\pm .3}$	100	$24.7_{\pm 5.5}$	$93_{\pm 29}$	$14.1_{\pm 16.9}$
Button Press	$83_{\pm 38}$	$26.9_{\pm 1.5}$	100	$2.9_{\pm .4}$	$90_{\pm 25}$	$23.2_{\pm 15.4}$	$77_{\pm 57}$	$15.7_{\pm 18.4}$
Coffee Button	100	$26.0_{\pm .4}$	100	$2.1_{\pm .4}$	100	$23.7_{\pm 9.8}$	100	$7.5_{\pm 6.2}$
Coffee Pull	$40_{\pm 50}$	$28.1_{\pm 2.5}$	$67_{\pm 143}$	$3.7_{\pm 1.0}$	$33_{\pm 72}$	$22.2_{\pm 23.9}$	$50_{\pm 108}$	$14.3_{\pm 21.0}$
Coffee Push	$13_{\pm 38}$	$28.0_{\pm 1.7}$	$33_{\pm 143}$	$3.5_{\pm .7}$	$33_{\pm 103}$	$28.1_{\pm 1.6}$	$30_{\pm 129}$	$19.4_{\pm 28.1}$
Dial Turn	$70_{\pm 66}$	$26.3_{\pm .8}$	100	$1.8_{\pm .1}$	$83_{\pm 52}$	$26.5_{\pm 1.5}$	$83_{\pm 72}$	$16.1_{\pm 20.8}$
Disassemble	$67_{\pm 143}$	$26.3_{\pm .2}$	$67_{\pm 143}$	$2.1_{\pm .6}$	$67_{\pm 143}$	$27.3_{\pm .8}$	$67_{\pm 143}$	$8.7_{\pm 5.1}$
Door Close	100	$26.0_{\pm .3}$	100	$2.1_{\pm .3}$	100	$15.7_{\pm 1.3}$	100	$5.6_{\pm 1.8}$
Door Lock	100	$25.8_{\pm 1.1}$	100	$2.0_{\pm .1}$	100	$15.0_{\pm 6.8}$	100	$4.9_{\pm .6}$
Door Open	100	$26.8_{\pm .8}$	100	$2.7_{\pm .2}$	$93_{\pm 29}$	$22.1_{\pm 14.4}$	90	$14.0_{\pm 28.1}$
Door Unlock	100	$26.2_{\pm .4}$	100	$1.8_{\pm .1}$	100	$19.2_{\pm 8.7}$	100	$5.7_{\pm 1.5}$
Drawer Close	100	$25.9_{\pm .7}$	100	$1.8_{\pm .2}$	100	$25.5_{\pm 4.0}$	100	$12.7_{\pm 18.4}$
Drawer Open	100	$26.5_{\pm 2.3}$	100	$1.7_{\pm .0}$	100	$26.5_{\pm 1.7}$	100	$5.0_{\pm .6}$
Faucet Close	100	$25.6_{\pm .5}$	100	$1.7_{\pm .0}$	100	$26.2_{\pm .4}$	100	$8.6_{\pm 17.3}$
Faucet Open	100	$25.7_{\pm .3}$	100	$1.7_{\pm .1}$	100	$24.0_{\pm 3.3}$	100	$9.3_{\pm 18.0}$
Hammer	100	$26.5_{\pm 1.0}$	100	$2.3_{\pm .0}$	100	$26.1_{\pm 3.7}$	100	$14.9_{\pm 13.0}$
Hand Insert	100	$27.5_{\pm 1.1}$	100	$3.1_{\pm .1}$	100	$26.7_{\pm 3.5}$	100	$16.4_{\pm 21.2}$
Handle Press Side	100	$26.2_{\pm 1.5}$	100	$1.7_{\pm .1}$	100	$26.8_{\pm .7}$	100	$14.2_{\pm 19.4}$
Handle Press	100	$26.5_{\pm 2.4}$	100	$1.8_{\pm .2}$	100	$22.0_{\pm 19.6}$	$93_{\pm 29}$	$10.0_{\pm 11.1}$
Handle Pull Side	0	$27.2_{\pm 2.7}$	0	$2.2_{\pm .1}$	0	$20.4_{\pm 11.6}$	0	$8.2_{\pm 13.2}$
Handle Pull	0	$26.9_{\pm .8}$	0	$2.2_{\pm .2}$	$3_{\pm 14}$	$24.4_{\pm 5.2}$	0	$18.4_{\pm 9.5}$
Lever Pull	$7_{\pm 14}$	$26.6_{\pm .8}$	$33_{\pm 143}$	$2.2_{\pm .3}$	$10_{\pm 25}$	$18.3_{\pm 13.9}$	$7_{\pm 14}$	$9.3_{\pm 3.0}$
Peg Insert Side	100	$26.7_{\pm 1.4}$	100	$2.5_{\pm .6}$	100	$27.4_{\pm 1.3}$	100	$13.0_{\pm 27.0}$
Peg Unplug Side	100	$26.3_{\pm 1.3}$	$67_{\pm 143}$	$2.1_{\pm .0}$	$87_{\pm 57}$	$27.2_{\pm .9}$	$97_{\pm 14}$	$21.9_{\pm 8.8}$
Pick Out Of Hole	100	$26.4_{\pm .2}$	100	$2.3_{\pm .3}$	100	$27.5_{\pm .9}$	100	$8.3_{\pm 6.2}$
Pick Place Wall	100	$27.8_{\pm 1.1}$	100	$3.5_{\pm .4}$	100	$27.7_{\pm .7}$	100	$12.7_{\pm 6.3}$
Pick Place	100	$27.8_{\pm 1.3}$	100	$3.3_{\pm .0}$	100	$28.2_{\pm 2.2}$	100	$8.2_{\pm 3.7}$
Plate Slide Back Side	100	$26.7_{\pm 2.1}$	100	$2.1_{\pm .3}$	100	$27.1_{\pm 2.2}$	100	$6.6_{\pm 2.5}$
Plate Slide Back	100	$26.0_{\pm .5}$	100	$2.0_{\pm .1}$	100	$26.9_{\pm .3}$	100	$6.3_{\pm 6.9}$
Plate Slide Side	$27_{\pm 38}$	$26.7_{\pm .8}$	$33_{\pm 143}$	$2.4_{\pm .4}$	$33_{\pm 143}$	$20.6_{\pm 15.4}$	$33_{\pm 143}$	$7.8_{\pm 5.1}$
Plate Slide	100	$26.5_{\pm .5}$	100	$2.2_{\pm .2}$	100	$26.1_{\pm 4.6}$	100	$7.8_{\pm 11.8}$
Push Back	$67_{\pm 143}$	$28.2_{\pm 1.8}$	$67_{\pm 143}$	$3.0_{\pm .1}$	$67_{\pm 143}$	$24.9_{\pm 8.0}$	$67_{\pm 143}$	$7.4_{\pm 4.2}$
Push Wall	$93_{\pm 29}$	$27.8_{\pm .2}$	100	$3.5_{\pm .2}$	100	$28.8_{\pm 1.1}$	100	$14.6_{\pm 18.0}$
Push	100	$28.3_{\pm .9}$	100	$3.4_{\pm .2}$	100	$28.5_{\pm .7}$	100	$16.1_{\pm 23.4}$
Reach Wall	100	$26.2_{\pm .8}$	100	$2.1_{\pm .2}$	100	$27.7_{\pm 1.1}$	100	$4.5_{\pm .3}$
Reach	$90_{\pm 43}$	$26.5_{\pm 1.6}$	100	$2.1_{\pm .1}$	$83_{\pm 72}$	$26.4_{\pm 1.0}$	100	$4.6_{\pm .3}$
Shelf Place	100	$28.4_{\pm 2.1}$	100	$3.3_{\pm .5}$	100	$26.8_{\pm 6.3}$	100	$13.7_{\pm 12.4}$
Soccer	100	$27.8_{\pm 1.0}$	100	$3.2_{\pm .1}$	$83_{\pm 72}$	$28.9_{\pm 2.3}$	100	$18.1_{\pm 10.2}$
Stick Pull	$7_{\pm 29}$	$28.0_{\pm 1.0}$	0	$3.4_{\pm .2}$	$33_{\pm 143}$	$28.1_{\pm 2.1}$	$33_{\pm 143}$	$27.4_{\pm 2.0}$
Stick Push	$90_{\pm 25}$	$27.9_{\pm .7}$	100	$3.3_{\pm .1}$	$83_{\pm 38}$	$27.8_{\pm 2.2}$	$83_{\pm 38}$	$20.4_{\pm 10.2}$
Sweep Into	$97_{\pm 14}$	$27.4_{\pm .2}$	100	$3.0_{\pm .1}$	$97_{\pm 14}$	$28.0_{\pm 1.7}$	100	$21.7_{\pm 19.1}$
Sweep	100	$27.2_{\pm 1.3}$	100	$2.8_{\pm .1}$	100	$28.2_{\pm 2.7}$	100	$21.4_{\pm 9.2}$
Window Close	$67_{\pm 143}$	$26.0_{\pm .4}$	100	$1.9_{\pm .4}$	$93_{\pm 29}$	$17.9_{\pm 13.1}$	$73_{\pm 115}$	$9.2_{\pm 15.0}$
Window Open	100	$26.3_{\pm 1.9}$	100	$1.9_{\pm .4}$	100	$23.4_{\pm 11.8}$	100	$6.5_{\pm 4.7}$

ization (SimNorm) (Lavoie et al., 2022) on their final layer to normalize the latent representation z, biasing it towards sparsity without enforcing hard discrete constraints. Mirroring this architecture, the amortized policy π_{S1} is a 4-layer MLP: three intermediate layers of linear $\rightarrow$ LayerNorm $\rightarrow$ Mish, followed by a final SimNorm layer. The backbone takes the frozen latent $z \in \mathbb{R}^{d_z}$ ($d_z = 512$) as input, maps to a 256-dimensional hidden representation, and a deterministic linear head produces the action. We use history length $L = 1$ and chunk size $C = 1$ (single-step prediction). Following TD-MPC2, actions are squashed with $\tanh$ at the output and dropout (rate 0.1) is applied to the first layer only. We summarize the π_{S1} architecture using PyTorch-like notation:

```
backbone = SimNormMLPBackbone(
    (0): NormedLinear(512, 256, act=Mish, dropout=0.1)
```

Table 16: **MyoSuite per-task results.** 10 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
MyoHand Key Turn Hard	$30_{\pm 66}$	$36.6_{\pm 1.5}$	$20_{\pm 86}$	$4.1_{\pm .5}$	$27_{\pm 57}$	$16.6_{\pm 6.0}$	$17_{\pm 38}$	$10.3_{\pm 2.2}$
MyoHand Key Turn	$27_{\pm 76}$	$36.1_{\pm 1.6}$	$33_{\pm 143}$	$4.6_{\pm 1.3}$	$27_{\pm 76}$	$36.1_{\pm 2.3}$	$23_{\pm 63}$	$34.2_{\pm 7.0}$
MyoHand Obj Hold Hard	$50_{\pm 66}$	$36.0_{\pm 1.4}$	$70_{\pm 50}$	$4.2_{\pm .7}$	$43_{\pm 29}$	$29.7_{\pm 6.3}$	$53_{\pm 63}$	$16.5_{\pm 20.0}$
MyoHand Obj Hold	0	$35.2_{\pm .5}$	0	$3.6_{\pm .2}$	0	$34.6_{\pm .7}$	0	$31.4_{\pm 6.3}$
MyoHand Pen Twirl Hard	$47_{\pm 29}$	$35.9_{\pm .8}$	$23_{\pm 14}$	$3.3_{\pm .6}$	$50_{\pm 75}$	$32.5_{\pm 5.2}$	$27_{\pm 14}$	$12.0_{\pm 5.9}$
MyoHand Pen Twirl	$27_{\pm 57}$	$35.1_{\pm 2.3}$	$33_{\pm 143}$	$3.1_{\pm .5}$	$27_{\pm 57}$	$35.3_{\pm 1.0}$	$27_{\pm 57}$	$35.2_{\pm 3.0}$
MyoHand Pose Hard	0	$34.8_{\pm 1.2}$	0	$3.1_{\pm .4}$	0	$28.2_{\pm 2.0}$	0	$7.2_{\pm 3.9}$
MyoHand Pose	$30_{\pm 129}$	$34.6_{\pm .5}$	$33_{\pm 143}$	$3.0_{\pm .2}$	$30_{\pm 129}$	$35.1_{\pm 1.0}$	$33_{\pm 143}$	$33.5_{\pm 3.2}$
MyoHand Reach Hard	$87_{\pm 38}$	$35.4_{\pm 1.5}$	$70_{\pm 25}$	$3.4_{\pm .3}$	$90_{\pm 25}$	$31.6_{\pm 7.7}$	$87_{\pm 38}$	$10.1_{\pm 10.8}$
MyoHand Reach	100	$35.5_{\pm 1.3}$	$67_{\pm 143}$	$3.3_{\pm .1}$	100	$34.8_{\pm .6}$	100	$35.2_{\pm 2.0}$

Table 17: **ManiSkill2 per-task results.** 4 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
Lift Cube	100	$46.1_{\pm 2.1}$	$20_{\pm 50}$	$8.2_{\pm 5.7}$	$93_{\pm 29}$	$38.1_{\pm 21.6}$	$60_{\pm 25}$	$17.0_{\pm 6.5}$
Pick Cube	$87_{\pm 57}$	$46.2_{\pm 2.3}$	$53_{\pm 72}$	$10.7_{\pm 1.3}$	$73_{\pm 94}$	$25.0_{\pm 8.1}$	$50_{\pm 66}$	$16.0_{\pm 1.3}$
Pick Ycb	$27_{\pm 38}$	$49.3_{\pm 3.8}$	$3_{\pm 14}$	$14.0_{\pm 2.4}$	$27_{\pm 57}$	$41.9_{\pm 8.0}$	$13_{\pm 14}$	$19.3_{\pm 9.1}$
Stack Cube	$47_{\pm 115}$	$41.9_{\pm 2.0}$	0	$9.7_{\pm .6}$	$23_{\pm 80}$	$30.5_{\pm 11.9}$	$7_{\pm 29}$	$11.6_{\pm 2.0}$

```
    (1): NormedLinear(256, 256, act=Mish)
    (2): NormedLinear(256, 256, act=Mish)
    (3): NormedLinear(256, 256, act=SimNorm(dim=8))
)
head = DeterministicHead(
    Linear(256, action_dim)
)
```

Parameter count. The backbone contains 330,752 parameters; the head contains $256 \times d_a + d_a$ parameters. Total: $\sim$332K for $d_a = 6$ (DMControl cheetah) to $\sim$341K for $d_a = 39$ (MyoSuite). Compared to TD-MPC2's 7.9M parameters, this represents a 23$\times$ reduction.

K.2 System 1 Training

Expert demonstrations. Expert demonstrations $\mathcal{D}_{\text{exp}} = \{(z_t, a_t^*)\}$ are collected by rolling out the pre-trained TD-MPC2 planner (3 seeds per task, 500 episodes per seed) and recording the latent $z_t = \mathcal{E}_\theta(s_t)$ and the planner's executed action a_t^*. Episodes are split 8:1:1 into training, validation, and testing.

Loss. We use L1 loss: $\ell(\hat{a}, a^*) = |\hat{a} - a^*|$, where $|\hat{a} - a^*|$ is the element-wise absolute difference, summed over the action dimension d_a.

Optimizer. AdamW (Loshchilov & Hutter, 2019) with learning rate (lr) 5×10^{-5}, weight decay 10^{-4}, betas $(0.9, 0.999)$, cosine learning rate schedule with 10% warmup and minimum lr 10^{-6}.

Training. Batch size 512, gradient clipping at 1.0, maximum 200 epochs with early stopping (patience 15, minimum delta 10^{-4}). All models are trained to full convergence.

Table 18: **DMControl full results under disturbances.** 39 tasks, 3 seeds each. Mean and 95% CI across all tasks. Return, per-step latency, S2 invocation rate (ρ), and peak GPU memory. *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Return**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
S2-alone (TD-MPC2)	$616_{\pm 104}$	$27.2_{\pm 1.1}$	100	$95_{\pm 0}$
S2 policy-only	$390_{\pm 102}$	$1.9_{\pm .3}$	0	$66_{\pm 0}$
S1 alone	$405_{\pm 98}$	$1.7_{\pm .3}$	0	$68_{\pm 0}$
Round-robin (S2 50%)	$577_{\pm 102}$	$14.5_{\pm .7}$	50	$96_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$617_{\pm 105}$	$24.5_{\pm 1.9}$	$88_{\pm 5}$	$91_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p75})	$607_{\pm 103}$	$19.3_{\pm 2.9}$	$67_{\pm 10}$	$86_{\pm 3}$
Fast-TD-MPC (*all*, τ_{p90})	$586_{\pm 101}$	$13.6_{\pm 3.0}$	$45_{\pm 11}$	$81_{\pm 3}$
Fast-TD-MPC (*all*, τ_{p95})	$537_{\pm 102}$	$9.8_{\pm 2.6}$	$32_{\pm 10}$	$77_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p99})	$439_{\pm 101}$	$5.6_{\pm 2.0}$	$16_{\pm 8}$	$72_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p25})	$566_{\pm 104}$	$17.5_{\pm 2.6}$	$61_{\pm 10}$	$85_{\pm 3}$
Fast-TD-MPC (*RG*, τ_{p50})	$543_{\pm 102}$	$14.3_{\pm 2.5}$	$49_{\pm 10}$	$80_{\pm 3}$
Fast-TD-MPC (*RG*, τ_{p75})	$524_{\pm 102}$	$12.2_{\pm 2.3}$	$41_{\pm 9}$	$80_{\pm 3}$

Table 19: **Meta-World full results under disturbances.** 50 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
S2-alone (TD-MPC2)	$51_{\pm 10}$	$26.8_{\pm .2}$	100	$95_{\pm 0}$
S2 policy-only	$49_{\pm 10}$	$2.8_{\pm .2}$	0	$66_{\pm 0}$
S1 alone	$47_{\pm 10}$	$2.6_{\pm .2}$	0	$67_{\pm 0}$
Round-robin (S2 50%)	$49_{\pm 11}$	$14.7_{\pm .2}$	50	$96_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$51_{\pm 11}$	$27.2_{\pm .2}$	$99_{\pm 0}$	$97_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p75})	$51_{\pm 11}$	$25.6_{\pm .9}$	$93_{\pm 3}$	$96_{\pm 1}$
Fast-TD-MPC (*all*, τ_{p90})	$49_{\pm 10}$	$22.8_{\pm 1.6}$	$82_{\pm 6}$	$95_{\pm 1}$
Fast-TD-MPC (*all*, τ_{p95})	$50_{\pm 10}$	$21.8_{\pm 1.8}$	$78_{\pm 7}$	$93_{\pm 2}$
Fast-TD-MPC (*all*, τ_{p99})	$50_{\pm 10}$	$20.2_{\pm 2.0}$	$71_{\pm 8}$	$92_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p25})	$49_{\pm 11}$	$20.9_{\pm 1.9}$	$74_{\pm 7}$	$91_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p50})	$50_{\pm 10}$	$18.9_{\pm 2.1}$	$66_{\pm 8}$	$89_{\pm 2}$
Fast-TD-MPC (*RG*, τ_{p75})	$49_{\pm 10}$	$17.4_{\pm 2.2}$	$60_{\pm 8}$	$87_{\pm 2}$

K.3 OOD Detector

The OOD detector uses Mahalanobis distance in the frozen encoder's latent space, operating on $d_z = 512$-dimensional latent features (z_t with history length $L = 1$). The in-distribution mean $\bar{z}$ and full covariance Σ are computed from 400 System 1 rollout episodes per task-seed. We apply ℓ_2 shrinkage: $\Sigma_{\text{reg}} = \Sigma + \lambda \mathbf{I}$ with $\lambda = 0.01$. The scoring function is $g(z_t) = (z_t - \bar{z})^\top \Sigma_{\text{reg}}^{-1} (z_t - \bar{z})$. The threshold τ is computed on a separate held-out set of 100 rollout episodes.

We consider two labeling modes for constructing the ID set:

- **Theoretical** (*all*): all visited latents from System 1 rollouts are treated as in-distribution. The threshold τ defaults to the median of $g(z_t)$ scores on the held-out set.
- **Reward-gated** (*RG*): a state is labeled in-distribution only if its total per-step reward or task completion proxy obtained from the environment meets or exceeds the TD-MPC2 expert's per-step counterpart at the same timestep. The threshold τ defaults to the midpoint between the median ID and OOD scores.

Table 20: **MyoSuite full results under disturbances.** 10 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
S2-alone (TD-MPC2)	$40_{\pm 22}$	$35.7_{\pm .5}$	100	$97_{\pm 0}$
S2 policy-only	$17_{\pm 15}$	$3.9_{\pm .4}$	0	$67_{\pm 0}$
S1 alone	$32_{\pm 15}$	$3.5_{\pm .3}$	0	$69_{\pm 0}$
Round-robin (S2 50%)	$37_{\pm 20}$	$19.4_{\pm .3}$	50	$98_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$36_{\pm 20}$	$32.2_{\pm 3.9}$	$89_{\pm 13}$	$94_{\pm 4}$
Fast-TD-MPC (*all*, τ_{p75})	$38_{\pm 22}$	$27.2_{\pm 5.9}$	$74_{\pm 19}$	$89_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p90})	$34_{\pm 22}$	$22.9_{\pm 7.8}$	$60_{\pm 24}$	$87_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p95})	$33_{\pm 20}$	$20.0_{\pm 7.8}$	$51_{\pm 25}$	$81_{\pm 5}$
Fast-TD-MPC (*all*, τ_{p99})	$30_{\pm 18}$	$13.3_{\pm 7.8}$	$30_{\pm 25}$	$79_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p25})	$36_{\pm 22}$	$25.7_{\pm 4.7}$	$70_{\pm 15}$	$92_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p50})	$33_{\pm 22}$	$21.4_{\pm 5.6}$	$56_{\pm 18}$	$89_{\pm 5}$
Fast-TD-MPC (*RG*, τ_{p75})	$35_{\pm 24}$	$17.7_{\pm 5.3}$	$44_{\pm 17}$	$82_{\pm 5}$

Table 21: **ManiSkill2 full results under disturbances.** 3 tasks, 3 seeds each. Mean and 95% CI across all tasks. Success rate, per-step latency, S2 invocation rate (ρ), and peak GPU memory. *all*: theoretical OOD mode; *RG*: reward-gated mode.

Method	**Succ.**	**Lat.** (ms)	ρ (%)	**GPU** (MB)
S2-alone (TD-MPC2)	$66_{\pm 75}$	$43.1_{\pm 4.7}$	100	$70_{\pm 0}$
S2 policy-only	$63_{\pm 66}$	$11.4_{\pm 5.7}$	0	$41_{\pm 0}$
S1 alone	$17_{\pm 52}$	$10.4_{\pm 4.2}$	0	$42_{\pm 0}$
Round-robin (S2 50%)	$58_{\pm 89}$	$27.1_{\pm 2.0}$	50	$72_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p50})	$66_{\pm 80}$	$34.6_{\pm 18.0}$	$74_{\pm 52}$	$63_{\pm 10}$
Fast-TD-MPC (*all*, τ_{p75})	$49_{\pm 108}$	$25.5_{\pm 22.3}$	$44_{\pm 65}$	$56_{\pm 10}$
Fast-TD-MPC (*all*, τ_{p90})	$46_{\pm 86}$	$16.9_{\pm 10.5}$	$18_{\pm 32}$	$47_{\pm 6}$
Fast-TD-MPC (*all*, τ_{p95})	$32_{\pm 62}$	$13.7_{\pm 5.2}$	$10_{\pm 18}$	$43_{\pm 0}$
Fast-TD-MPC (*all*, τ_{p99})	$19_{\pm 46}$	$11.2_{\pm 3.1}$	$2_{\pm 4}$	$43_{\pm 0}$
Fast-TD-MPC (*RG*, τ_{p25})	$62_{\pm 55}$	$25.9_{\pm 25.6}$	$46_{\pm 91}$	$53_{\pm 10}$
Fast-TD-MPC (*RG*, τ_{p50})	$51_{\pm 61}$	$23.1_{\pm 23.6}$	$37_{\pm 84}$	$53_{\pm 10}$
Fast-TD-MPC (*RG*, τ_{p75})	$43_{\pm 60}$	$18.8_{\pm 14.7}$	$23_{\pm 54}$	$50_{\pm 8}$

K.4 System 2

System 2. The MPPI planner (System 2) is used as-is from TD-MPC2, with default hyperparameters: planning horizon $H = 3$, 6 refinement iterations, $N = 512$ samples, $N_\pi = 24$ policy-prior trajectories, top-$k = 64$ elite trajectories, min/std 0.05/2.0, temperature 0.5.

L Extended Related Work

Data-Driven MPC and the TD-MPC Family. Data-driven model predictive control combines learned world models with online trajectory optimization for planning or data generation. TD-MPC2 (Hansen et al., 2024) combines a task-oriented latent dynamics model with MPPI planning and TD-learning, achieving state-of-the-art performance across 104 continuous control tasks with a single set of hyperparameters, outperforming strong model-free baselines such as SAC (Haarnoja et al., 2018) and model-based methods such as DreamerV3 (Hafner et al., 2025). Its growing ecosystem addresses training efficiency (Evers et al., 2026), humanoid locomotion (Nguyen et al., 2025; 2026b), hierarchical planning (Chitnis et al., 2024), policy constraints (Lin et al., 2025a), discrete latent representations (Scannell et al., 2025), and world model reliability (Hansen & Wang, 2026).

Table 22: **DMControl per-task results under disturbances.** 39 tasks, 3 seeds each. Mean and 95% CI per task. Return and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Return	**Lat.** (ms)	**Return**	**Lat.** (ms)	**Return**	**Lat.** (ms)	**Return**	**Lat.** (ms)
Acrobot Swingup	$443_{\pm 44}$	$25.4_{\pm 1.4}$	$284_{\pm 171}$	$1.3_{\pm .0}$	$386_{\pm 31}$	$20.6_{\pm 13.7}$	$368_{\pm 207}$	$3.8_{\pm 1.5}$
Cartpole Balance Sparse	$122_{\pm 132}$	$25.6_{\pm .4}$	$118_{\pm 128}$	$1.3_{\pm .1}$	$124_{\pm 135}$	$25.6_{\pm .5}$	$123_{\pm 132}$	$25.5_{\pm 1.2}$
Cartpole Balance	$324_{\pm 245}$	$25.8_{\pm 1.3}$	$323_{\pm 117}$	$1.5_{\pm .0}$	$322_{\pm 237}$	$25.6_{\pm .9}$	$325_{\pm 242}$	$26.1_{\pm 3.7}$
Cartpole Swingup Sparse	$9_{\pm 19}$	$25.3_{\pm .5}$	$4_{\pm 15}$	$1.3_{\pm .1}$	$2_{\pm 4}$	$21.4_{\pm 11.2}$	$3_{\pm 3}$	$11.9_{\pm 28.4}$
Cartpole Swingup	$244_{\pm 151}$	$25.8_{\pm .4}$	$213_{\pm 105}$	$1.5_{\pm .0}$	$241_{\pm 168}$	$24.5_{\pm 8.0}$	$246_{\pm 155}$	$18.6_{\pm 28.0}$
Cheetah Jump	$796_{\pm 22}$	$25.8_{\pm .9}$	$660_{\pm 250}$	$1.3_{\pm .0}$	$786_{\pm 20}$	$14.8_{\pm 2.4}$	$787_{\pm 61}$	$5.1_{\pm .7}$
Cheetah Run Back	$742_{\pm 70}$	$25.5_{\pm 1.6}$	$377_{\pm 475}$	$1.2_{\pm .0}$	$748_{\pm 81}$	$21.6_{\pm 6.6}$	$740_{\pm 253}$	$7.4_{\pm 4.7}$
Cheetah Run Backwards	$726_{\pm 181}$	$25.5_{\pm 1.1}$	$598_{\pm 450}$	$1.2_{\pm .0}$	$693_{\pm 163}$	$23.9_{\pm 7.3}$	$690_{\pm 97}$	$6.6_{\pm 4.3}$
Cheetah Run Front	$674_{\pm 105}$	$25.5_{\pm 1.0}$	$357_{\pm 228}$	$1.2_{\pm .1}$	$642_{\pm 115}$	$20.7_{\pm 4.1}$	$658_{\pm 139}$	$5.9_{\pm 1.0}$
Cheetah Run	$843_{\pm 90}$	$25.6_{\pm 1.8}$	$483_{\pm 594}$	$1.2_{\pm .1}$	$853_{\pm 124}$	$17.7_{\pm 17.5}$	$785_{\pm 296}$	$3.8_{\pm 1.7}$
Cup Catch	$981_{\pm 2}$	$26.0_{\pm 2.2}$	$878_{\pm 249}$	$1.3_{\pm .0}$	$969_{\pm 43}$	$25.3_{\pm .4}$	$972_{\pm 14}$	$24.8_{\pm 2.4}$
Cup Spin	$815_{\pm 69}$	$25.7_{\pm .9}$	$825_{\pm 54}$	$1.5_{\pm .1}$	$822_{\pm 44}$	$26.3_{\pm 1.6}$	$822_{\pm 43}$	$25.9_{\pm .6}$
Dog Run	$587_{\pm 449}$	$34.6_{\pm .8}$	$92_{\pm 242}$	$3.9_{\pm .7}$	$580_{\pm 435}$	$35.2_{\pm 2.2}$	$568_{\pm 353}$	$29.0_{\pm 12.6}$
Dog Stand	$948_{\pm 128}$	$35.5_{\pm .8}$	$415_{\pm 742}$	$4.1_{\pm .9}$	$941_{\pm 129}$	$35.4_{\pm .9}$	$866_{\pm 442}$	$17.8_{\pm 26.3}$
Dog Trot	$873_{\pm 103}$	$35.5_{\pm 1.6}$	$123_{\pm 154}$	$4.0_{\pm .4}$	$897_{\pm 56}$	$35.3_{\pm 1.0}$	$880_{\pm 143}$	$28.9_{\pm 20.6}$
Dog Walk	$861_{\pm 303}$	$35.2_{\pm 1.9}$	$665_{\pm 66}$	$3.7_{\pm .2}$	$879_{\pm 295}$	$35.9_{\pm 1.9}$	$775_{\pm 162}$	$14.4_{\pm 18.0}$
Finger Spin	$901_{\pm 177}$	$25.5_{\pm .4}$	$826_{\pm 313}$	$1.2_{\pm .0}$	$904_{\pm 170}$	$25.7_{\pm 1.9}$	$905_{\pm 175}$	$21.9_{\pm 16.0}$
Finger Turn Easy	$831_{\pm 47}$	$25.4_{\pm 1.2}$	$520_{\pm 336}$	$1.3_{\pm .0}$	$856_{\pm 209}$	$20.0_{\pm 8.9}$	$727_{\pm 81}$	$3.9_{\pm 1.6}$
Finger Turn Hard	$910_{\pm 117}$	$25.6_{\pm 1.4}$	$537_{\pm 287}$	$1.3_{\pm .0}$	$867_{\pm 85}$	$17.1_{\pm 5.8}$	$840_{\pm 215}$	$5.0_{\pm 5.0}$
Fish Swim	$1_{\pm 0}$	$25.1_{\pm .8}$	$1_{\pm 1}$	$1.4_{\pm .0}$	$1_{\pm 1}$	$26.2_{\pm .9}$	$1_{\pm 1}$	$9.9_{\pm 14.6}$
Hopper Hop Backwards	$761_{\pm 104}$	$25.4_{\pm .2}$	$563_{\pm 407}$	$1.3_{\pm .0}$	$816_{\pm 136}$	$24.6_{\pm 3.0}$	$784_{\pm 130}$	$6.3_{\pm .7}$
Hopper Hop	$431_{\pm 132}$	$25.6_{\pm .6}$	$350_{\pm 407}$	$1.3_{\pm .1}$	$468_{\pm 211}$	$18.1_{\pm 5.2}$	$392_{\pm 372}$	$4.8_{\pm 1.8}$
Hopper Stand	$863_{\pm 5}$	$26.1_{\pm 1.2}$	$818_{\pm 240}$	$1.4_{\pm .0}$	$899_{\pm 145}$	$25.6_{\pm 1.5}$	$924_{\pm 132}$	$2.8_{\pm 1.5}$
Humanoid Run	$388_{\pm 834}$	$33.4_{\pm 1.6}$	$7_{\pm 19}$	$2.1_{\pm .1}$	$390_{\pm 837}$	$33.6_{\pm 1.0}$	$390_{\pm 838}$	$33.3_{\pm 1.1}$
Humanoid Stand	$584_{\pm 1246}$	$34.4_{\pm 2.2}$	$37_{\pm 82}$	$2.0_{\pm .0}$	$591_{\pm 1259}$	$33.7_{\pm 2.3}$	$447_{\pm 959}$	$14.5_{\pm 29.5}$
Humanoid Walk	$578_{\pm 1234}$	$33.0_{\pm .5}$	$23_{\pm 86}$	$2.0_{\pm .1}$	$568_{\pm 1219}$	$33.6_{\pm 1.6}$	$551_{\pm 1186}$	$12.1_{\pm 14.7}$
Pendulum Spin	$853_{\pm 23}$	$25.9_{\pm 1.0}$	$861_{\pm 88}$	$1.2_{\pm .0}$	$845_{\pm 89}$	$17.4_{\pm 2.0}$	$849_{\pm 87}$	$3.6_{\pm 2.0}$
Pendulum Swingup	$394_{\pm 856}$	$25.2_{\pm 1.7}$	$441_{\pm 950}$	$1.2_{\pm .0}$	$416_{\pm 896}$	$18.7_{\pm 14.1}$	$417_{\pm 897}$	$7.2_{\pm 6.4}$
Quadruped Run	$948_{\pm 9}$	$26.7_{\pm 1.0}$	$944_{\pm 6}$	$1.6_{\pm .0}$	$953_{\pm 22}$	$25.1_{\pm 1.0}$	$952_{\pm 13}$	$4.5_{\pm 1.2}$
Quadruped Walk	$967_{\pm 17}$	$25.8_{\pm .6}$	$971_{\pm 29}$	$1.6_{\pm .0}$	$978_{\pm 9}$	$26.6_{\pm 2.2}$	$978_{\pm 21}$	$19.2_{\pm 14.9}$
Reacher Easy	$74_{\pm 97}$	$25.8_{\pm 1.4}$	$82_{\pm 44}$	$1.2_{\pm .0}$	$62_{\pm 35}$	$26.0_{\pm .5}$	$63_{\pm 35}$	$23.3_{\pm 9.9}$
Reacher Hard	$6_{\pm 11}$	$25.7_{\pm .9}$	$5_{\pm 5}$	$1.2_{\pm .0}$	$7_{\pm 11}$	$25.5_{\pm .2}$	$6_{\pm 11}$	$25.2_{\pm 2.1}$
Reacher Three Easy	$134_{\pm 210}$	$25.4_{\pm .7}$	$70_{\pm 51}$	$1.2_{\pm .0}$	$128_{\pm 52}$	$26.2_{\pm 1.6}$	$126_{\pm 59}$	$18.3_{\pm 15.9}$
Reacher Three Hard	$7_{\pm 20}$	$25.3_{\pm .2}$	$4_{\pm 11}$	$1.2_{\pm .0}$	$24_{\pm 82}$	$26.2_{\pm 1.7}$	$16_{\pm 51}$	$25.3_{\pm 1.5}$
Walker Run Backwards	$901_{\pm 19}$	$25.6_{\pm .5}$	$92_{\pm 154}$	$1.5_{\pm .0}$	$893_{\pm 108}$	$24.9_{\pm 3.1}$	$681_{\pm 912}$	$18.6_{\pm 30.5}$
Walker Run	$848_{\pm 34}$	$25.6_{\pm .5}$	$398_{\pm 431}$	$1.5_{\pm .0}$	$839_{\pm 43}$	$11.3_{\pm 8.2}$	$558_{\pm 421}$	$4.7_{\pm 1.4}$
Walker Stand	$991_{\pm 2}$	$26.2_{\pm 1.5}$	$830_{\pm 325}$	$1.5_{\pm .1}$	$992_{\pm 11}$	$23.4_{\pm 12.1}$	$992_{\pm 12}$	$2.8_{\pm .6}$
Walker Walk Backwards	$679_{\pm 1274}$	$25.8_{\pm 1.4}$	$420_{\pm 858}$	$1.5_{\pm .0}$	$682_{\pm 1303}$	$17.3_{\pm 8.6}$	$674_{\pm 1281}$	$3.4_{\pm 1.7}$
Walker Walk	$979_{\pm 7}$	$25.7_{\pm .6}$	$586_{\pm 328}$	$1.5_{\pm .0}$	$981_{\pm 14}$	$20.0_{\pm 2.0}$	$979_{\pm 16}$	$2.8_{\pm .2}$

Concurrent approaches include gradient-based planners that replace sampling-based methods with differentiable optimization (Spieler & Behnke, 2026). Fast-TD-MPC builds directly on the TD-MPC2 framework but, unlike these works, targets *inference-time* acceleration rather than training improvements or architectural modifications.

Planner Amortization and Speedup. A natural approach to reducing the cost of online planning is to distill the planner into a lightweight feed-forward policy. Kuzmenko & Shvai (2025) distill a 317M-parameter multi-task TD-MPC2 agent into a smaller quantized model, but MPPI still runs at every step. Wang et al. (2025) learn a policy by imitating an MPC expert via lazy reanalyze, but similarly still execute the planner at deployment, with the policy only seeding the optimization. Nguyen et al. (2026a) replace iterative planning entirely with a goal-conditioned inverse dynamics model, achieving 100-130$\times$ cost reduction, but provide no fallback when the amortized policy fails on out-of-distribution states. Gao & Xu (2026) accelerate the planner itself by replacing autoregressive latent rollouts with parallel action-prefix prediction, and Huang et al. (2026) amortize iterative planning with a generative flow-matching model; both still execute the full planning loop. Brudermüller et al. (2026) similarly amortize sampling-based MPC via flow matching but always run refinement iterations. Alternative acceleration mechanisms include GPU-parallelized MPC solvers (Bravo-Palacios et al., 2026) and gradient-based MPC in latent space (Spieler & Behnke, 2026), which are orthogonal to our approach. Fast-TD-MPC differs by training a *separately amortized*

Table 23: **Meta-World per-task results under disturbances.** 50 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
Assembly	$37_{\pm 29}$	$26.6_{\pm .3}$	$23_{\pm 63}$	$2.3_{\pm .3}$	$47_{\pm 52}$	$27.9_{\pm 3.7}$	$33_{\pm 38}$	$26.2_{\pm 1.5}$
Basketball	$13_{\pm 14}$	$26.2_{\pm 1.2}$	$20_{\pm 25}$	$2.0_{\pm .1}$	$10_{\pm 25}$	$27.4_{\pm 1.3}$	$10_{\pm 25}$	$26.9_{\pm 1.1}$
Bin Picking	$7_{\pm 29}$	$25.9_{\pm .3}$	$10_{\pm 25}$	$2.1_{\pm .1}$	$7_{\pm 29}$	$26.6_{\pm 1.1}$	$3_{\pm 14}$	$26.5_{\pm 1.0}$
Box Close	$77_{\pm 14}$	$27.4_{\pm .6}$	$53_{\pm 63}$	$2.9_{\pm .6}$	$77_{\pm 29}$	$28.5_{\pm .9}$	$77_{\pm 29}$	$26.8_{\pm 2.6}$
Button Press Topdown Wall	$97_{\pm 14}$	$27.0_{\pm .9}$	100	$2.8_{\pm .2}$	100	$26.6_{\pm 1.5}$	$93_{\pm 14}$	$15.3_{\pm 5.6}$
Button Press Topdown	100	$26.9_{\pm 1.3}$	100	$2.7_{\pm .2}$	$97_{\pm 14}$	$27.0_{\pm 1.3}$	$97_{\pm 14}$	$17.9_{\pm 7.4}$
Button Press Wall	$97_{\pm 14}$	$26.7_{\pm .6}$	$90_{\pm 25}$	$2.9_{\pm .1}$	100	$27.5_{\pm 2.3}$	$93_{\pm 14}$	$23.7_{\pm 7.2}$
Button Press	$67_{\pm 29}$	$27.3_{\pm 1.8}$	$80_{\pm 25}$	$2.9_{\pm .1}$	$77_{\pm 29}$	$27.0_{\pm .6}$	$80_{\pm 50}$	$21.5_{\pm 10.9}$
Coffee Button	$20_{\pm 50}$	$26.4_{\pm .4}$	$23_{\pm 52}$	$2.0_{\pm .1}$	$30_{\pm 43}$	$26.3_{\pm 1.3}$	$33_{\pm 52}$	$25.4_{\pm 3.5}$
Coffee Pull	$50_{\pm 25}$	$28.1_{\pm .5}$	$43_{\pm 76}$	$3.4_{\pm .3}$	$30_{\pm 50}$	$27.7_{\pm .1}$	$23_{\pm 38}$	$27.5_{\pm .8}$
Coffee Push	$7_{\pm 29}$	$27.9_{\pm 1.7}$	$10_{\pm 25}$	$3.5_{\pm .8}$	$3_{\pm 14}$	$28.4_{\pm 2.2}$	$13_{\pm 29}$	$27.4_{\pm .8}$
Dial Turn	$77_{\pm 52}$	$25.7_{\pm .0}$	$80_{\pm 0}$	$1.9_{\pm .0}$	$80_{\pm 43}$	$27.2_{\pm 1.1}$	$80_{\pm 25}$	$25.2_{\pm 3.9}$
Disassemble	$47_{\pm 103}$	$26.4_{\pm 1.0}$	$20_{\pm 50}$	$2.5_{\pm .3}$	$40_{\pm 86}$	$27.3_{\pm 1.1}$	$33_{\pm 72}$	$25.2_{\pm 3.7}$
Door Close	$90_{\pm 43}$	$26.6_{\pm 1.3}$	$90_{\pm 43}$	$2.2_{\pm .2}$	$90_{\pm 43}$	$24.7_{\pm 7.6}$	$90_{\pm 43}$	$7.8_{\pm 9.5}$
Door Lock	100	$26.4_{\pm 1.1}$	100	$2.1_{\pm .0}$	100	$27.7_{\pm .9}$	100	$15.2_{\pm 8.6}$
Door Open	$90_{\pm 25}$	$26.8_{\pm .3}$	$83_{\pm 29}$	$2.7_{\pm .2}$	$90_{\pm 25}$	$26.2_{\pm 6.0}$	$87_{\pm 14}$	$22.5_{\pm 8.2}$
Door Unlock	100	$26.0_{\pm .3}$	100	$1.9_{\pm .0}$	100	$26.7_{\pm .9}$	100	$8.9_{\pm 1.4}$
Drawer Close	$87_{\pm 29}$	$26.9_{\pm .9}$	$80_{\pm 43}$	$1.9_{\pm .3}$	$80_{\pm 25}$	$26.6_{\pm 2.1}$	$80_{\pm 50}$	$23.2_{\pm 9.6}$
Drawer Open	100	$25.6_{\pm .6}$	$87_{\pm 57}$	$2.0_{\pm .9}$	100	$26.7_{\pm 1.1}$	100	$20.8_{\pm 3.3}$
Faucet Close	100	$26.0_{\pm 1.1}$	100	$2.0_{\pm 1.0}$	100	$26.9_{\pm 1.9}$	100	$10.4_{\pm 18.8}$
Faucet Open	$93_{\pm 14}$	$25.9_{\pm 1.0}$	$87_{\pm 38}$	$1.8_{\pm .0}$	$93_{\pm 14}$	$25.5_{\pm .5}$	$90_{\pm 25}$	$7.9_{\pm 5.3}$
Hammer	$73_{\pm 14}$	$26.7_{\pm 1.5}$	$67_{\pm 38}$	$2.4_{\pm .2}$	$87_{\pm 29}$	$27.1_{\pm .9}$	$77_{\pm 38}$	$26.2_{\pm 1.0}$
Hand Insert	$3_{\pm 14}$	$27.0_{\pm .2}$	$3_{\pm 14}$	$3.2_{\pm .0}$	0	$27.4_{\pm .2}$	0	$28.1_{\pm .9}$
Handle Press Side	$93_{\pm 29}$	$25.9_{\pm 1.0}$	$97_{\pm 14}$	$1.9_{\pm .4}$	$97_{\pm 14}$	$26.3_{\pm 1.5}$	$93_{\pm 14}$	$18.0_{\pm 21.6}$
Handle Press	100	$26.5_{\pm .3}$	$87_{\pm 29}$	$2.0_{\pm .4}$	100	$26.1_{\pm .3}$	100	$17.4_{\pm 22.9}$
Handle Pull Side	$23_{\pm 63}$	$26.4_{\pm 1.6}$	0	$2.3_{\pm .1}$	$20_{\pm 43}$	$27.1_{\pm 1.0}$	$20_{\pm 50}$	$24.8_{\pm 5.4}$
Handle Pull	$3_{\pm 14}$	$26.5_{\pm .4}$	$3_{\pm 14}$	$2.3_{\pm .1}$	$7_{\pm 14}$	$27.9_{\pm 1.7}$	$7_{\pm 14}$	$26.0_{\pm 2.3}$
Lever Pull	$3_{\pm 14}$	$26.5_{\pm .9}$	$7_{\pm 29}$	$2.3_{\pm .2}$	$13_{\pm 38}$	$26.5_{\pm 2.7}$	$7_{\pm 29}$	$20.2_{\pm 8.2}$
Peg Insert Side	$27_{\pm 76}$	$26.9_{\pm .9}$	$23_{\pm 63}$	$2.4_{\pm .2}$	$30_{\pm 75}$	$27.7_{\pm 1.0}$	$20_{\pm 50}$	$26.5_{\pm .3}$
Peg Unplug Side	$67_{\pm 29}$	$26.8_{\pm .6}$	$37_{\pm 94}$	$2.4_{\pm .3}$	$73_{\pm 14}$	$26.9_{\pm 1.3}$	$53_{\pm 57}$	$26.0_{\pm 1.5}$
Pick Out Of Hole	$7_{\pm 29}$	$26.5_{\pm .7}$	$10_{\pm 25}$	$2.3_{\pm .2}$	$10_{\pm 25}$	$26.6_{\pm .9}$	$10_{\pm 25}$	$26.5_{\pm 2.7}$
Pick Place Wall	$43_{\pm 52}$	$27.5_{\pm .9}$	$30_{\pm 25}$	$3.5_{\pm .1}$	$37_{\pm 14}$	$27.8_{\pm .9}$	$37_{\pm 38}$	$27.7_{\pm .5}$
Pick Place	$57_{\pm 80}$	$28.1_{\pm 1.9}$	$43_{\pm 38}$	$3.3_{\pm .1}$	$57_{\pm 63}$	$28.0_{\pm 1.3}$	$53_{\pm 52}$	$27.6_{\pm .3}$
Plate Slide Back Side	$50_{\pm 75}$	$25.9_{\pm .2}$	$33_{\pm 80}$	$2.3_{\pm .6}$	$57_{\pm 52}$	$26.9_{\pm 1.4}$	$53_{\pm 80}$	$22.4_{\pm 17.3}$
Plate Slide Back	$47_{\pm 72}$	$26.5_{\pm 1.5}$	$20_{\pm 25}$	$2.1_{\pm .0}$	$37_{\pm 76}$	$27.6_{\pm 1.3}$	$40_{\pm 66}$	$22.2_{\pm 5.1}$
Plate Slide Side	$33_{\pm 29}$	$27.0_{\pm 1.2}$	$37_{\pm 76}$	$2.6_{\pm .2}$	$37_{\pm 63}$	$28.4_{\pm 1.7}$	$37_{\pm 80}$	$24.0_{\pm 3.0}$
Plate Slide	100	$26.8_{\pm 1.4}$	$93_{\pm 14}$	$2.4_{\pm .2}$	100	$27.5_{\pm 1.4}$	100	$19.9_{\pm 18.0}$
Push Back	$3_{\pm 14}$	$27.9_{\pm 1.4}$	$13_{\pm 38}$	$3.2_{\pm .1}$	$7_{\pm 14}$	$27.9_{\pm .9}$	$3_{\pm 14}$	$22.2_{\pm 24.9}$
Push Wall	$43_{\pm 52}$	$27.6_{\pm 1.3}$	$30_{\pm 25}$	$3.6_{\pm .2}$	$57_{\pm 14}$	$27.9_{\pm .6}$	$47_{\pm 29}$	$28.3_{\pm .8}$
Push	$50_{\pm 25}$	$27.5_{\pm .8}$	$40_{\pm 0}$	$3.4_{\pm .0}$	$50_{\pm 43}$	$28.0_{\pm .5}$	$57_{\pm 14}$	$28.4_{\pm .8}$
Reach Wall	$20_{\pm 25}$	$26.3_{\pm 1.1}$	$20_{\pm 0}$	$2.1_{\pm .0}$	$13_{\pm 14}$	$27.0_{\pm 1.3}$	$13_{\pm 14}$	$25.2_{\pm 2.0}$
Reach	$13_{\pm 38}$	$26.0_{\pm .5}$	$10_{\pm 43}$	$2.1_{\pm .0}$	$7_{\pm 14}$	$26.9_{\pm 1.1}$	$10_{\pm 43}$	$23.6_{\pm 5.4}$
Shelf Place	0	$27.7_{\pm .6}$	0	$3.2_{\pm .1}$	0	$27.7_{\pm .7}$	0	$27.8_{\pm .4}$
Soccer	$7_{\pm 29}$	$28.2_{\pm 1.9}$	$10_{\pm 25}$	$3.3_{\pm .2}$	$7_{\pm 29}$	$28.1_{\pm 2.3}$	$7_{\pm 29}$	$27.8_{\pm 2.2}$
Stick Pull	0	$27.7_{\pm .6}$	0	$3.6_{\pm .5}$	0	$28.7_{\pm 1.3}$	0	$28.4_{\pm .6}$
Stick Push	$7_{\pm 14}$	$27.4_{\pm 1.1}$	$7_{\pm 14}$	$3.4_{\pm .0}$	0	$27.7_{\pm .8}$	$3_{\pm 14}$	$27.6_{\pm .2}$
Sweep Into	$13_{\pm 29}$	$27.4_{\pm .1}$	$7_{\pm 14}$	$3.4_{\pm .2}$	$13_{\pm 14}$	$27.9_{\pm .9}$	$13_{\pm 14}$	$28.4_{\pm .6}$
Sweep	$10_{\pm 25}$	$26.9_{\pm 1.3}$	$20_{\pm 25}$	$2.9_{\pm .1}$	$10_{\pm 25}$	$27.4_{\pm 1.9}$	$7_{\pm 29}$	$27.5_{\pm .7}$
Window Close	$90_{\pm 43}$	$25.8_{\pm .9}$	100	$2.1_{\pm .4}$	$93_{\pm 29}$	$25.6_{\pm 3.4}$	$87_{\pm 38}$	$13.2_{\pm 14.7}$
Window Open	$93_{\pm 29}$	$25.8_{\pm .7}$	100	$2.0_{\pm .4}$	100	$26.7_{\pm .5}$	100	$13.3_{\pm 14.0}$

policy that can entirely replace the planner on in-distribution states, while retaining the full MPPI planner as a fallback triggered by OOD detection.

Adaptive and Intermittent Replanning. The idea of conditionally skipping replanning has a long history in classical control theory. Tabuada (2007) introduced event-triggered control, showing that a stabilizing controller need not execute at every timestep but only when a state-dependent condition is violated. Heemels et al. (2012) provided a canonical survey formalizing event-triggered and self-triggered control. Hashimoto et al. (2017) proposed *intermittent* MPC for nonlinear systems, deliberately skipping replanning when the system is tracking well. This is the closest classical analogue to Fast-TD-MPC's conditional use of the fast policy.

Table 24: **MyoSuite per-task results under disturbances.** 10 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
MyoHand Key Turn Hard	$30_{\pm 43}$	$36.9_{\pm 1.4}$	$17_{\pm 38}$	$4.1_{\pm .6}$	$30_{\pm 75}$	$15.2_{\pm 7.9}$	$7_{\pm 29}$	$10.5_{\pm 12.4}$
MyoHand Key Turn	$40_{\pm 90}$	$36.4_{\pm 1.6}$	$47_{\pm 125}$	$4.3_{\pm .7}$	$40_{\pm 90}$	$37.0_{\pm 1.5}$	$43_{\pm 103}$	$34.8_{\pm 8.3}$
MyoHand Obj Hold Hard	$70_{\pm 25}$	$36.9_{\pm 2.5}$	$50_{\pm 66}$	$4.2_{\pm .8}$	$60_{\pm 25}$	$33.8_{\pm 4.8}$	$43_{\pm 29}$	$16.7_{\pm 7.6}$
MyoHand Obj Hold	0	$36.0_{\pm 1.3}$	0	$3.5_{\pm .3}$	0	$35.4_{\pm 1.5}$	0	$32.3_{\pm 12.0}$
MyoHand Pen Twirl Hard	30	$35.4_{\pm 1.9}$	$27_{\pm 38}$	$3.2_{\pm .6}$	$30_{\pm 50}$	$32.7_{\pm 5.0}$	$33_{\pm 38}$	$13.9_{\pm 9.2}$
MyoHand Pen Twirl	$3_{\pm 14}$	$35.4_{\pm 1.2}$	$13_{\pm 57}$	$3.1_{\pm .4}$	$3_{\pm 14}$	$35.5_{\pm 2.7}$	$3_{\pm 14}$	$35.7_{\pm 3.7}$
MyoHand Pose Hard	$7_{\pm 29}$	$35.0_{\pm 2.8}$	0	$3.1_{\pm .1}$	0	$30.2_{\pm 7.5}$	0	$6.2_{\pm 4.6}$
MyoHand Pose	$27_{\pm 115}$	$34.8_{\pm .9}$	$30_{\pm 129}$	$3.0_{\pm .1}$	$27_{\pm 115}$	$35.1_{\pm 2.0}$	$30_{\pm 129}$	$33.9_{\pm 3.8}$
MyoHand Reach Hard	$97_{\pm 14}$	$34.8_{\pm .9}$	$67_{\pm 52}$	$3.5_{\pm .1}$	$73_{\pm 14}$	$31.8_{\pm 3.1}$	$87_{\pm 14}$	$8.8_{\pm 8.3}$
MyoHand Reach	$97_{\pm 14}$	$35.6_{\pm 2.2}$	$67_{\pm 143}$	$3.4_{\pm .0}$	$97_{\pm 14}$	$35.2_{\pm 1.6}$	$97_{\pm 14}$	$35.6_{\pm 1.5}$

Table 25: **ManiSkill2 per-task results under disturbances.** 3 tasks, 3 seeds each. Mean and 95% CI per task. Success rate (%) and per-step latency (ms) for 4 key methods. *all*: theoretical OOD mode.

Task	S2-alone (TD-MPC2)		S1 alone		Fast-TD-MPC (all, τ_{p50})		Fast-TD-MPC (all, τ_{p90})	
	Succ.	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)	**Succ.**	**Lat.** (ms)
Lift Cube	$93_{\pm 14}$	$44.3_{\pm 1.2}$	$10_{\pm 25}$	$8.5_{\pm 2.1}$	$93_{\pm 14}$	$42.7_{\pm 6.5}$	$73_{\pm 29}$	$21.8_{\pm 3.0}$
Pick Cube	$70_{\pm 90}$	$44.1_{\pm 1.5}$	$40_{\pm 50}$	$11.6_{\pm 2.5}$	$73_{\pm 94}$	$28.7_{\pm 9.3}$	$57_{\pm 80}$	$14.6_{\pm 2.9}$
Stack Cube	$33_{\pm 100}$	$40.9_{\pm 2.1}$	0	$11.2_{\pm .7}$	$30_{\pm 75}$	$32.4_{\pm 10.4}$	$7_{\pm 29}$	$14.4_{\pm 6.5}$

In the data-driven MPC setting, Cheng et al. (2026) propose AdaReP, a training-free wrapper that adapts replanning tolerance based on model mismatch, achieving over 80% fewer planner queries. However, AdaReP reuses the *last* cached plan rather than executing a separately learned policy, and its gate is a continuous mismatch threshold rather than a learned OOD detector. Lin et al. (2025b) apply speculative execution — a paradigm originating in LLM inference (Leviathan et al., 2023; Chen et al., 2023), directly to TD-MPC2, generating action queues from the planner's own latent rollouts and falling back to full replanning on mismatch. This approach is comparable to Fast-TD-MPC, but differs in three key aspects: (i) the fast path uses the planner's own predictions rather than a separately trained amortized policy, (ii) the gate is a mismatch threshold rather than a learned OOD detector, and (iii) their fast path executes open-loop action queues, whereas Fast-TD-MPC's System 1 acts closed-loop with per-step OOD gating.

Dual-Process Control and Adaptive Computation. The dual-process framework of distinguishing a fast, reflexive System 1 from a slow, deliberative System 2 originates in cognitive science (Kahneman, 2011; Daw et al., 2005; Botvinick et al., 2009). A growing body of work adopts this framing for control: Labiosa & Hanna (2026) study learning when to switch between reactive control and deliberative planning, while Li et al. (2026a) learn state-dependent compute schedules for generative control policies. Muppidi et al. (2026) learn planning budgets in real-time RL where the environment does not wait for the agent, directly addressing the think-vs-act trade-off that motivates Fast-TD-MPC. Within the adaptive computation paradigm, Chun et al. (2025) dynamically adjust integration steps for stochastic interpolant policies based on task difficulty, and Salla et al. (2026) learn a prefix-optimal stopping criterion for diffusion policies. Zhao et al. (2026) propose a consistency gate that invokes expensive Best-of-N selection only when the initial rollout appears internally inconsistent, recovering most of the benefit at a fraction of the cost: a design philosophy shared with Fast-TD-MPC. Concurrent work has proliferated in the VLA domain, including speculation-verification frameworks (Wang et al., 2026), uncertainty-driven "cognitive clutch" mechanisms (Li et al., 2026b), and learned binary gates for cheap/expensive policy switching (Chen, 2026; Liu et al., 2026a). Yan et al. (2026) apply the dual-process framework to motion planning with symbolic

Table 26: **MPPI planner warm-start ablation across 4 domains (103 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. *all* denotes the *theoretical* OOD mode; *RG* denotes the *reward-gated* mode. Cold start and K-shifted warm start yield equivalent results across all configurations.

		DMControl		Meta-World		MyoSuite		ManiSkill2	
Config	**Warm-Start**	**Return**	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)
(*all*, τ_{p50})	Cold start	$829_{\pm 55}$	$19.4_{\pm 2.4}$	$85_{\pm 8}$	$24.8_{\pm 1.1}$	$39_{\pm 21}$	$31.4_{\pm 3.6}$	$54_{\pm 55}$	$33.9_{\pm 12.1}$
	K-shifted	$829_{\pm 54}$	$19.1_{\pm 2.5}$	$85_{\pm 8}$	$24.6_{\pm 1.1}$	$38_{\pm 22}$	$31.6_{\pm 3.5}$	$55_{\pm 59}$	$33.9_{\pm 13.9}$
(*all*, τ_{p75})	Cold start	$823_{\pm 55}$	$12.2_{\pm 2.7}$	$85_{\pm 8}$	$18.7_{\pm 1.5}$	$37_{\pm 21}$	$26.6_{\pm 5.8}$	$47_{\pm 54}$	$23.3_{\pm 10.7}$
	K-shifted	$821_{\pm 54}$	$12.3_{\pm 2.7}$	$85_{\pm 8}$	$18.8_{\pm 1.5}$	$39_{\pm 22}$	$26.9_{\pm 5.8}$	$48_{\pm 51}$	$23.5_{\pm 11.6}$
(*all*, τ_{p90})	Cold start	$785_{\pm 58}$	$6.5_{\pm 2.0}$	$85_{\pm 8}$	$12.2_{\pm 1.6}$	$37_{\pm 21}$	$22.6_{\pm 7.6}$	$32_{\pm 42}$	$16.0_{\pm 5.2}$
	K-shifted	$785_{\pm 63}$	$6.5_{\pm 2.0}$	$85_{\pm 8}$	$12.3_{\pm 1.6}$	$35_{\pm 20}$	$22.8_{\pm 7.2}$	$32_{\pm 42}$	$16.4_{\pm 5.8}$
(*all*, τ_{p95})	Cold start	$745_{\pm 74}$	$3.8_{\pm 1.4}$	$85_{\pm 8}$	$9.8_{\pm 1.5}$	$38_{\pm 23}$	$19.8_{\pm 7.6}$	$28_{\pm 46}$	$14.2_{\pm 3.9}$
	K-shifted	$744_{\pm 75}$	$3.8_{\pm 1.3}$	$86_{\pm 8}$	$9.8_{\pm 1.5}$	$39_{\pm 21}$	$19.8_{\pm 7.3}$	$28_{\pm 45}$	$14.4_{\pm 4.8}$
(*all*, τ_{p99})	Cold start	$650_{\pm 92}$	$1.9_{\pm .3}$	$85_{\pm 8}$	$6.2_{\pm 1.3}$	$28_{\pm 17}$	$10.9_{\pm 5.8}$	$20_{\pm 39}$	$11.5_{\pm 3.9}$
	K-shifted	$645_{\pm 93}$	$1.8_{\pm .3}$	$86_{\pm 8}$	$6.2_{\pm 1.3}$	$31_{\pm 16}$	$11.0_{\pm 6.0}$	$20_{\pm 39}$	$11.7_{\pm 4.4}$
(*RG*, τ_{p25})	Cold start	$784_{\pm 68}$	$10.8_{\pm 2.3}$	$85_{\pm 8}$	$12.3_{\pm 1.7}$	$39_{\pm 22}$	$24.5_{\pm 4.6}$	$48_{\pm 43}$	$28.5_{\pm 20.9}$
	K-shifted	$784_{\pm 67}$	$10.8_{\pm 2.3}$	$85_{\pm 8}$	$12.4_{\pm 1.7}$	$39_{\pm 21}$	$24.4_{\pm 5.0}$	$49_{\pm 39}$	$29.4_{\pm 22.8}$
(*RG*, τ_{p50})	Cold start	$764_{\pm 70}$	$8.1_{\pm 1.8}$	$85_{\pm 8}$	$10.0_{\pm 1.5}$	$35_{\pm 21}$	$20.6_{\pm 5.2}$	$43_{\pm 43}$	$25.6_{\pm 18.9}$
	K-shifted	$768_{\pm 70}$	$7.9_{\pm 1.8}$	$85_{\pm 8}$	$9.9_{\pm 1.5}$	$41_{\pm 22}$	$20.0_{\pm 5.3}$	$46_{\pm 43}$	$25.7_{\pm 18.5}$
(*RG*, τ_{p75})	Cold start	$744_{\pm 74}$	$6.4_{\pm 1.5}$	$84_{\pm 8}$	$8.6_{\pm 1.5}$	$39_{\pm 21}$	$15.7_{\pm 4.7}$	$39_{\pm 51}$	$20.9_{\pm 10.8}$
	K-shifted	$741_{\pm 74}$	$6.4_{\pm 1.6}$	$84_{\pm 8}$	$8.6_{\pm 1.5}$	$36_{\pm 20}$	$16.8_{\pm 4.1}$	$34_{\pm 48}$	$19.9_{\pm 10.9}$

Table 27: **MPPI planner warm-start ablation under disturbances across 4 domains (102 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. *all* denotes the *theoretical* OOD mode; *RG* denotes the *reward-gated* mode. Cold start and K-shifted warm start yield equivalent results under perturbation.

		DMControl		Meta-World		MyoSuite		ManiSkill2	
Config	**Warm-Start**	**Return**	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)
(*all*, τ_{p50})	Cold start	$617_{\pm 105}$	$24.5_{\pm 1.9}$	$51_{\pm 11}$	$27.2_{\pm .2}$	$36_{\pm 20}$	$32.2_{\pm 3.9}$	$66_{\pm 80}$	$34.6_{\pm 18.0}$
	K-shifted	$618_{\pm 105}$	$24.5_{\pm 1.9}$	$51_{\pm 11}$	$27.1_{\pm .2}$	$38_{\pm 22}$	$32.0_{\pm 3.2}$	$69_{\pm 75}$	$35.4_{\pm 17.7}$
(*all*, τ_{p75})	Cold start	$607_{\pm 103}$	$19.3_{\pm 2.9}$	$51_{\pm 11}$	$25.6_{\pm .9}$	$38_{\pm 22}$	$27.2_{\pm 5.9}$	$49_{\pm 108}$	$25.5_{\pm 22.3}$
	K-shifted	$609_{\pm 103}$	$19.4_{\pm 2.9}$	$50_{\pm 10}$	$25.4_{\pm .9}$	$40_{\pm 22}$	$27.1_{\pm 6.1}$	$54_{\pm 108}$	$25.9_{\pm 21.4}$
(*all*, τ_{p90})	Cold start	$586_{\pm 101}$	$13.6_{\pm 3.0}$	$49_{\pm 10}$	$22.8_{\pm 1.6}$	$34_{\pm 22}$	$22.9_{\pm 7.8}$	$46_{\pm 86}$	$16.9_{\pm 10.5}$
	K-shifted	$584_{\pm 101}$	$13.5_{\pm 3.0}$	$50_{\pm 11}$	$22.9_{\pm 1.6}$	$33_{\pm 21}$	$22.4_{\pm 8.0}$	$43_{\pm 84}$	$17.1_{\pm 10.0}$
(*all*, τ_{p95})	Cold start	$537_{\pm 102}$	$9.8_{\pm 2.6}$	$50_{\pm 10}$	$21.8_{\pm 1.8}$	$33_{\pm 20}$	$20.0_{\pm 7.8}$	$32_{\pm 62}$	$13.7_{\pm 5.2}$
	K-shifted	$529_{\pm 103}$	$9.6_{\pm 2.5}$	$51_{\pm 10}$	$21.6_{\pm 1.8}$	$32_{\pm 22}$	$19.9_{\pm 7.8}$	$34_{\pm 74}$	$14.2_{\pm 5.7}$
(*all*, τ_{p99})	Cold start	$439_{\pm 101}$	$5.6_{\pm 2.0}$	$50_{\pm 10}$	$20.2_{\pm 2.0}$	$30_{\pm 18}$	$13.3_{\pm 7.8}$	$19_{\pm 46}$	$11.2_{\pm 3.1}$
	K-shifted	$446_{\pm 101}$	$5.6_{\pm 1.9}$	$50_{\pm 10}$	$20.2_{\pm 2.0}$	$31_{\pm 17}$	$13.1_{\pm 7.8}$	$20_{\pm 50}$	$11.1_{\pm 2.7}$
(*RG*, τ_{p25})	Cold start	$566_{\pm 104}$	$17.5_{\pm 2.6}$	$49_{\pm 11}$	$20.9_{\pm 1.9}$	$36_{\pm 22}$	$25.7_{\pm 4.7}$	$62_{\pm 55}$	$25.9_{\pm 25.6}$
	K-shifted	$572_{\pm 103}$	$17.3_{\pm 2.6}$	$50_{\pm 11}$	$20.8_{\pm 1.9}$	$36_{\pm 22}$	$25.2_{\pm 5.2}$	$60_{\pm 52}$	$25.6_{\pm 29.1}$
(*RG*, τ_{p50})	Cold start	$543_{\pm 102}$	$14.3_{\pm 2.5}$	$50_{\pm 10}$	$18.9_{\pm 2.1}$	$33_{\pm 22}$	$21.4_{\pm 5.6}$	$51_{\pm 61}$	$23.1_{\pm 23.6}$
	K-shifted	$546_{\pm 102}$	$14.2_{\pm 2.5}$	$50_{\pm 10}$	$18.9_{\pm 2.1}$	$37_{\pm 23}$	$21.2_{\pm 5.9}$	$59_{\pm 49}$	$21.8_{\pm 23.5}$
(*RG*, τ_{p75})	Cold start	$524_{\pm 102}$	$12.2_{\pm 2.3}$	$49_{\pm 10}$	$17.4_{\pm 2.2}$	$35_{\pm 24}$	$17.7_{\pm 5.3}$	$43_{\pm 60}$	$18.8_{\pm 14.7}$
	K-shifted	$523_{\pm 102}$	$12.2_{\pm 2.4}$	$49_{\pm 10}$	$17.4_{\pm 2.1}$	$34_{\pm 23}$	$18.0_{\pm 5.6}$	$43_{\pm 60}$	$19.8_{\pm 14.1}$

solvers as System 2. However, these works operate in VLA, symbolic planning, or diffusion policy domains. Fast-TD-MPC is, to our knowledge, the first to leverage the dual-process framework within MBRL, where the learned world model and MPPI planner provide a principled System 2 that other domains approximate with less grounded alternatives.

Out-of-Distribution Detection in RL. OOD detection is a mature technique in deep learning (Hendrycks & Gimpel, 2017; Lee et al., 2018), but its application to RL and control settings has received comparatively less attention. Zhang et al. (2024) introduce a distance-based anomaly detection framework for deep RL, directly comparable to our Mahalanobis approach. Kaur et al.

Table 28: **OOD metric ablation across 4 domains (103 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. *all* denotes the *theoretical* OOD mode; *RG* denotes the *reward-gated* mode.

		DMControl		Meta-World		MyoSuite		ManiSkill2	
Config	**Metric**	**Return**	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)
(*all*, τ_{p50})	Maha.	$829_{\pm 55}$	$19.4_{\pm 2.4}$	$85_{\pm 8}$	$24.8_{\pm 1.1}$	$39_{\pm 21}$	$31.4_{\pm 3.6}$	$54_{\pm 55}$	$33.9_{\pm 12.1}$
	Tan.	$819.9_{\pm 53.5}$	$15.4_{\pm 2.4}$	$69.9_{\pm 6.3}$	$21.5_{\pm 1.2}$	$40.2_{\pm 21}$	$27.8_{\pm 4.5}$	$48.0_{\pm 33.3}$	$24.7_{\pm 6.7}$
(*all*, τ_{p75})	Maha.	$823_{\pm 55}$	$12.2_{\pm 2.7}$	$85_{\pm 8}$	$18.7_{\pm 1.5}$	$37_{\pm 21}$	$26.6_{\pm 5.8}$	$47_{\pm 54}$	$23.3_{\pm 10.7}$
	Tan.	$796.9_{\pm 56.1}$	$9.9_{\pm 2.5}$	$69.4_{\pm 6.3}$	$13.6_{\pm 1.2}$	$38.4_{\pm 21.2}$	$19.1_{\pm 5.5}$	$41.1_{\pm 34.8}$	$18.8_{\pm 7.3}$
(*all*, τ_{p90})	Maha.	$785_{\pm 58}$	$6.5_{\pm 2.0}$	$85_{\pm 8}$	$12.2_{\pm 1.6}$	$37_{\pm 21}$	$22.6_{\pm 7.6}$	$32_{\pm 42}$	$16.0_{\pm 5.2}$
	Tan.	$733.8_{\pm 68.9}$	$5.2_{\pm 1.8}$	$69.4_{\pm 6.3}$	$6.3_{\pm 0.8}$	$38.6_{\pm 20}$	$10.6_{\pm 3.8}$	$28.3_{\pm 28.3}$	$12.3_{\pm 1.6}$
(*all*, τ_{p95})	Maha.	$745_{\pm 74}$	$3.8_{\pm 1.4}$	$85_{\pm 8}$	$9.8_{\pm 1.5}$	$38_{\pm 23}$	$19.8_{\pm 7.6}$	$28_{\pm 46}$	$14.2_{\pm 3.9}$
	Tan.	$696.6_{\pm 79.6}$	$3.5_{\pm 0.9}$	$69.9_{\pm 6.4}$	$4.6_{\pm 0.6}$	$33.7_{\pm 19.6}$	$7.4_{\pm 2.1}$	$23.3_{\pm 24.8}$	$11.4_{\pm 1.8}$
(*all*, τ_{p99})	Maha.	$650_{\pm 92}$	$1.9_{\pm .3}$	$85_{\pm 8}$	$6.2_{\pm 1.3}$	$28_{\pm 17}$	$10.9_{\pm 5.8}$	$20_{\pm 39}$	$11.5_{\pm 3.9}$
	Tan.	$650.9_{\pm 91.2}$	$1.9_{\pm 0.2}$	$69.1_{\pm 6.8}$	$3.2_{\pm 0.3}$	$32.4_{\pm 18.7}$	$4.5_{\pm 0.7}$	$19.5_{\pm 21.2}$	$10.7_{\pm 2.1}$
(*RG*, τ_{p25})	Maha.	$784_{\pm 68}$	$10.8_{\pm 2.3}$	$85_{\pm 8}$	$12.3_{\pm 1.7}$	$39_{\pm 22}$	$24.5_{\pm 4.6}$	$48_{\pm 43}$	$28.5_{\pm 20.9}$
	Tan.	$807.1_{\pm 58}$	$12.0_{\pm 2.1}$	$69.9_{\pm 6.6}$	$12.3_{\pm 1.6}$	$41.1_{\pm 21.5}$	$24.3_{\pm 4.5}$	$46.1_{\pm 34.5}$	$25.0_{\pm 11.3}$
(*RG*, τ_{p50})	Maha.	$764_{\pm 70}$	$8.1_{\pm 1.8}$	$85_{\pm 8}$	$10.0_{\pm 1.5}$	$35_{\pm 21}$	$20.6_{\pm 5.2}$	$43_{\pm 43}$	$25.6_{\pm 18.9}$
	Tan.	$786.7_{\pm 63.5}$	$8.6_{\pm 1.7}$	$69.3_{\pm 6.7}$	$11.4_{\pm 1.5}$	$40.6_{\pm 21.2}$	$19.8_{\pm 4.6}$	$43.4_{\pm 29.7}$	$23.4_{\pm 10.3}$
(*RG*, τ_{p75})	Maha.	$744_{\pm 74}$	$6.4_{\pm 1.5}$	$84_{\pm 8}$	$8.6_{\pm 1.5}$	$39_{\pm 21}$	$15.7_{\pm 4.7}$	$39_{\pm 51}$	$20.9_{\pm 10.8}$
	Tan.	$769.1_{\pm 70.1}$	$6.6_{\pm 1.3}$	$69.2_{\pm 6.7}$	$9.6_{\pm 1.3}$	$38.1_{\pm 19.3}$	$17.4_{\pm 4.2}$	$38.0_{\pm 30.7}$	$19.1_{\pm 7.4}$

Table 29: **OOD metric ablation under disturbances across 4 domains (102 tasks, 3 seeds each).** Mean and 95% CI across all tasks in each domain. *all* denotes the *theoretical* OOD mode; *RG* denotes the *reward-gated* mode.

		DMControl		Meta-World		MyoSuite		ManiSkill2	
Config	**Metric**	**Return**	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)	**Succ.** (%)	**Lat.** (ms)
(*all*, τ_{p50})	Maha.	$617_{\pm 105}$	$24.5_{\pm 1.9}$	$51_{\pm 11}$	$27.2_{\pm .2}$	$36_{\pm 20}$	$32.2_{\pm 3.9}$	$66_{\pm 80}$	$34.6_{\pm 18.0}$
	Tan.	$606.0_{\pm 103.5}$	$20.5_{\pm 2.2}$	$42.7_{\pm 8.4}$	$25.9_{\pm 0.8}$	$34.8_{\pm 20.8}$	$27.3_{\pm 5.2}$	$57.9_{\pm 32.6}$	$26.7_{\pm 8.1}$
(*all*, τ_{p75})	Maha.	$607_{\pm 103}$	$19.3_{\pm 2.9}$	$51_{\pm 11}$	$25.6_{\pm .9}$	$38_{\pm 22}$	$27.2_{\pm 5.9}$	$49_{\pm 108}$	$25.5_{\pm 22.3}$
	Tan.	$571.4_{\pm 99.5}$	$14.8_{\pm 2.7}$	$43.0_{\pm 8.4}$	$22.0_{\pm 1.7}$	$35.0_{\pm 20.4}$	$19.6_{\pm 6}$	$45_{\pm 38.6}$	$19.3_{\pm 10.8}$
(*all*, τ_{p90})	Maha.	$586_{\pm 101}$	$13.6_{\pm 3.0}$	$49_{\pm 10}$	$22.8_{\pm 1.6}$	$34_{\pm 22}$	$22.9_{\pm 7.8}$	$46_{\pm 86}$	$16.9_{\pm 10.5}$
	Tan.	$503.0_{\pm 97.8}$	$10.1_{\pm 2.6}$	$42.1_{\pm 8.4}$	$14.5_{\pm 2.3}$	$34.3_{\pm 20.4}$	$12.6_{\pm 4.2}$	$31.4_{\pm 34}$	$13.9_{\pm 3.5}$
(*all*, τ_{p95})	Maha.	$537_{\pm 102}$	$9.8_{\pm 2.6}$	$50_{\pm 10}$	$21.8_{\pm 1.8}$	$33_{\pm 20}$	$20.0_{\pm 7.8}$	$32_{\pm 62}$	$13.7_{\pm 5.2}$
	Tan.	$461.4_{\pm 99.9}$	$7.8_{\pm 2.4}$	$41.9_{\pm 8.3}$	$12.5_{\pm 2.3}$	$33.2_{\pm 20.1}$	$8.2_{\pm 2.9}$	$27.8_{\pm 27.8}$	$11.5_{\pm 0.9}$
(*all*, τ_{p99})	Maha.	$439_{\pm 101}$	$5.6_{\pm 2.0}$	$50_{\pm 10}$	$20.2_{\pm 2.0}$	$30_{\pm 18}$	$13.3_{\pm 7.8}$	$19_{\pm 46}$	$11.2_{\pm 3.1}$
	Tan.	$413.3_{\pm 99.9}$	$5.1_{\pm 2.1}$	$41.6_{\pm 8.1}$	$10.8_{\pm 2.2}$	$32.7_{\pm 19.4}$	$5.4_{\pm 1.8}$	$21.5_{\pm 23.6}$	$10.0_{\pm 1.1}$
(*RG*, τ_{p25})	Maha.	$566_{\pm 104}$	$17.5_{\pm 2.6}$	$49_{\pm 11}$	$20.9_{\pm 1.9}$	$36_{\pm 22}$	$25.7_{\pm 4.7}$	$62_{\pm 55}$	$25.9_{\pm 25.6}$
	Tan.	$585.2_{\pm 102.4}$	$17.6_{\pm 2.3}$	$42.5_{\pm 8.4}$	$19.2_{\pm 1.9}$	$40.6_{\pm 20}$	$24.1_{\pm 4.8}$	$54.3_{\pm 32.9}$	$21.3_{\pm 7}$
(*RG*, τ_{p50})	Maha.	$543_{\pm 102}$	$14.3_{\pm 2.5}$	$50_{\pm 10}$	$18.9_{\pm 2.1}$	$33_{\pm 22}$	$21.4_{\pm 5.6}$	$51_{\pm 61}$	$23.1_{\pm 23.6}$
	Tan.	$562.5_{\pm 102.1}$	$14.2_{\pm 2.4}$	$42.7_{\pm 8.5}$	$18.1_{\pm 2}$	$38.6_{\pm 20.5}$	$21.2_{\pm 4.6}$	$53.5_{\pm 36.7}$	$20.5_{\pm 9}$
(*RG*, τ_{p75})	Maha.	$524_{\pm 102}$	$12.2_{\pm 2.3}$	$49_{\pm 10}$	$17.4_{\pm 2.2}$	$35_{\pm 24}$	$17.7_{\pm 5.3}$	$43_{\pm 60}$	$18.8_{\pm 14.7}$
	Tan.	$539.3_{\pm 103.2}$	$12.2_{\pm 2.4}$	$42.7_{\pm 8.5}$	$16.7_{\pm 2.1}$	$37.6_{\pm 19.4}$	$18.5_{\pm 4.1}$	$45.4_{\pm 31.7}$	$15.2_{\pm 2.3}$

(2023) apply conformal OOD detection to time-series data for safety-critical autonomous systems, and Tang & Zhi (2026) use OOD detection to trigger intervention in action-chunking policies — a gating mechanism analogous to Fast-TD-MPC's, though in the VLA domain. Other works provide formal guarantees for OOD detection in deep RL via transition estimation (Prashant et al., 2025) and benchmark OOD detection methods specifically for RL (Mittag et al., 2026). Fast-TD-MPC repurposes Mahalanobis OOD detection (Lee et al., 2018) as a per-timestep routing mechanism between an amortized policy and a model-based planner — using distributional shift not for anomaly alerting, but as a signal for adaptive compute allocation, a role that, to our knowledge, has not been explored.