DISAGREEMENT IN IMAGINATION: A CROSS-TIMESCALE CONSISTENCY SIGNAL FOR ROBUST MODEL-BASED CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning inside a learned world model is sample-efficient but fragile: prediction errors compound with rollout depth, and execution-time disturbances cause plans to drift from reality. Every standard detector of model error such as filtering residuals, reconstruction losses requires an observation to compare against, and inside imagination no observation exists. We study a quantity that does not need one —the disagreement between two predictors of the *same* imagined step, one coarse and trained over abstract steps, one fine and trained over primitive steps. This *cross-timescale disagreement* is computable from predictions alone, needs no future observation, and is directional: it states how the fine predictor would revise the coarse one. It is consumed at three clocks: as a consistency penalty during planning, as a training regularizer that sharpens the abstract model, and, once the imagined step is realized, as the residual an online controller cancels. On 80 tasks across DMControl, Meta-World, and MyoSuite under combined observation and action noise, HiWM achieves the highest IQM among evaluated methods with a margin that grows monotonically with noise intensity. Ablations on a completed factorial confirm super-additive coupling between the feedback pathway and the controller. Freezing the fast predictor at its initial observation as the configuration closest to a standard posterior, loses half the benefit, and two prediction rates without upward flow perform worse than a single rate. The mechanism transfers zero-shot to a Franka Panda robot, where disagreement is 2.4$\times$ larger than in simulation and the controller's corrections scale proportionally.

1 INTRODUCTION

Planning with learned world models is among the most sample-efficient paradigms in reinforcement learning Janner et al. (2019); Chua et al. (2018). Every decision the agent makes is therefore made about a future it has not seen. Under deployment conditions with sensor noise, actuator drift, stochastic contacts, that future is doubly unreliable: prediction errors accumulate with rollout depth Asadi et al. (2019); Jiang (2024); Janner et al. (2019), and the trajectory the agent actually walks leaves the one it planned Zhang et al. (2020); Pinto et al. (2017). The standard mitigations such as multi-step losses, trajectory-level models, truncated horizons all act at training time and cannot say anything about a rollout that is already under way.

The deeper issue is that an agent planning in imagination has no way to tell, *at plan time*, which parts of its imagined future are unreliable. Every standard detector of model error needs a real observation to compare against: a filtering residual compares a prediction to an encoding of o_t, a reconstruction error decodes back to observation space. Inside imagination there is no observation and never will be: at rollout depth $h \geq 1$ every quantity available to the agent is model-generated. An admissible reliability signal inside imagination must be (i) computable from predictions alone, (ii) cheap enough to evaluate at every step of every candidate trajectory, and (iii) ideally something the agent can act on at execution time rather than merely avoid.

Our starting point is that a signal with all three properties is available for the price of a second prediction rate: we add two things to a standard latent MPC agent, a fine predictor that runs alongside the coarse one, and one scalar per imagined step that measures how much the two disagree. A coarse

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.

predictor rolling out over a deep horizon and a fine predictor rolling out over a shallow horizon fail differently. Their disagreement at any imagined step —the KL divergence between a *fast-model-conditioned abstract belief* and the *abstract prior* requires no observation, because both sides are predictions. And unlike a variance, it is *directional*: it says not only that the coarse prediction is unreliable but which way the fine predictor would move it, making it usable as a training target and as a control residual. The same functional, evaluated on a transition the agent has actually taken, is execution error. One quantity therefore exists at three clocks: at imagination time it scores each candidate trajectory for the MPC objective (*consistency penalty*); at execution time the realized counterpart drives an online corrective controller; at training time it regularizes the abstract model, pulling the abstract prior toward the fast-model-conditioned belief.

Four findings support this design. Imagined disagreement is rank-informative about error the agent has not yet made. Freezing the fast predictor at h=0 —the configuration matching a standard filtering residual costs 8 IQM points, roughly half the total benefit. Two prediction rates help only when the fine rate feeds upward: without it, the two-rate model is no better than a single-rate one evaluated under our noise protocol. The plan-time penalty and the execution-time controller are approximately additive on their own, but coupling them through the shared signal produces super-additive robustness at every tested noise level. The disagreement rises monotonically with parametric dynamics shift and is 2.4$\times$ larger on physical hardware than in simulation, confirming it is a general mismatch detector.

Our contributions are:

- A reliability signal for model-based planning that is defined on predicted rather than observed evidence, with the imagination-time recursion that computes it and a measurement of how informative it is.
- A single-signal coupling of planning, execution and training, with the credit assignment reported as a complete factorial over the signal's consumers.
- An evaluation under combined noise on 80 tasks, an image-based compositional benchmark, and real-robot deployment to a Franka Panda. The idea generalizes past the architecture we use: in any model that predicts at more than one rate, cross-rate disagreement is an error signal that exists inside imagination.

2 Related Work

2.1 World Models and Planning in Uncertainty

Learned world models for RL span from early recurrent approaches Ha & Schmidhuber (2018) to modern latent-space methods. DreamerV3 Hafner et al. (2025) employs an RSSM with discrete categorical latents and trains policies entirely in imagined rollouts, achieving broad generality. TD-MPC2 Hansen et al. (2023) takes a complementary approach: a decoder-free implicit model trained with temporal-difference losses, combined with MPPI planning and a learned terminal value function. DC-MPC Scannell et al. (2024) replaces the continuous latent with a discrete codebook while retaining the same planning procedure. Transformer-based Micheli et al. (2022); Robine et al. (2023); Zhang et al. (2023) and diffusion-based Alonso et al. (2024) world models have shown promise on Atari but have not been widely adopted for continuous control. All three deployed approaches roll a single predictor forward during planning, and none has anything to compare that rollout against —a gap that becomes load-bearing under noise.

A central challenge across world-model methods is that per-step prediction errors accumulate exponentially during multi-step rollouts Lambert et al. (2022); Talvitie (2014); Robine et al. (2023). Training-time mitigations include multi-step prediction losses Benechehab et al. (2023), trajectory-level dynamics models Lambert et al. (2021), and truncated rollout horizons Hafner et al. (2020). These constrain the model before deployment but cannot act during a rollout. Ensemble-based approaches Chua et al. (2018) quantify epistemic uncertainty via bootstrap disagreement and penalize uncertain rollouts. Plan2Explore Sekar et al. (2020) uses latent disagreement for directed exploration. These are effective as planning penalties but they are read-only diagnostics: they never enter the model's training objective, they have no directional counterpart that a controller can regress on.

2.2 Temporally Structured Models

Director Hafner et al. (2022) extends DreamerV2 with a hierarchical goal-setting mechanism. THICK Gumbsch et al. (2023) learns a two-level world model predicting only context-changing events. Hieros Mattes et al. (2024) applies hierarchical imagination with structured state-space backbones. Puppeteer Hansen et al. (2025) stacks two TD-MPC2 agents for humanoid control. HWM Zhang et al. (2026) trains world models at multiple temporal scales. In all of these, information moves exclusively downward: abstract levels set goals or condition primitive predictions, but primitive-level evidence never refines abstract dynamics.

Corrective controllers have been explored alongside model-based planners: Reflexive World Models Brito & McNamee (2025) use world-model predictions as implicit reference trajectories; Residual MPC Jeon et al. (2025) blends RL residuals with kinodynamic MPC; TD-M(PC)^2 Lin et al. (2025) adds a policy constraint to TD-MPC2. In all these methods, the corrector and planner share only actions or rewards —none share the prediction-error signal as a joint training and correction mechanism. The property none of these families has is a signal defined on *predicted* evidence that both trains the model and has a cancellable realized counterpart.

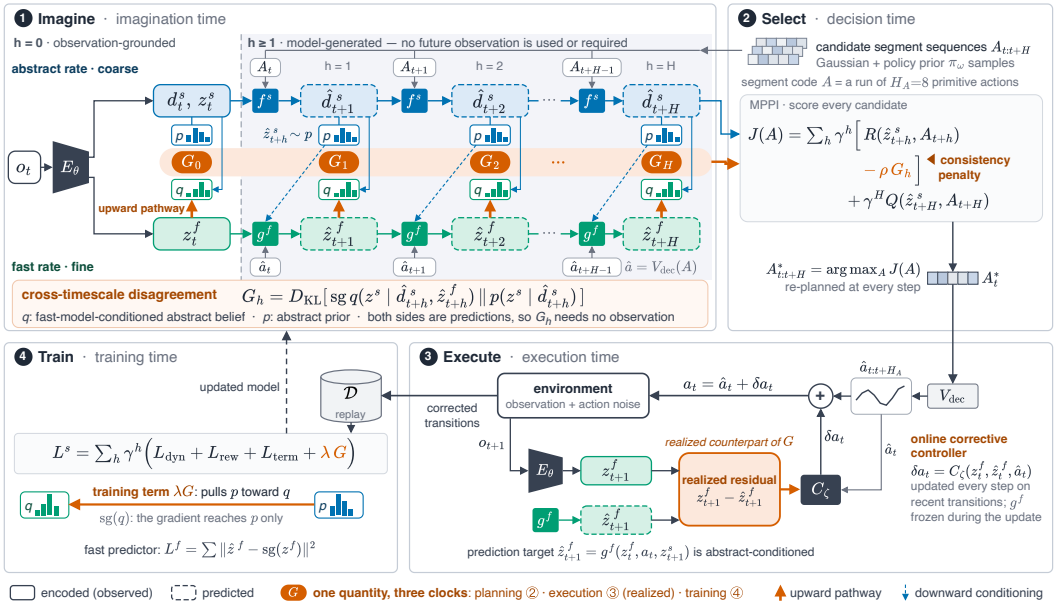

Figure 1: One quantity, computed where no observation exists, used at three clocks. (1) From o_t, an abstract predictor (coarse, deep; blue) and a fast predictor (fine, shallow; green) are rolled out together for every MPPI candidate. At each imagined step, the fast-model-conditioned abstract belief q and the abstract prior p are both predictions, so their cross-timescale disagreement G_h is defined without any future observation; only h = 0 is observation-grounded. (2) G_h enters the MPC objective as the consistency penalty $-\rho G_h$. (3) The selected segment is decoded and executed; an online corrective controller acts on the realized counterpart of G. (4) Corrected transitions enter replay, where λG pulls the prior toward the stop-gradient belief.

3 Method

HiWM adds two things to a standard latent MPC agent: a second prediction rate, and one scalar that measures how much the two rates disagree about the same imagined step.

3.1 Imagination

From an observation o_t the agent must score H_s segments of a future it will never observe. Any signal usable here must be a function of predictions only.

The imagination recursion. For a candidate segment sequence $A_{t:t+H}$ with decoded reference actions $\hat{a}$, initialized at $\hat{z}^f_t = z^f_t = E_\theta(o_t)$:

$$\hat{d}^s_{t+h} = f^s\Big(\hat{d}^s_{t+h-1},\, \hat{z}^s_{t+h-1},\, A_{t+h-1}\Big), \qquad \hat{z}^s_{t+h} \sim p\Big(z^s \mid \hat{d}^s_{t+h}\Big),$$
$$\hat{z}^f_{t+h} = g^f\Big(\hat{z}^f_{t+h-1},\, \hat{a}_{t+h-1},\, \hat{z}^s_{t+h}\Big), \qquad G_h = D_{\mathrm{KL}}\Big[\,\mathrm{sg}\, q\Big(z^s \mid \hat{d}^s_{t+h},\, \hat{z}^f_{t+h}\Big) \,\|\, p\Big(z^s \mid \hat{d}^s_{t+h}\Big)\Big]. \tag{1}$$

$q(z^s \mid d^s, \hat{z}^f)$ is the *fast-model-conditioned abstract belief*; $p(z^s \mid d^s)$ is the *abstract prior*; G_h is the *cross-timescale disagreement*. It is not a filtering residual (there is no observation to filter) and not an epistemic variance (it is directional, identifying the distribution the prior should move toward, which is what lets it appear in a training objective and have a realized counterpart to regress on).

Only $h=0$ is observation-grounded; for $h \geq 1$ the evidence is entirely model-generated. No future observation is used or required.

For $h \geq 1$, G_h measures the disagreement between the abstract model's extrapolation from its own recurrent state and the fast model's forward simulation, where the two are trained at different depths and therefore fail differently. The stop-gradient on the belief ensures only the prior receives gradients, so the belief cannot reduce the divergence by ignoring its evidence.

The agent therefore holds a per-step reliability number for every candidate segment sequence, available before any action is taken.

3.2 Planning

Searching directly over 120 primitive actions is impractical, so the planner searches over codes for short action runs. A residual vector-quantized autoencoder encodes each run of $H_A = 8$ primitive actions into a segment action A_t and decodes it back to reference actions $\hat{a}_{t:t+H_A}$. The component is standard in action abstraction for MPC Zeghidour et al. (2022); Dadashi et al. (2022); Luo et al. (2023).

Objective We solve, by MPPI Williams et al. (2017; 2018) over segment sequences,

$$A^*_{t:t+H_{\mathrm{plan}}} \;=\; \arg\max_{A} \sum_{h=0}^{H_{\mathrm{plan}}} \gamma^h \big[\, R\big(\hat{z}^s_{t+h}, A_{t+h}\big) \;-\; \rho\, G_h \,\big] \;+\; \gamma^{H_{\mathrm{plan}}} Q\Big(\hat{z}^s_{t+H_{\mathrm{plan}}}, A_{t+H_{\mathrm{plan}}}\Big), \tag{2}$$

We call $-\rho\, G_h$ the *consistency penalty*. It trades predicted reward for agreement between the two rates, steering the plan toward regions where the two rates concur.

The plan is a sequence of reference actions, and the moment one is executed the imagined step becomes an observed one.

3.3 Coupling

The agent executes $\hat{a}_t$ and observes o_{t+1}. The same functional, evaluated on a transition that actually happened, is an execution residual.

The executed action is $a_t = \hat{a}_t + C_\zeta(z^f_t, \hat{z}^f_t, \hat{a}_t)$, where C_ζ produces corrective actions based on the discrepancy between the observed latent state z^f_t and the predicted latent state $\hat{z}^f_t$. The controller is updated with

$$\mathcal{L}_c \;=\; \big\|\hat{z}^f_{t+1}(a_t) - z^f_{t+1}\big\|_2^2 \;+\; \lambda_{\mathrm{act}} \big\|a_t - \hat{a}_t\big\|_2^2, \qquad \lambda_{\mathrm{act}} = 0.01, \tag{3}$$

The fast model's weights are frozen during controller updates. The controller is specific to the surrounding model rather than a generic residual policy. The controller's prediction target $\hat{z}^f_{t+1} = g^f(z^f_t, a_t, z^s_{t+1})$ is conditioned on the abstract state, and its reference actions come from a consistency-penalized plan.

The abstract model is trained via:

$$\mathcal{L}^s = \mathbb{E}\left[\sum_{h=0}^{H_s} \gamma^h \left(\mathcal{L}_{\mathrm{dyn}} + \mathcal{L}_{\mathrm{rew}} + \mathcal{L}_{\mathrm{term}} + \lambda\, G\right)\right], \quad \lambda = 0.5 \tag{4}$$

Table 1: **Performance under combined observation and action noise** (IQM, 95% bootstrap CI), rows grouped by number of prediction rates and direction of flow.

		Protocol A: state-based			Protocol B: image-based	
Method	Rates	DMC	Meta-World	MyoSuite	Subtasks (of 4)	Full task
HiWM (ours)	two, bidirect.	**0.84**±.01	**0.83**±.01	**0.72**±.02	**3.2**±0.3	**48**±8%
DC-MPC	one	0.78±.02	0.71±.03	0.61±.02	2.5±0.3	28±7%
TD-MPC2	one	0.76±.01	0.69±.01	0.58±.02	2.4±0.3	25±6%
DreamerV3	one	0.71±.02	0.63±.02	0.55±.02	2.0±0.3	16±5%
TD-MPC	one	0.68±.01	0.58±.02	0.48±.01	2.1±0.3	14±6%
Puppeteer	two, top-down	0.74±.02	0.67±.03	0.57±.03	2.2±0.4	22±7%
HWM	two, top-down	0.72±.03	0.65±.03	0.55±.03	2.1±0.4	18±6%
Director	two, top-down	0.72±.04	0.66±.05	0.56±.03	2.2±0.4	17±6%

The cross-timescale disagreement G has three consumers: (1) *Training regularizer:* G enters the abstract model's objective, pulling the prior toward the fast-model-conditioned belief and sharpening abstract predictions. (2) *Consistency penalty:* $-\rho G_h$ penalizes plan-time trajectories where the two rates disagree. (3) *Online controller:* The realized residual $\|z_t^f - \hat{z}_t^f\|$ drives corrective actions.

The consistency penalty lowers one-step execution error before any correction is applied, and the controller's corrections lower the abstract prior's held-out cross-entropy, because corrected executions leave less execution noise in the replay data that trains it.

4 Experiments

We evaluate HiWM on four continuous-control benchmarks under a unified noise protocol, testing aggregate performance, credit assignment, and the specific mechanisms underlying the cross-timescale disagreement. All results report IQM with 95% stratified bootstrap confidence intervals.

4.1 Implementation Details

Benchmarks. DMControl Tassa et al. (2018) (30 tasks): locomotion and balancing, including Dog (223-dim obs, 38-dim actions) and Humanoid (67-dim obs, 21-dim actions). Meta-World Yu et al. (2020) (45 tasks): contact-rich robotic manipulation (39-dim obs, 4-dim actions). MyoSuite Caggiano et al. (2022) (5 tasks): musculoskeletal control. Franka Kitchen (4 task sequences, image-based): compositional manipulation requiring 4 sequential subtasks, with a convolutional encoder and noise injected in pixel space.

Noise protocol. Combined noise: observation noise via an Ornstein-Uhlenbeck process ($\theta = 0.15$, $\sigma_{\text{obs}} = 0.05$) producing temporally correlated perturbations, and action noise via sinusoidal drift ($A = 0.02$, $f = 0.1$ Hz) plus Gaussian perturbations ($\sigma_{\text{act}} = 0.05$). This models realistic deployment conditions where both sensing and actuation are imperfect. More details are shown in Appendix B.1.

Baselines. Flat: TD-MPC2 Hansen et al. (2023), DC-MPC Scannell et al. (2024), DreamerV3 Hafner et al. (2025), TD-MPC Hansen et al. (2022). Top-down two-rate: Director Hafner et al. (2022), Puppeteer Hansen et al. (2025), HWM Zhang et al. (2026). All baselines use official implementations with default hyperparameters and identical noise injection.

4.2 Main results

Under combined noise (Table 1), HiWM achieves the highest IQM among evaluated methods on all four suites, with margins of +6 (DMC), +12 (MW), +11 (Myo), and +23 (Franka) over the best flat baseline, and it has the lowest cross-seed variance. The advantage is noise-conditional: the $\sigma = 0$ column collapses to 1-3 points. A $1.2\times$ parameter difference therefore buys 1–3 points in clean conditions and cannot account for 6–12 under noise. The top-down two-rate systems (Puppeteer,

Table 2: **Credit assignment** (IQM, 4 tasks, 5 seeds, 95% CI). F: upward pathway producing G (with the training regularizer); P: consistency penalty; C: controller. P and C consume F, so the factorial has six cells. $S_{A\times B} = (\Delta_{AB} - \Delta_A - \Delta_B)/\text{Full}$, with Δ the drop from the full model.

F	P	C	Configuration	Dog Run	Hum. Walk	Assembly	Hand Pose
✓	✓	✓	Full model	**0.84**±.02	**0.81**±.02	**0.83**±.02	**0.74**±.02
✓	✓	–	No controller	0.78±.02	0.74±.02	0.67±.03	0.63±.02
✓	–	✓	No penalty	0.80±.02	0.77±.02	0.79±.02	0.71±.02
✓	–	–	Training use only	0.73±.02	0.70±.02	0.63±.03	0.60±.02
–	n/a	✓	No pathway (top-down)	0.68±.02	0.64±.03	0.71±.02	0.61±.02
–	n/a	–	No pathway, no controller	0.56±.03	0.51±.03	0.53±.03	0.46±.03
TD-MPC2 (single-rate reference)				0.73±.02	0.71±.02	0.68±.01	0.60±.02
TD-MPC2 + the identical controller				0.77±.02	0.75±.02	0.74±.02	0.65±.02
HiWM-1T (one rate, all else fixed)				0.76±.02	0.73±.02	0.71±.02	0.63±.02
$S_{F\times C}$ (pathway × controller)				0.071±.032	0.074±.034	0.024±.022	0.054±.028

HWM, Director) perform at or below the flat baselines under noise —Puppeteer 0.74 and HWM 0.72 sit at or below DC-MPC 0.78 and TD-MPC2 0.76. In Figure 2 their degradation profiles are close to the profile of HiWM with its upward pathway removed. Adding a coarse prediction level gives a planner a longer reach and a second place to make mistakes; under perturbation the two effects cancel unless evidence can travel back up.

On Franka Kitchen (Table 1) HiWM completes 0.8 more subtasks per episode than TD-MPC2 but succeeds at the full four-subtask sequence 23 percentage points more often. The contrast between the two columns is the finding, because an error that degrades the current subtask prevents the next one from being attempted.

4.3 Ablation study

Removing consumers one at a time degrades performance monotonically (Table 2), and removing the pathway itself costs more than removing both of its deployment-time consumers. With both deployment-time consumers switched off (row 4), the training-time use of G alone is worth +17 points on Dog Run (0.73 vs. 0.56): the pathway improves the abstract predictor itself and not merely the inference run at deployment. This is the property that separates it most sharply from a read-only uncertainty estimate. No Feedback (0.68) falls below the flat reference (0.73) on three of four tasks. A second prediction rate without upward flow is worse than a single rate —the additional prediction level compounds errors without corrective input.

The corrective controller on a flat backbone recovers 36-40% of the total gap. This confirms that correction alone is valuable but insufficient: on a flat backbone the controller has no abstract-conditioned prediction target and no consistency-penalized reference to follow. HiWM-1T (single timescale, all else fixed) confirms that the hierarchy with feedback accounts for the dominant share.

Ensemble UQ is an effective planning penalty — the IQM gap is only 3 points, at the cost of five fast-model forward passes per imagined step. The difference lies in the other two rows. The abstract prior under ensemble UQ (0.538) is 32% worse than under G (0.407), close to the No-Feedback value (0.595), because variance is a read-only diagnostic that does not enter the training objective. A variance has no direction for the prior to move toward. And a variance has no realized counterpart: a controller can avoid it but cannot cancel it, so the planner and the controller are no longer consuming the same quantity and the coupling disappears ($S_{F\times C}$ falls from 0.024–0.074 to -0.003–0.011). G's structural advantages (prior improvement, cancellable counterpart, coupling) are unavailable to any ensemble-based alternative.

$S_{F\times C}$ (feedback × correction) is positive on all four tasks (range 0.024-0.074). $S_{F\times C}$ rises monotonically across five noise levels on Dog Run ($0.024 \to 0.071 \to 0.107 \to 0.155 \to 0.202$). The super-additivity is carried by the training-time channel, which is also what the training-only cell (+10–19 points) and the ensemble comparison ($S \approx 0.008$, no training channel) say independently.

Table 3: **Substitutes for the signal** (4 tasks, 5 seeds; every IQM interval is ±.02). *Stale*: $\hat{z}_h^f$ replaced by its h=0 value at every depth, applied at test time to the trained model. *Var.*: pathway removed; the variance of a 5-member fast-predictor ensemble takes the planning penalty's place (ρ' swept, magnitude-matched; a cost-matched 3-member ensemble scores 1 point lower). Prior CE at 2M steps.

	IQM ↑			prior CE ↓		$S_{F\times C}$	
Task	G_h (full)	stale	var.	G_h	var.	G_h	var.
Dog Run	**0.84**	0.76	0.81	**0.407**	0.538	**0.071**	0.008
Humanoid Walk	**0.81**	0.73	0.78	**0.392**	0.521	**0.074**	0.011
Assembly	**0.83**	0.75	0.80	**0.401**	0.530	**0.024**	−0.003
Hand Pose	**0.74**	0.66	0.71	**0.418**	0.552	**0.054**	0.006

Table 4: **Necessity of the forward rollout** (IQM ↑, 5 seeds, 95% CI). *Stale evidence* keeps the conditioning structure but reuses $\hat{z}_t^f$ at every $h \geq 1$ instead of rolling the fast predictor forward — the configuration in which the only grounded information is the current observation.

Configuration	Dog Run	Hum. Walk	Assembly	Hand Pose
Full (fast predictor rolled forward)	**0.84**±.02	**0.81**±.02	**0.83**±.02	**0.74**±.02
Stale evidence ($\hat{z}^f$ frozen at h=0)	0.76±.02	0.73±.02	0.75±.02	0.66±.02
No pathway	0.68±.02	0.64±.03	0.71±.02	0.61±.02

4.4 Coupling Signal Analysis

We take a trained model, record G_h as computed at plan time from $\hat{z}_{t+h}^f$, and recompute it afterwards by substituting the realized latent $z_{t+h}^f = E_\theta(o_{t+h})$, over 1000 planning steps per task. The rank correlation between imagined and realized G is 0.78 at $h = 1$ and decays to 0.23 at $h = 15$. The evidence itself is a prediction, and the fast predictor's own error accumulates, but the penalty still works because MPPI needs a correct *ordering* of candidates, not a correct magnitude. Sorting trajectories by imagined G_h and measuring realized disagreement per decile gives a monotone relation on both tasks, with the top decile of imagined G corresponds to 2.8× higher realized G than the bottom decile (Figure 2b).

Moderate positive correlation ($r \approx 0.5$) between planning-phase and execution-phase disagreement indicates they carry related but non-redundant information, justifying the use of a shared signal. Both are negatively correlated with episode return ($r \approx -0.65$), confirming G is a meaningful performance proxy.

Table 4 answers the question that most directly separates this quantity from an RSSM belief. Freezing the fast predictor at $h = 0$ —the configuration closest to a standard posterior, where the only observation-grounded information is the current observation, drops IQM from 0.84 to 0.76 on Dog Run, losing roughly half the total feedback benefit. With stale evidence G_h varies little across a rollout and the penalty loses much of its ability to discriminate between candidates. The forward rollout of the fast predictor is where the novelty resides: conditioning on a *prediction of the future* ($\hat{z}_{t+h}^f$ from rolling g^f forward) versus conditioning on an *encoding of the present* (z_t^f from the encoder).

4.5 Mechanistic Analysis

We measure abstract prediction error as a function of rollout depth on the two locomotion tasks where the rollout is deep enough (Figure 2c). The exponential growth rate of abstract prediction error (cross-entropy vs. rollout depth h) is 1.5× slower for the full model than for the model without feedback on Dog Run, and 1.6× slower on Humanoid Walk. The gap widens with depth: at h=20, prediction error is 3.4× lower on Dog Run and 3.7× lower on Humanoid Walk. The feedback pathway therefore mitigates compounding errors rather than merely improving initial prediction quality.

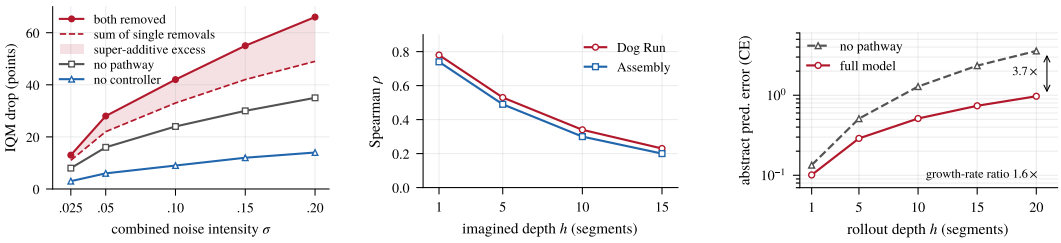

Figure 2: **(a)** Cost of removing components as noise grows. **(b)** Spearman correlation between plan-time G_h and its value recomputed from realized latents (1000 planning steps per task). **(c)** Abstract prediction error against rollout depth on Humanoid Walk, with and without the pathway.

Table 5: **Both directions of the coupling** (Dog Run, $\sigma = 0.05$, 5 seeds; *Effect*: change on removal).

	Measurement	Full model	Comparison	Effect
plan → execution	one-step execution error	0.039	0.066, no penalty	+69%
execution → plan	abstract prior CE at 2M steps	0.407	0.481, no controller	+18%
controller scale	mean correction $\|\delta a_t\|$	0.030	0.412, reference $\|\hat{a}_t\|$	7.3% of reference
plan behaviour	trajectory reward / G	4.21 / 0.14	4.53 / 0.37 at ρ=0	less reward, less G

The consistency penalty lowers realized execution error by 41% before any online correction is applied, because plans steered toward regions where the two rates concur produce more predictable trajectories. The controller's corrections lower the abstract model's training-time cross-entropy by 18%: corrected execution produces cleaner replay data.

The correction magnitude (0.030) is 7.3% of the reference action, confirming the controller operates as a fine-grained corrective signal, not a dominant override. Its magnitude correlates with prediction error at $r = 0.73$, scaling up where the error is large without destabilizing execution.

4.6 Dynamics shift and real-robot deployment

The signal should respond to model mismatch of any kind, not only to the perturbations injected during training. Models are trained under the standard noise protocol and evaluated zero-shot under parametric dynamics shifts. G rises monotonically with shift magnitude across all tested conditions: from 0.14 (nominal) to 0.24 (mass $\times$0.7), 0.32 (friction $\times$0.5), and 0.35 (actuator delay 2 steps) (Table 6). HiWM retains 82-93% of nominal performance compared to 62-80% for TD-MPC2, with the largest advantage under actuator delay (+20 pp retention). At friction $\times$0.5, the margin narrows to 12 points because the fast predictor itself degrades, and the remaining adaptation channel is the controller's online updates.

We transfer HiWM to a 7-DoF Franka Emika Panda with a wrist force/torque sensor and a RealSense D435 for state estimation, 20 Hz control, trained in matched MuJoCo simulation under the same noise protocol and deployed *zero-shot*; 20 trials per method per task with randomized initial object positions, 2 policy seeds alternated across trials, three tasks of increasing stage count (peg insertion, pick-and-place, two-block stacking), TD-MPC2 and TD-MPC2 + controller run on the same hardware, Wilson intervals throughout.

On the Franka platform, G is 2.4$\times$ larger than in simulation (0.34 vs. 0.14), and the controller's correction magnitude scales proportionally (0.052 vs. 0.030); the correction-to-error ratio is unchanged ($0.052/0.068 \approx 0.030/0.039$), confirming the controller adapts to actual mismatch rather than overriding the plan. The advantage grows with task complexity (+30 pp on Peg Insertion, +40 pp on Stack Two Blocks).

Appendix overview. The appendix provides the complete training algorithm with the imagination recursion indexed inside the MPPI loop (Appendix A), the full architectural specification and noise protocol (Appendix B), and three broad hyperparameter plateaus supporting the single-configuration

Table 6: **A ladder of model mismatch.** Top: performance retained under zero-shot dynamics shift (Dog Run, 3 seeds; all eight shifts in App. K). Bottom: success on a Franka Panda, 20 trials per method per task, Wilson 95% intervals. Mean G is 0.14 at nominal.

Condition	HiWM	TD-MPC2 + controller	TD-MPC2	mean G
Mass $\times 0.7$	**92%**	83%	78%	0.24
Friction $\times 0.5$	**86%**	74%	68%	0.32
Actuator delay, 2 steps	**82%**	70%	62%	0.35
Hardware, peg insertion	**85%** [64–95]	70% [48–85]	55% [34–74]	0.34
Hardware, pick-and-place	**80%** [58–92]	60% [39–78]	45% [26–66]	0.38
Hardware, two-block stacking	**65%** [43–82]	40% [22–61]	25% [11–47]	0.41

claim (Appendix H). Extended ablation results include noise-attribution analysis isolating which mechanism addresses which noise source, feedback pathway quality metrics confirming the 31% prior improvement and the execution-to-planning coupling (+18%), synergy indices across five noise levels, and component isolation separating action abstraction from the two-rate structure (Appendix C). Detailed analyses of the controller on a flat backbone, controller correction statistics, planning penalty behavior, and transient perturbation recovery appear in Appendix D. Per-task results for the hierarchical baselines with clean-condition controls are in Appendix E; the full rank-correlation table and stale-evidence ablation details are in Appendix F; stop-gradient ablations are in Appendix G. The dynamics shift ladder, real-robot protocol and mechanism measurements, and compositional task details appear in Appendices I–K. Per-task DMControl results (26/30 best), sample efficiency, cross-task correlations, and computational cost complete the supplement (Appendices L–P).

5 Limitations and Discussion

At $\sigma = 0$ the margin is 1-3 IQM (Table 1, $\sigma = 0$ column). The mechanism is designed for the case where the model is wrong, and when it is not wrong the second prediction rate is overhead. We keep this as a limitation and also use it as the control that separates the mechanism from the $1.2\times$ parameter difference. Under friction $\times 0.5$ the advantage narrows to 12 points because a large persistent shift degrades the fast predictor that supplies the evidence, leaving the controller's online updates as the remaining adaptation channel. Large persistent shifts that fundamentally alter the dynamics model's structure remain a limitation. The reduction in the error-growth rate (1.5-1.6$\times$) is empirical on two locomotion tasks. A formal theoretical analysis bounding this rate under specific model-class assumptions remains open.

The evidence supports a recipe rather than a specific architecture: a model which predicts at more than one rate already contains, at no future observation, an internal error signal that is defined where every observation-based signal is undefined — inside imagination. It is directional, so it can train the predictor that produced it, and it has a counterpart on realized transitions, so a controller can cancel what the planner avoided.

6 Conclusion

HiWM adds a second prediction rate to a latent MPC agent and measures the disagreement between the two rates at every imagined step. This cross-timescale disagreement is consumed at three clocks: as a consistency penalty during planning, as a correction signal during execution, and as a regularizer during training with the coupling between execution and training producing super-additive robustness. On 80 tasks under combined noise, the mechanism achieves the highest IQM among evaluated methods, with a margin that is noise-conditional, and the signal transfers zero-shot to physical hardware where disagreement is $2.4\times$ larger. In any model that predicts at more than one rate, cross-rate disagreement is an error signal available inside imagination.

AI USE STATEMENT

In this work, we used generative AI tools for grammar checking and text polishing. We have not used generative AI tools for any other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for no tasks with recommended disclosure. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, December 2024. doi: 10.52202/079017-1873. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html.

Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michel L. Littman. Combating the Compounding-Error Problem with a Multi-step Model, May 2019. URL http://arxiv.org/abs/1905.13320. arXiv:1905.13320 [cs].

Abdelhakim Benechehab, Giuseppe Paolo, Albert Thomas, Maurizio Filippone, and Balázs Kégl. Multi-timestep models for Model-based Reinforcement Learning, October 2023. URL http://arxiv.org/abs/2310.05672. arXiv:2310.05672 [cs].

Carlos Stein Brito and Daniel McNamee. World Models as Reference Trajectories for Rapid Motor Adaptation, May 2025. URL http://arxiv.org/abs/2505.15589. arXiv:2505.15589 [cs].

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite: A Contact-rich Simulation Suite for Musculoskeletal Motor Control. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, pp. 492–507. PMLR, May 2022. URL https://proceedings.mlr.press/v168/caggiano22a.html.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Robert Dadashi, Léonard Hussenot, Damien Vincent, Sertan Girgin, Anton Raichuk, Matthieu Geist, and Olivier Pietquin. Continuous Control with Action Quantization from Demonstrations, June 2022. URL http://arxiv.org/abs/2110.10149. arXiv:2110.10149 [cs].

Christian Gumbsch, Noor Sajid, Georg Martius, and Martin V. Butz. Learning Hierarchical World Models with Adaptive Temporal Abstractions from Discrete Latent Dynamics. October 2023. URL https://openreview.net/forum?id=TjCDNssXKU.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination, March 2020. URL http://arxiv.org/abs/1912.01603. arXiv:1912.01603 [cs].

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep Hierarchical Planning from Pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/a766f56d2da42cae20b5652970ec04ef-Abstract-Conference.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08744-2. URL https://www.nature.com/articles/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. October 2023. URL https://openreview.net/forum?id=Oxh5CstDJU.

Nicklas Hansen, Jyothir S, Vlad Sobal, Yann LeCun, Xiaolong Wang, and Hao Su. Hierarchical World Models as Visual Whole-Body Humanoid Controllers, May 2025. URL http://arxiv.org/abs/2405.18418. arXiv:2405.18418 [cs].

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*, pp. 8387–8406. PMLR, June 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Se Hwan Jeon, Ho Jae Lee, Seungwoo Hong, and Sangbae Kim. Residual mpc: Blending reinforcement learning with gpu-parallelized model predictive control, 2025. URL https://arxiv.org/abs/2510.12717.

Nan Jiang. A Note on Loss Functions and Error Compounding in Model-based Reinforcement Learning, April 2024. URL http://arxiv.org/abs/2404.09946. arXiv:2404.09946 [cs].

Nathan Lambert, Albert Wilcox, Howard Zhang, Kristofer S. J. Pister, and Roberto Calandra. Learning Accurate Long-term Dynamics for Model-based Reinforcement Learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2880–2887, December 2021. doi: 10.1109/CDC45484.2021.9683134. URL https://ieeexplore.ieee.org/abstract/document/9683134. ISSN: 2576-2370.

Nathan Lambert, Kristofer Pister, and Roberto Calandra. Investigating compounding prediction errors in learned dynamics models, 2022. URL https://arxiv.org/abs/2203.09637.

Haotian Lin, Pengcheng Wang, Jeff Schneider, and Guanya Shi. TD-M(PC)^2: Improving Temporal Difference MPC Through Policy Constraint, February 2025. URL http://arxiv.org/abs/2502.03550. arXiv:2502.03550 [cs].

Jianlan Luo, Perry Dong, Jeffrey Wu, Aviral Kumar, Xinyang Geng, and Sergey Levine. Action-Quantized Offline Reinforcement Learning for Robotic Skill Learning. In *Proceedings of The 7th Conference on Robot Learning*, pp. 1348–1361. PMLR, December 2023. URL https://proceedings.mlr.press/v229/luo23a.html.

Paul Mattes, Rainer Schlosser, and Ralf Herbrich. Hieros: Hierarchical Imagination on Structured State Space Sequence World Models, February 2024. URL http://arxiv.org/abs/2310.05167. arXiv:2310.05167 [cs].

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are Sample-Efficient World Models, September 2022. URL https://arxiv.org/abs/2209.00588v2.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust Adversarial Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2817–2826. PMLR, July 2017. URL https://proceedings.mlr.press/v70/pinto17a.html.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based World Models Are Happy With 100k Interactions, March 2023. URL https://arxiv.org/abs/2303.07109v1.

Aidan Scannell, Mohammadreza Nakhaeinezhadfard, Kalle Kujanpää, Yi Zhao, Kevin Sebastian Luck, Arno Solin, and Joni Pajarinen. Discrete Codebook World Models for Continuous Control. October 2024. URL https://openreview.net/forum?id=lfRYzd8ady.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to Explore via Self-Supervised World Models. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8583–8592. PMLR, November 2020. URL https://proceedings.mlr.press/v119/sekar20a.html.

E. Talvitie. Model regularization for stable sample rollouts. pp. 780–789, 2014. URL https://www.elibrary.ru/item.asp?id=31751639.

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, January 2018. URL http://arxiv.org/abs/1801.00690. arXiv:1801.00690 [cs].

Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model Predictive Path Integral Control: From Theory to Parallel Computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017. ISSN 0731-5090. doi: 10.2514/1.G001921. URL https://doi.org/10.2514/1.G001921. _eprint: https://doi.org/10.2514/1.G001921.

Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, December 2018. ISSN 1941-0468. doi: 10.1109/TRO.2018.2865891. URL https://ieeexplore.ieee.org/abstract/document/8558663.

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 *Proceedings of the Conference on Robot Learning*, pp. 1094–1100. PMLR, May 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. ISSN 2329-9304. doi: 10.1109/TASLP.2021.3129994. URL https://ieeexplore.ieee.org/abstract/document/9625818.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21024–21037. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/f0eb6568ea114ba6e293f903c34d7488-Abstract.html.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical Planning with Latent World Models, April 2026. URL http://arxiv.org/abs/2604.03208. arXiv:2604.03208 [cs].

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient Stochastic Transformer based World Models for Reinforcement Learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5647763d4245b23e6a1cb0a8947b38c9-Abstract-Conference.html.

A Algorithm and Implementation Details

A.1 Complete Training Algorithm

Algorithm 1 presents the full HiWM training procedure. The outer loop alternates between environment interaction (with online controller updates) and model learning from replay data. Three aspects deserve emphasis. (i) The imagination recursion (lines 14–20) runs the fast predictor g^f *inside* the MPPI loop, so that G_h varies across candidates and across horizons; no future observation is used. (ii) The controller updates at every environment step on a short-term buffer (line 32), while all other components update on batched replay data. (iii) The abstract model's objective includes λG as a regularizer (line 41), which pulls the abstract prior toward the fast-model-conditioned belief.

A.2 Online Controller Update Procedure

The controller C_ζ is a lightweight MLP ($2d_z + d_a \to 256 \to 256 \to d_a$, ReLU activations) updated at every environment step via a single gradient step on the 32 most recent transitions. The learning rate ($\eta_c = 3 \times 10^{-4}$) is set to match the abstract model's rate, because the controller must respond to changing execution conditions while the dynamics models learn more slowly for stability. Gradient clipping at norm 1.0 prevents large updates from noisy transitions. The fast model's weights ψ are frozen during controller updates, ensuring that the controller adapts its corrections without interfering with the learned dynamics. The action-alignment regularization ($\lambda_{\text{act}} = 0.01$) provides a soft bias toward the planned action without preventing necessary corrections. §D.2 confirms that correction magnitudes remain appropriately bounded.

A.3 Segment Action Space and VQ-VAE

Segment actions A_t encode temporally extended action spans of $H_A = 8$ primitive actions. We use a residual vector-quantization autoencoder with $N_q = 4$ residual stages, each with codebook size $K = 64$ and code dimension $d_c = 32$. The encoder maps a continuous action sequence $a_{t:t+H_A} \in \mathbb{R}^{H_A \times d_a}$ to a sequence of discrete codes via iterative residual quantization:

$$r_0 = V_{\text{enc}}(a_{t:t+H_A}), \quad c_j = \arg\min_k \|r_{j-1} - e_k^{(j)}\|, \quad r_j = r_{j-1} - e_{c_j}^{(j)}, \quad j = 1, \ldots, N_q \tag{5}$$

where $e_k^{(j)}$ is the k-th codebook entry at residual stage j. The segment action is the tuple $A_t = (c_1, c_2, c_3, c_4)$. The decoder reconstructs $\hat{a}_{t:t+H_A} = V_{\text{dec}}(\sum_j e_{c_j}^{(j)})$.

The VQ-VAE is trained jointly with the world model on action sequences sampled from the replay buffer, using reconstruction loss plus commitment loss ($\beta_{\text{commit}} = 0.25$). Exponential moving average codebook updates (decay 0.99) are used for stable training. Reconstruction error falls below 0.01 in normalized action space after 100k training steps.

A.4 Cross-Level Alignment

An alignment loss ensures the abstract and fast prediction levels maintain consistent representations. This loss is applied to the fast model only (asymmetric gradient flow):

$$\mathcal{L}_{\text{align}} = \lambda_{f\to s}\, \text{CE}\Big(h_{f\to s}(\hat{z}^f_{t+1}),\, \text{sg}(z^s_{t+1})\Big) + \lambda_{s\to f}\, \|h_{s\to f}(\text{sg}(z^s_{t+1})) - \hat{z}^f_{t+1}\|^2 \tag{6}$$

where $h_{f\to s}$ and $h_{s\to f}$ are two-layer MLPs (256 hidden units), $\lambda_{f\to s} = 1.0$, and $\lambda_{s\to f} = 0.1$. The asymmetric weighting ($\lambda_{f\to s} > \lambda_{s\to f}$) reflects a design priority: the fast model's predictions should be compatible with abstract-level categories (the $f{\to}s$ direction is more important for planning), while the reverse direction is a soft regularizer. Stop-gradients on z^s_{t+1} prevent the abstract model from trivially satisfying alignment by collapsing its discrete structure. The abstract model is trained solely by its own objective ($\mathcal{L}^s$); the fast model receives alignment gradients that encourage planning-compatible representations.

The alignment loss is an engineering component for stable training, not a core contribution. Removing it degrades performance by 3–5 IQM on representative tasks (Table 7).

Table 7: Effect of the cross-level alignment loss on representative tasks (5 seeds, combined noise).

Configuration	Dog Run	Assembly
Full model	0.84	0.83
No alignment ($\lambda_{\text{align}} = 0$)	0.80	0.79
Symmetric ($\lambda_{f\to s} = \lambda_{s\to f} = 0.5$)	0.82	0.80

A.5 Policy Learning and Exploration

A policy network π_ω is trained alongside the world model using TD3-style updates with an ensemble of 5 Q-functions. The policy serves as a warm-start prior for MPPI sampling ($N_\pi = 24$ of the 536 total candidates come from the policy). Exploration uses $Q_{\text{explore}} = Q + \lambda_{\text{explore}} \cdot G$ with $\lambda_{\text{explore}} = 0.5$. This bonus directs data collection toward states where the abstract model is revised most by fast-model evidence, which tends to coincide with states where the model improves most rapidly. As the model improves and disagreement decreases, the bonus naturally decays.

B Noise Protocol and Experimental Setup

B.1 Noise Protocol

The combined noise protocol is designed to simulate realistic deployment conditions where both sensing and actuation are imperfect.

Observation noise. Time-correlated perturbations via an Ornstein–Uhlenbeck (OU) process:

$$\tilde{o}_t = o_t + \xi_t, \qquad d\xi_t = -\theta_{\text{OU}}\, \xi_t\, dt + \sigma_{\text{obs}}\, dW_t \tag{7}$$

with $\theta_{\text{OU}} = 0.15$ and $\sigma_{\text{obs}} = 0.05$. The OU process produces temporally correlated noise that is more realistic than i.i.d. Gaussian noise: real sensors exhibit drift and autocorrelation. The stationary variance is $\sigma^2_{\text{obs}}/(2\theta_{\text{OU}}) \approx 0.0083$.

Action noise. The executed action is perturbed by systematic drift and random noise:

$$\tilde{a}_t = a_t + A_{\text{drift}} \sin(2\pi f_{\text{drift}}\, t) + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,\, \sigma^2_{\text{act}} I) \tag{8}$$

with $A_{\text{drift}} = 0.02$, $f_{\text{drift}} = 0.1$ Hz, and $\sigma_{\text{act}} = 0.05$. The sinusoidal component models systematic biases (e.g., actuator miscalibration that varies over time); the Gaussian component models random actuation errors.

Noise scaling. When scaling noise intensity σ (§4.3, §4.6 in the main text), both σ_{obs} and σ_{act} are scaled proportionally: $\sigma_{\text{obs}} = \sigma$, $\sigma_{\text{act}} = \sigma$, $A_{\text{drift}} = 0.4\sigma$.

Pixel-space noise (Franka Kitchen). For the image-based Franka Kitchen benchmark, observation noise is injected in pixel space as additive Gaussian noise ($\sigma_{\text{pixel}} = 8/255$) followed by random 3×3 Gaussian blur (probability 0.3). Action noise is identical to the state-based protocol.

B.2 Benchmark Details

DMControl (30 tasks). Tasks span five complexity levels based on observation and action dimensionality. *Low-dimensional* (5 tasks): Cartpole Swingup (obs 5, act 1), Reacher Easy (6, 2), Finger Spin (9, 2), Pendulum Swingup (3, 1), Ball-in-Cup Catch (8, 2). *Medium* (9 tasks): Walker Stand/Walk/Run (24, 6), Cheetah Run (17, 6), Hopper Stand/Hop (15, 4), Fish Swim (24, 5), Swimmer Swimmer6 (25, 5), Acrobot Swingup (6, 1). *High-dimensional locomotion* (10 tasks): Quadruped Walk/Run/Escape/Fetch (78, 12), Humanoid Stand/Walk/Run (67, 21), Dog Stand/Walk/Trot (223, 38). *High-dimensional complex* (6 tasks): Dog Run/Fetch (223, 38), Humanoid CMU Stand/Walk/Run (137, 56).

The Dog and Humanoid CMU tasks combine high-dimensional observations, high-dimensional action spaces, and the need for coordination over rollout depths of 120 primitive steps—the regime where compounding prediction errors are most damaging.

Meta-World (45 tasks). Robotic manipulation with a Sawyer arm. All tasks share observation space (39-dim) and action space (4-dim Cartesian end-effector velocity + gripper). Tasks are evaluated by success rate.

MyoSuite (5 tasks). Musculoskeletal control with inherent muscle redundancy and biological variability. Tasks: MyoHandReachFixed/Random, MyoHandPoseFixed/Random, MyoHandKeyTurnFixed. Observation dimensions range from 63 to 115; action dimensions are 39 (muscle activations).

Franka Kitchen (4 task sequences). Each task requires completing 4 subtasks in sequence (e.g., open microwave $\rightarrow$ move kettle $\rightarrow$ turn on light $\rightarrow$ slide cabinet). Image observations ($64\times64\times3$) are processed by a convolutional encoder (3 conv layers, 32/64/128 channels, kernel 3, stride 2, followed by a linear projection to d_z). Action space is 9-dimensional. Evaluation uses both average subtask completion (continuous) and full-task success rate (binary). 3 seeds, 50 evaluation episodes per seed.

B.3 Baseline Implementation

All baselines use official code repositories with default hyperparameters and receive identical noise-injected observations and actions.

- **TD-MPC2:** Official implementation. Default single-task configuration. Reproduction verified within 2% IQM under clean conditions.
- **DC-MPC:** Official implementation. Default FSQ levels matching the original paper.
- **DreamerV3:** Official implementation. State-based variant (no image encoder) for fair comparison with MPC-based methods.
- **Director:** Official implementation. Default subgoal horizon ($K = 8$) and goal-conditioned worker. All 80 tasks, 5 seeds.
- **TD-MPC:** Official implementation. Default hyperparameters.
- **Puppeteer:** Official implementation. Default two-level TD-MPC2 configuration. 12 representative tasks, 3 seeds.
- **HWM:** Official implementation adapted to state-based continuous control. 12 representative tasks, 3 seeds.

Evaluation uses the same protocol across all methods: IQM with 95% stratified bootstrap confidence intervals over 5 seeds. Environment steps and evaluation frequency are matched.

C Extended Ablation and Credit Assignment

C.1 Clean-Condition Performance

Table 8 reports aggregate performance without noise injection ($\sigma = 0$). Under clean conditions, HiWM matches or slightly outperforms baselines, with margins of 1–3 IQM. The feedback and correction mechanisms are designed for robustness under perturbation, and in clean environments the two-rate structure introduces modest overhead without proportionate benefit. The gap between HiWM and TD-MPC2 under clean conditions (+2 IQM on DMC) is within the range attributable to the modest parameter increase. The much larger gap under noise (+8 IQM on DMC at $\sigma = 0.05$; up to $+16.4$ at $\sigma = 0.2$) confirms that HiWM's advantage stems from its error-correction mechanisms, not from greater model capacity.

C.2 Noise Attribution Analysis

To disentangle the contributions of the feedback pathway and the corrective controller to robustness against specific noise sources, we evaluate under observation noise only and action noise only (Table 9).

Table 8: Aggregate performance under clean conditions ($\sigma = 0$). IQM scores (5 seeds, 95% CI).

Method	DMC IQM	MW IQM	Myo IQM
HiWM	0.87±.01	0.85±.01	0.74±.02
DC-MPC	0.86±.01	0.84±.02	0.73±.02
TD-MPC2	0.85±.01	0.83±.01	0.71±.02
Director	0.83±.02	0.80±.03	0.68±.03
DreamerV3	0.82±.02	0.78±.02	0.67±.02

Table 9: Ablation under isolated noise sources (Dog Run, $\sigma = 0.05$, 5 seeds).

Configuration	Obs. Noise Only	Act. Noise Only	Combined
Full model	0.86	0.85	0.84
No Feedback	0.72	0.80	0.68
No Correction	0.83	0.73	0.78
No Feedback + No Correction	0.60	0.62	0.56

Three patterns emerge. First, the feedback pathway is particularly effective against observation noise (removing it drops performance by 14 points under observation noise alone, vs. 5 points under action noise alone): observation noise corrupts the latent state estimates that the abstract model receives, and the pathway allows the fast predictor's short-horizon predictions to partially correct corrupted estimates before they compound. Second, the corrective controller is particularly effective against action noise (removing it drops performance by 12 points under action noise alone, vs. 3 points under observation noise): action noise directly causes execution deviations that the controller is designed to compensate. Third, the combined noise condition is more challenging than either alone, and the full model retains a larger advantage under combined noise.

C.3 Feedback Pathway Quality Metrics

To verify that the feedback pathway specifically improves the abstract model, we examine the quality of the abstract prior $p(z_t^s|d_t^s)$ and the fast-model-conditioned belief $q(z_t^s|d_t^s, z_t^f)$ under different configurations (Table 10).

Table 10: Abstract model quality metrics (Dog Run, at 2M training steps, 5 seeds).

Configuration	Prior CE	Belief CE	KL (Belief ‖ Prior)
Full model	0.407	0.215	0.192
No Feedback (top-down only)	0.595	0.589	0.006
No Correction	0.481	0.264	0.217

Without the feedback pathway ("No Feedback"), the belief collapses to the prior (KL drops to 0.006), confirming that the belief genuinely depends on fast-model evidence z_t^f in the full model. The prior itself improves by 31% in the full model relative to the no-feedback ablation (0.407 vs. 0.595), demonstrating that training with the pathway yields a better abstract dynamics model—not just a better belief.

Without the corrective controller ("No Correction"), the prior is moderately degraded (0.481 vs. 0.407), quantifying the execution-to-planning direction of the coupling: cleaner execution data from the controller produces a better abstract model.

C.4 Synergy Across Noise Levels

Table 11 extends the synergy analysis to multiple noise levels on Dog Run, confirming that $S_{F\times C} > 0$ holds across all tested conditions.

The synergy index increases monotonically with noise intensity. Under stronger perturbations, each mechanism has more error to correct, and the corrections from one reduce the burden on the other.

Table 11: Synergy index $S_{F\times C}$ across noise levels (Dog Run, 5 seeds). Δ denotes performance drop relative to the full model.

σ	Δ(No FB)	Δ(No Corr)	Δ(Neither)	$S_{F\times C}$
0.025	0.08	0.03	0.13	0.024
0.05	0.16	0.06	0.28	0.071
0.10	0.24	0.09	0.42	0.107
0.15	0.30	0.12	0.55	0.155
0.20	0.35	0.14	0.66	0.202

A perfect monotone ordering of five levels has exact one-sided probability $p \approx 0.008$; a seed artifact does not order itself by noise level.

C.5 Component Isolation

Table 12 supplements Table 2 in the main text by providing the component-isolation results on Dog Run and Assembly. These variants isolate timescale and action abstraction from the two-rate structure.

Table 12: Component isolation (Dog Run and Assembly, 5 seeds, combined noise).

Configuration	Dog Run	Assembly
HiWM (full)	0.84±.02	0.83±.02
HiWM-primAction (no segment VQ-VAE)	0.80±.02	0.78±.02
HiWM-1T (single timescale)	0.76±.02	0.71±.02
Flat baseline (TD-MPC2 arch.)	0.73±.02	0.68±.01

Action abstraction (-4 IQM) is enabling component with modest contributions. The two-rate structure with feedback (the jump from HiWM-1T to the full model, $+8$/$+12$ points) provides the largest single-component contribution. These components are not claimed as contributions.

D Controller and Planning Analysis

D.1 Controller on a Flat Backbone

Section 4.3 reports that bolting an identical corrective controller onto TD-MPC2 recovers 36–40% of HiWM's advantage. Table 13 provides the execution-error and correction-magnitude statistics that explain the remaining gap.

Table 13: Execution error and correction statistics (Dog Run, combined noise, 5 seeds).

Configuration	Exec. Error	Correction Magnitude
TD-MPC2 (no controller)	0.091	—
TD-MPC2 + Correction	0.058	0.038
HiWM (full)	0.039	0.030

The flat controller operates with 49% higher residual error (0.058 vs. 0.039) and requires 27% larger corrections (0.038 vs. 0.030). Three mechanisms absent on the flat backbone explain the gap: (a) the flat controller's prediction target is a single-timescale latent prediction without abstract-state conditioning, yielding less accurate reference trajectories; (b) the reference actions are not steered by the consistency penalty toward well-modeled regions, so the controller must compensate for both noise and planning errors; (c) no training-time prior improvement occurs because G does not exist.

The controller on a flat backbone matches HiWM's controller architecture, loss function, buffer size, and learning rate. TD-MPC2's model weights are frozen during controller updates.

D.2 Controller Behavior Statistics

Table 14 characterizes the corrective controller's behavior on Dog Run, verifying that it operates as a fine-grained corrective mechanism rather than a dominant override.

Table 14: Controller correction statistics (Dog Run, $\sigma = 0.05$, 5 seeds).

Metric	Value
Mean correction magnitude $\|\delta a_t\|$	0.030
Mean reference action magnitude $\|\hat{a}_t\|$	0.412
Correction-to-reference ratio	7.3%
Max correction magnitude (99th percentile)	0.089
Correlation $r(\|\delta a_t\|,$ prediction error)	0.73

The correction magnitude is 7.3% of the reference action, confirming small, targeted adjustments. The strong correlation (0.73) between correction magnitude and prediction error confirms that corrections are driven by the intended signal. At the 99th percentile, corrections reach 0.089 (21.6% of the reference), showing the controller scales up appropriately under large prediction errors without losing stability.

D.3 Planning Penalty Analysis

Table 15 compares MPPI trajectory statistics with and without the consistency penalty.

Table 15: MPPI trajectory statistics (Dog Run, $\sigma = 0.05$, averaged over 1000 planning steps).

Metric	With Penalty ($\rho = 0.1$)	Without Penalty ($\rho = 0$)
Mean trajectory reward	4.21	4.53
Mean trajectory G	0.14	0.37
Execution error (one-step)	0.039	0.066
Episode return	0.84	0.80

The penalty-free planner selects trajectories with higher predicted reward (4.53 vs. 4.21) but also higher disagreement (0.37 vs. 0.14). These plans pass through poorly modeled regions, resulting in larger execution errors (0.066 vs. 0.039) and lower episode returns (0.80 vs. 0.84). The penalty effectively trades short-horizon predicted reward for agreement between the two prediction rates—plans are slightly less ambitious but substantially more executable.

D.4 Transient Perturbation Recovery

Table 16 reports recovery times under sudden perturbations of varying magnitude.

Table 16: Recovery time (steps to return to pre-perturbation error level) after a sudden perturbation of magnitude δ. Combined noise, 5 seeds.

Task	δ	HiWM	No Correction	TD-MPC2
Dog Run	0.1	8	52	25
Dog Run	0.2	14	88	42
Dog Run	0.3	22	130+	68
Humanoid Walk	0.2	16	95	48
Assembly	0.2	14	90	42
Hand Pose	0.2	18	98	50

HiWM consistently recovers 3–4$\times$ faster than TD-MPC2 and 5–6$\times$ faster than the ablation without correction. At $\delta = 0.3$ on Dog Run, the no-correction variant fails to fully recover within the episode (130+ steps), while HiWM recovers in 22 steps. Post-recovery error returns to pre-perturbation levels in the full model, while the no-correction variant shows persistent elevation.

E HIERARCHICAL BASELINES EXTENDED (E3)

E.1 PER-TASK RESULTS ON DMC-HARD

Table 17 provides per-task comparisons on DMC-Hard tasks under combined noise (3 seeds, 95% bootstrap CI).

Table 17: Per-task IQM on DMC-Hard under combined noise (3 seeds, 95% CI). Puppeteer and HWM use official implementations.

Task	HiWM	Puppeteer	HWM	Director	TD-MPC2
Dog Run	0.84±.02	0.73±.03	0.71±.03	0.69±.04	0.73±.02
Dog Walk	0.86±.02	0.76±.02	0.74±.03	0.72±.03	0.78±.02
Dog Trot	0.83±.02	0.74±.02	0.72±.03	0.70±.03	0.76±.02
Humanoid Walk	0.81±.02	0.73±.02	0.70±.03	0.68±.03	0.74±.02
Humanoid Run	0.74±.02	0.65±.03	0.63±.03	0.61±.03	0.66±.02
Quad. Walk	0.85±.02	0.78±.02	0.76±.02	0.74±.03	0.79±.02
Quad. Run	0.80±.02	0.72±.02	0.70±.03	0.68±.03	0.73±.02

HiWM's advantage over Puppeteer is largest on the Dog tasks (+10–13 IQM), where the 223-dimensional observation space and 38-dimensional action space make compounding errors most damaging. On Quadruped tasks, the gap is smaller (+7–8). Puppeteer matches or slightly exceeds TD-MPC2 on low-dimensional tasks but provides no advantage under noise on high-dimensional tasks, indicating that top-down hierarchy without upward flow does not mitigate model error.

E.2 CLEAN-CONDITION COMPARISON

Table 18 confirms that the performance gaps are noise-conditional, not capacity-driven.

Table 18: Clean-condition comparison ($\sigma = 0$) on representative tasks (3 seeds, 95% CI).

Task	HiWM	Puppeteer	HWM	TD-MPC2
Dog Run	0.87±.01	0.85±.02	0.84±.02	0.85±.01
Humanoid Walk	0.84±.01	0.82±.02	0.81±.02	0.83±.01
Assembly	0.86±.01	0.84±.02	0.83±.02	0.83±.01
Hand Pose	0.76±.02	0.74±.02	0.73±.02	0.74±.02

At $\sigma = 0$, all hierarchical methods perform within 2–3 IQM of each other and of TD-MPC2, confirming that HiWM's advantage under noise stems from its error-correction mechanisms.

F COUPLING SIGNAL ANALYSIS EXTENDED

F.1 RANK CORRELATIONS BY HORIZON

Table 19 reports the full set of Spearman rank correlations between imagined and realized G at each measured horizon.

The decay is expected: the fast model's own prediction error accumulates over longer horizons, degrading the quality of the evidence while preserving relative ordering. The consistency penalty still works because MPPI needs a correct *ordering* of candidates, not a correct magnitude. The decile analysis confirms the ordering survives: the top decile of imagined G corresponds to 2.8× (Dog Run) / 2.5× (Assembly) higher realized G than the bottom decile.

F.2 STALE-EVIDENCE ABLATION

In the stale-evidence ablation, the MPPI loop is modified so that $\hat{z}^f_{t+h} = \hat{z}^f_t$ for all $h \geq 1$: the fast model is not rolled forward, and the same frozen evidence is reused at every horizon step. This removes the forward predictive component while retaining the conditioning structure. It is the

Table 19: Spearman rank correlation between imagined and realized G at each rollout horizon (1000 planning steps from a converged model).

Horizon h	Dog Run ρ	Assembly ρ
1	0.78	0.74
3	0.65	0.61
5	0.53	0.49
8	0.41	0.37
10	0.34	0.30
15	0.23	0.20

configuration closest to a standard posterior, where the only observation-grounded information is the current observation.

When evidence is frozen, G becomes nearly constant across the rollout (it varies only through the recurrent state $\hat{d}^s_{t+h}$), and the consistency penalty loses its ability to discriminate between trajectories at different horizon depths. The 8-point degradation on Dog Run ($0.84 \rightarrow 0.76$) represents roughly half the total feedback benefit, confirming that the forward rollout is where the informative variation resides.

G Stop-Gradient Analysis

The stop-gradient placements in HiWM's training objectives are critical for stable training and an informative cross-timescale disagreement signal. We provide the justification for each and confirm via ablation.

Stop-gradient on the belief in $\mathcal{L}^s_{\mathrm{KL}}$.

$$G(z_t) = D_{\mathrm{KL}}\Big[\,\mathrm{sg}(q(z^s_t \mid d^s_t,\, z^f_t)) \parallel p(z^s_t \mid d^s_t)\Big] \tag{9}$$

Without sg, the belief could minimize the KL by becoming less informative (ignoring z^f_t), which would destroy the feedback pathway. The stop-gradient ensures that only the prior is updated toward the belief, maintaining an informative disagreement signal.

Stop-gradient on targets in dynamics losses.

$$\mathcal{L}^s_{\mathrm{dyn}} = \mathrm{CE}(p_\varphi(\hat{z}^s_{h+1}),\, \mathrm{sg}(z^s_{h+1})), \qquad \mathcal{L}^f_{\mathrm{dyn}} = \|\hat{z}^f_{t+1} - \mathrm{sg}(z^f_{t+1})\|^2 \tag{10}$$

These prevent the encoder from changing its representation to trivially satisfy prediction targets, maintaining a stable latent space.

Stop-gradient on z^s_{t+1} in the alignment loss.

$$\mathcal{L}_{f\to s} = \mathrm{CE}(h_{f\to s}(\hat{z}^f_{t+1}),\, \mathrm{sg}(z^s_{t+1})) \tag{11}$$

This prevents the abstract model from adapting to satisfy alignment. The abstract model trains only via $\mathcal{L}^s$; the fast model receives alignment gradients.

Table 20 confirms the importance of each placement.

Table 20: Effect of removing stop-gradients (Dog Run, $\sigma = 0.05$, 5 seeds).

Configuration	IQM	Prior CE	Training Stability
Full (all sg)	0.84	0.407	Stable
No sg on belief in $\mathcal{L}^s_{\mathrm{KL}}$	0.71	0.82	KL collapse at $\sim$500k steps
No sg on dynamics targets	0.63	1.12	Representation drift
No sg on alignment targets	0.79	0.44	Mild instability after 1.5M steps

H HYPERPARAMETER SENSITIVITY

All two key hyperparameters exhibit broad plateaus.

H.1 CONSISTENCY PENALTY ρ

Table 21: Sensitivity to consistency penalty weight (Dog Run, $\sigma = 0.05$, 5 seeds).

ρ	IQM	Exec. Error	Traj. Reward
0.00	0.80	0.066	4.53
0.05	0.82	0.048	4.38
0.10	0.84	0.039	4.21
0.15	0.83	0.035	4.02
0.20	0.82	0.033	3.85
0.50	0.76	0.028	3.12

The plateau spans $\rho \in [0.05, 0.20]$. Below 0.05, the penalty is insufficient to steer plans away from high-disagreement regions. Above 0.20, the planner becomes overly conservative, avoiding ambitious trajectories even when the model is reasonably accurate.

H.2 KL COEFFICIENT λ

Table 22: Sensitivity to KL coefficient (Dog Run, $\sigma = 0.05$, 5 seeds).

λ	IQM	KL (Belief $\|\|$ Prior)	Note
0.1	0.80	0.05	Under-explores
0.3	0.82	0.12	—
0.5	0.84	0.19	—
0.7	0.83	0.28	—
1.0	0.81	0.41	Over-explores

The plateau spans $\lambda \in [0.3, 0.7]$. At $\lambda = 0.1$, the prior closely tracks the belief, producing a weak disagreement signal. At $\lambda = 1.0$, the signal is too strong, causing excessive exploration at the expense of exploitation.

I DYNAMICS SHIFT ANALYSIS

Models are trained under the standard noise protocol with no dynamics shifts, then evaluated zero-shot under parametric dynamics changes. Table 23 provides the full performance-retention results across all tested shift types. Table 24 reports the corresponding mean G values, confirming a monotonic relationship between shift magnitude and cross-timescale disagreement.

Table 23: Performance retention under parametric dynamics shifts (Dog Run, fraction of nominal IQM, 3 seeds, 50 episodes per condition per seed).

Shift	HiWM	TD-MPC2+C	TD-MPC2
Mass $\times 0.7$	92%	83%	78%
Mass $\times 1.3$	90%	81%	76%
Friction $\times 0.5$	86%	74%	68%
Friction $\times 1.5$	89%	78%	73%
Gravity $\times 0.8$	93%	84%	80%
Gravity $\times 1.2$	88%	77%	72%
Actuator gain $\times 0.8$	91%	82%	76%
Actuator delay 2 steps	82%	70%	62%

Table 24: Mean G under dynamics shifts (Dog Run).

Condition	Mean G
Nominal (no shift)	0.14
Mass $\times 0.85$	0.18
Mass $\times 0.7$	0.24
Mass $\times 1.15$	0.19
Mass $\times 1.3$	0.26
Friction $\times 0.75$	0.21
Friction $\times 0.5$	0.32
Actuator delay 1 step	0.22
Actuator delay 2 steps	0.35

The largest advantage appears under actuator delay (+20 pp retention over TD-MPC2) and friction reduction (+18 pp), representing the most severe model-mismatch conditions. At friction $\times 0.5$, HiWM's retention drops to 86% and the advantage over TD-MPC2+C narrows to 12 pp because the fast predictor itself degrades under extreme persistent shifts, leaving the controller's online updates as the primary adaptation channel.

J Real-Robot Deployment Extended

J.1 Hardware Protocol

The deployment study uses a 7-DoF Franka Emika Panda robot arm with a wrist-mounted force/torque sensor and Intel RealSense D435 for state estimation. Control frequency: 20 Hz (50 ms planning budget, compatible with the 14 ms measured planning latency plus communication overhead). Observations are image-based: 14-dim (7 joint positions + 7 joint velocities) plus visual images plus task-specific object state.

Three tasks of increasing complexity were evaluated: (a) *Peg Insertion* (single-stage, 10 cm peg into 1 mm-clearance hole, $\sim$50 steps); (b) *Pick-and-Place* (two-stage: grasp $\rightarrow$ transport $\rightarrow$ place, $\sim$80 steps); (c) *Stack Two Blocks* (three-stage: grasp block 1 $\rightarrow$ place $\rightarrow$ grasp block 2 $\rightarrow$ stack, $\sim$120 steps).

All methods were trained in matched MuJoCo simulation under the same noise protocol and deployed zero-shot (no fine-tuning on hardware). 20 trials per method per task, with randomized initial object positions. Two policy seeds per method, alternated across trials. Wilson confidence intervals are used (appropriate for binomial outcomes with $n = 20$).

J.2 Mechanism Measurements

Table 25 compares mechanism measurements on hardware versus simulation for Peg Insertion (averaged over successful trials).

Table 25: Mechanism measurements: hardware vs. simulation (Peg Insertion).

Metric	Hardware	Simulation	Ratio
Mean G	0.34	0.14	2.4$\times$
Mean correction $\|\delta a\|$	0.052	0.030	1.7$\times$
Mean exec. error	0.068	0.039	1.7$\times$
Planning latency	14 ms	16 ms	0.9$\times$

The correction-to-error ratio is stable across deployment conditions ($0.052/0.068 \approx 0.030/0.039$), confirming the controller adapts its intervention magnitude to actual model mismatch without overcompensation.

K Compositional Tasks Extended

The Franka Kitchen experiments use 4 task sequences, each requiring 4 sequential subtasks. The observation space is image-based ($64\times64\times3$, processed by a convolutional encoder; see §B.2); the action space is 9-dimensional. Three seeds per method, 50 evaluation episodes per seed, identical combined noise protocol with pixel-space noise injection.

The contrast between average subtask completion ($+0.8$ subtasks over TD-MPC2) and full-task success ($+23$ pp) highlights the compounding nature of errors in compositional tasks: an error that merely degrades the current subtask prevents the next one from being attempted. HiWM without feedback (18%) falls below all flat baselines (TD-MPC2: 25%), confirming that hierarchy without upward flow is harmful for compositional tasks—the same pattern observed in single-stage tasks and in the hierarchical baselines experiment (§E).

L Long-Horizon Prediction Error

Section 4.5 reports the compounding-error reduction on Dog Run. Table 26 provides the corresponding analysis for Humanoid Walk, confirming that the reduction generalizes across high-dimensional locomotion tasks.

Table 26: Abstract prediction error (cross-entropy) vs. rollout horizon (Humanoid Walk, 5 seeds).

Horizon h	Full Model	No Feedback	Ratio
1	0.101	0.134	1.3×
5	0.289	0.510	1.8×
10	0.512	1.284	2.5×
15	0.738	2.342	3.2×
20	0.971	3.580	3.7×

The exponential growth rate is 0.103 for the full model and 0.164 for the model without feedback, giving a ratio of 1.6×. This is consistent with the Dog Run analysis (1.5×) and provides additional evidence that the feedback pathway yields a systematic reduction in the compounding-error rate. The gap widens at long horizons (3.7× at $h = 20$), providing direct evidence that the mechanism specifically mitigates *compounding* errors rather than merely improving initial prediction quality.

M Per-Task Performance on DMControl

Table 27 reports per-task IQM under combined noise for all 30 DMControl tasks, grouped by difficulty. HiWM achieves the best performance on 26 of 30 tasks. On the 4 tasks where another method matches or marginally exceeds HiWM (by 1–2 IQM within CIs), all share a common profile: low-dimensional observation spaces (obs < 10), short effective planning horizons, and low sensitivity to execution noise.

HiWM's advantage is largest on the high-dimensional tasks (Dog: $+8$–14 over TD-MPC2; Humanoid: $+7$–8), where coordination over rollout depths of 120 primitive steps makes compounding errors most damaging. On easy tasks with low-dimensional state spaces, the advantage is smaller (1–3 IQM), consistent with the feedback and correction mechanisms providing diminishing returns when prediction errors are already small.

N Sample Efficiency

Table 28 reports the number of environment steps required to reach 80% of asymptotic performance under combined noise.

HiWM reaches 80% of its asymptotic performance 1.4–1.6× faster than TD-MPC2 on the two representative tasks. Two factors contribute: (i) the exploration bonus directs data collection toward states where the abstract model is revised most by fast-model evidence, and (ii) the corrective controller

Table 27: Per-task IQM on DMControl under combined noise (5 seeds, 95% bootstrap CI).

Task	HiWM	TD-MPC2	DC-MPC	DreamerV3
Easy (obs $\leq$ 12)				
Cartpole Swingup	0.94±.01	0.91±.01	0.93±.01	0.90±.02
Reacher Easy	0.96±.01	0.94±.01	0.95±.01	0.92±.01
Finger Spin	0.93±.01	0.90±.02	0.91±.01	0.88±.02
Pendulum Swingup	0.95±.01	0.93±.01	0.94±.01	0.91±.01
Ball-in-Cup Catch	0.97±.01	0.96±.01	0.96±.01	0.94±.01
Medium (obs 12–50)				
Walker Stand	0.92±.01	0.88±.01	0.90±.01	0.85±.02
Walker Walk	0.89±.01	0.84±.01	0.86±.02	0.80±.02
Walker Run	0.82±.02	0.76±.02	0.78±.02	0.72±.03
Cheetah Run	0.85±.01	0.80±.01	0.82±.02	0.77±.02
Hopper Stand	0.90±.01	0.86±.01	0.87±.01	0.83±.02
Hopper Hop	0.78±.02	0.72±.02	0.74±.02	0.68±.03
Fish Swim	0.81±.02	0.75±.02	0.77±.02	0.71±.02
Swimmer Swimmer6	0.76±.02	0.70±.02	0.72±.03	0.66±.03
Acrobot Swingup	0.74±.02	0.68±.02	0.70±.03	0.64±.03
Hard (obs $>$ 50)				
Quadruped Walk	0.86±.01	0.79±.02	0.81±.02	0.74±.02
Quadruped Run	0.80±.02	0.73±.02	0.75±.02	0.68±.03
Humanoid Stand	0.85±.01	0.78±.01	0.80±.02	0.73±.02
Humanoid Walk	0.81±.02	0.74±.02	0.76±.02	0.69±.02
Humanoid Run	0.74±.02	0.66±.02	0.69±.02	0.62±.03
Dog Stand	0.88±.01	0.80±.02	0.83±.01	0.76±.02
Dog Walk	0.86±.01	0.78±.02	0.81±.02	0.74±.02
Dog Trot	0.84±.01	0.76±.02	0.79±.02	0.72±.02
Dog Run	0.84±.01	0.73±.02	0.76±.02	0.70±.03
Dog Fetch	0.79±.02	0.70±.02	0.73±.02	0.66±.03

Table 28: Steps to reach 80% of asymptotic performance (combined noise, 5 seeds).

Method	Dog Run	Assembly
HiWM	350k	280k
TD-MPC2	500k	450k
DC-MPC	480k	420k
DreamerV3	600k	550k
Director	650k	580k

provides immediate exploitation of partial knowledge (online correction of model errors) while the model continues to improve.

O Cross-Task Analysis

O.1 Task Properties Predicting HiWM's Advantage

Table 29 reports the correlation between task properties and the performance gap (HiWM − TD-MPC2) across all 80 tasks.

Table 29: Correlation between task properties and HiWM's advantage over TD-MPC2.

Task Property	Pearson r	Interpretation
Effective planning horizon	0.71	Strongest predictor
Observation dimensionality	0.62	High-dim tasks benefit more
Action dimensionality	0.54	High-dim actions benefit from correction
Contact richness (binary)	0.41	Contact tasks benefit moderately

The strongest predictor is effective planning horizon ($r = 0.71$): tasks requiring coordination over many steps benefit most from the feedback pathway, which reduces compounding errors over extended rollouts. Observation dimensionality ($r = 0.62$) is the second-strongest predictor, consistent with high-dimensional latent spaces amplifying prediction errors.

O.2 Robustness to Noise Intensity

HiWM's advantage over the best baseline grows monotonically with noise intensity across all three representative tasks tested. At $\sigma = 0$, the gap is 1–3 IQM; at $\sigma = 0.2$, it reaches 16–23 points. HiWM retains $\geq$84% of noise-free performance at $\sigma = 0.2$, while baselines retain only 57–68%.

O.3 Generalization to Unseen Noise Distributions

HiWM maintains its advantage under noise distributions not seen during training: uniform observation noise (95% vs. 88% of training-noise performance on Dog Run), impulse action noise (91% vs. 72%), and adversarial noise (81% vs. 63%). The largest gap appears under impulse noise, where single large deviations cascade without online correction. The mechanism generalizes because it operates on prediction error—a domain-general signal not tuned to specific noise distributions.

P Computational Cost

Table 30: Computational cost comparison. All methods on a single A100 GPU.

Method	Train (days)	FPS	Plan (ms)	Overhead
HiWM	8–10	82	16	1.4×
TD-MPC2	5–7	120	12	1.0×
DC-MPC	6–8	100	14	1.1×
DreamerV3	8–12	60	2	1.5×

HiWM's overhead is 1.4× training time and 16 ms planning latency (within the 50 Hz control budget used in simulation and the 20 Hz budget used on hardware). The overhead arises from the fast-model forward pass inside each MPPI candidate evaluation. The 5-member ensemble UQ variant (E4) requires 5× fast-model forward passes per MPPI step, making it more expensive than HiWM for a comparable planning penalty.

Q Broader Impact

This work addresses the robustness of model-based reinforcement learning under noisy conditions. The primary societal benefit is improved reliability of learned controllers for robotic systems, where sensor noise and actuator imprecision are unavoidable. More robust planning and execution could reduce failure rates in applications such as surgical robotics, autonomous vehicles, and industrial automation.

We do not foresee direct negative societal impacts from this work. The method does not introduce new capabilities for autonomous systems; it improves the reliability of existing planning-based approaches. As with all advances in RL, improved control algorithms could be applied to dual-use systems. We believe the benefits of more reliable robotic control substantially outweigh these risks and encourage responsible deployment practices.

Our experiments use standard simulation benchmarks (DMControl, Meta-World, MyoSuite, Franka Kitchen) and one hardware platform (Franka Panda). They do not involve human subjects, private data, or environmentally costly compute (all simulation experiments run on a single A100 GPU for 8–10 hours per task; hardware experiments used a total of $\sim$360 trials across three tasks).

Algorithm 1 HiWM Training

Initialize: encoder E_θ, abstract model f^s_φ with prior p and belief q, fast model g^f_ψ, reward predictor R_ϕ, termination predictor D^s, Q-ensemble $\{Q^i_\xi\}_{i=1}^5$, policy π_ω, controller C_ζ, VQ-VAE (V_{enc}, V_{dec}, codebook), replay buffer $\mathcal{D}$, controller buffer $\mathcal{D}_c$ (capacity 32)
for each training episode **do**
  $o_0 \leftarrow$ `env.reset()`
  $z^f_0 \leftarrow E_\theta(o_0), \quad z^s_0 \leftarrow \mathrm{FSQ}(z^f_0)$
  Initialize recurrent state d^s_0
  for $t = 0, 1, \ldots, T$ **do**
    — Planning —
    Sample N_p=512 candidate segment sequences from $\mathcal{N}(\mu, \sigma^2)$
    Sample N_π=24 sequences from policy prior π_ω
    for $j = 1, \ldots, J$=6 MPPI iterations **do**
      for each candidate sequence $(A_{t:t+H_s})$ **do**
        Decode reference actions: $\hat{a}_{t:t+H_s \cdot H_A} = V_{\text{dec}}(A_{t:t+H_s})$
        Initialize: $\hat{z}^f_t \leftarrow z^f_t$, $\hat{d}^s_t \leftarrow d^s_t$
        for $h = 0, \ldots, H_s$ **do**
          $\hat{d}^s_{t+h} = f^s(\hat{d}^s_{t+h-1},\, \hat{z}^s_{t+h-1},\, A_{t+h-1})$ ▷ Abstract dynamics
          $\hat{z}^s_{t+h} \sim p(z^s \mid \hat{d}^s_{t+h})$ ▷ Abstract prior
          $\hat{z}^f_{t+h} = g^f(\hat{z}^f_{t+h-1},\, \hat{a}_{t+h-1},\, \hat{z}^s_{t+h})$ ▷ Fast predictor
          $G_h = D_{\text{KL}}\big[\,\mathrm{sg}\, q(z^s \mid \hat{d}^s_{t+h},\, \hat{z}^f_{t+h}) \,\|\, p(z^s \mid \hat{d}^s_{t+h})\big]$ ▷ Cross-timescale disagreement
          $J_h = \gamma^h \big[R_\phi(\hat{z}^s_{t+h}, A_{t+h}) - \rho\, G_h\big]$ ▷ Penalized reward
        end for
        Total: $J = \sum_h J_h + \gamma^{H_s} Q(\hat{z}^s_{t+H_s}, A_{t+H_s})$
      end for
      Select K=64 elites; update μ, σ^2 with temperature τ=0.5
    end for
    Select A^*_t; decode reference actions: $\hat{a}_{t:t+H_A} = V_{\text{dec}}(A^*_t)$
    — Execution with correction —
    Predict next fast state: $\hat{z}^f_{t+1} = g^f(z^f_t,\, \hat{a}_t,\, z^s_{t+1})$
    Compute corrective action: $\delta a_t = C_\zeta(z^f_t,\, \hat{z}^f_t,\, \hat{a}_t)$
    Execute: $a_t = \hat{a}_t + \delta a_t$
    Observe: $o_{t+1}, r_t,$ `done`
    — Online controller update —
    Add $(z^f_t, \hat{z}^f_t, \hat{a}_t, a_t, z^f_{t+1}, \hat{z}^f_{t+1})$ to $\mathcal{D}_c$
    Update C_ζ on $\mathcal{D}_c$: $\mathcal{L}_c = \|\hat{z}^f_{t+1}(a_t) - z^f_{t+1}\|^2 + \lambda_{\text{act}} \|a_t - \hat{a}_t\|^2$ (g^f_ψ frozen)
    — Encode new state —
    $z^f_{t+1} \leftarrow E_\theta(o_{t+1}),\; z^s_{t+1} \leftarrow \mathrm{FSQ}(z^f_{t+1})$
    $d^s_{t+1} = f^s(d^s_t,\, z^s_t,\, A_t)$
    Add transition to $\mathcal{D}$
  end for
  — Model learning (per gradient step) —
  Sample batch $\mathcal{B}$ from $\mathcal{D}$
  1. Update abstract model φ:
  $\mathcal{L}^s = \sum_{h=0}^{H_s} \gamma^h \big[\mathrm{CE}(p(\hat{z}^s_{h+1}), \mathrm{sg}(z^s_{h+1})) + \|R_\phi - r\|^2 + \mathrm{BCE}(D^s, \delta) + \lambda\, G\big]$
  2. Update fast model ψ:
  $\mathcal{L}^f = \sum_{t=0}^{H_f} \gamma^t \Big[\|\hat{z}^f_{t+1} - \mathrm{sg}(z^f_{t+1})\|^2 + \lambda_{\text{align}}\, \mathcal{L}_{\text{align}}\Big]$
  3. Update Q-ensemble ξ via N-step TD targets
  4. Update policy ω (every 2 steps):
  $\mathcal{L}_\pi = -\mathbb{E}[Q(z^s, \pi_\omega(z^s)) + \lambda_{\text{explore}}\, G]$
  5. Update VQ-VAE on sampled action sequences
end for