FILTERING FROZEN WORLD MODELS: COVARIANCE-AWARE SURPRISE AND BETTER FORECASTS AT TEST TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Trained world models are usually run as fixed maps: each observation is encoded at face value and rolled forward, so sensor noise passes straight into the latent state. We show that classical state estimation recovers much of the lost accuracy without retraining. KalWM (Kalman World Model) applies an extended Kalman filter to the latent state of a frozen predictor, leaving every weight unchanged, and provides two outputs: a posterior that seeds the model's own rollout, and a covariance-weighted innovation that scores unexpected observations. Ours results shows that on released checkpoints of a trained visual world model for planning tasks, filtering the same three-frame history reduces six-step prediction error by 2.2%, 31.0%, and 1.5% under Gaussian image noise of standard deviation 0.1, and covariance-weighted scoring raises sensor-change detection AUROC. Ablations with fixed innovations show that these two benefits have distinct sources: filtering improves the prediction stream, while covariance weighting improves how its errors are ranked.

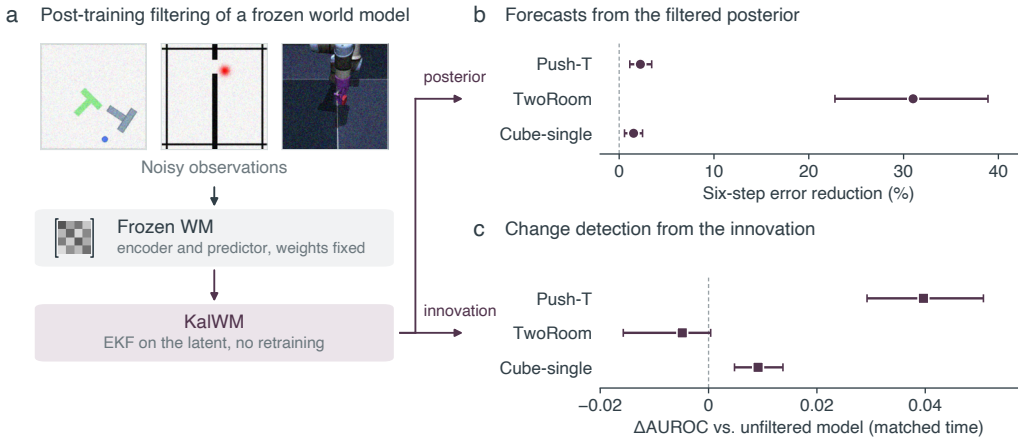

Figure 1: **Filtering a frozen world model, and its two separate effects. a**, KalWM applies an EKF to the latent of a frozen world model, drawn as a fixed weight matrix. Images are actual noisy benchmark frames. The experiments use released checkpoints of a trained world model. **b**, The posterior seeds the model's native rollout: relative reduction in six-step prediction error at image-noise sd 0.1. **c**, The innovation scores changes: matched-time AUROC difference between KalWM's Gaussian score and the unfiltered model for a noise increase from 0.03 to 0.1. Positive values favor KalWM; bars are paired 95% episode-bootstrap intervals. The TwoRoom detection interval includes zero, and clean one-step predictions worsen on all three tasks (Table 1).

1 INTRODUCTION

Latent world models learn how observations evolve under actions and serve as forward models for planning (Hafner et al., 2019; LeCun, 2022; Zhou et al., 2025; Maes et al., 2026). Once trained, such

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.

a model is typically used as a fixed map: each new frame is encoded, taken at face value, and rolled forward. When the sensor degrades, the encoder passes that noise directly into the latent state, and the predictor carries it into every forecast.

Classical state estimation offers a solution that requires no retraining. Treat the latent as a hidden state, and infer it by combining the model's own prediction with each new measurement. An extended Kalman filter (EKF) does exactly this around a frozen encoder and predictor. We call this post-training method KalWM. It leaves every network weight unchanged and alters only inference. The filter then provides two things at once: a posterior that can replace the raw encoding as the starting point for forecasts, and an innovation, weighted by its predicted covariance, that scores how unexpected each observation is.

The benefits are distinct. Filtering changes the prediction error itself, whereas covariance weighting only changes which errors are ranked as surprising. A filter can improve forecasts without improving detection, and a detector can improve through better state estimates rather than through its covariance. The covariance can even hurt, by discounting a meaningful change along a direction it treats as noisy. Judging KalWM therefore requires two comparisons: against the frozen model it wraps, and against itself with the innovations held fixed and only the score varied. We organize the study around three questions that follow this separation:

1. **RQ1: Does filtering improve a frozen visual world model?** We compare a trained world model with the same checkpoint plus KalWM on prediction and sensor-change detection in Push-T, TwoRoom, and OGBench cube-single.
2. **RQ2: What does covariance weighting add to detection?** We compare raw, fixed-whitened, and dynamic Gaussian scores on identical filter innovations, with separate alarm calibration and temporal controls.
3. **RQ3: How does covariance structure affect detection, likelihood, and cost?** We compare dense, diagonal, low-rank, and derivative-free filters on a controlled synthetic representation.

The answers are conditional. Under strong image noise, filtering reduces six-step forecast error on all three tasks, most clearly on TwoRoom (31.0%), yet it worsens clean one-step predictions everywhere. Sensor-change detection improves on Push-T and cube-single but not demonstrably on TwoRoom, the task with the largest forecast gain. Where detection does improve, the source differs: on Push-T most of the gain comes from the filtered innovations, while on cube-single most comes from covariance scoring. In the controlled benchmark, the covariance preferred by Gaussian likelihood is not the one that detects best. A directional decomposition of the score shows how such a tradeoff can arise, although we do not establish that mechanism in the learned representations.

Our contribution is thus an empirical account of when an established inference procedure helps a frozen world model, and for which purpose; we do not introduce a new Kalman estimator. Supporting state-based prediction and simulated Panda studies use different models and are reported separately (Appendix H). The Panda controller in particular uses a learned linear-Gaussian model with classical disturbance augmentation, so its success does not establish the utility of the nonlinear visual wrapper.

2 Related Work

World models and latent uncertainty. Deterministic joint-embedding models predict in representation space, using either learned visual encoders as in LeWM (Maes et al., 2026) or pretrained features as in DINO-WM (Zhou et al., 2025). LeWM also evaluates prediction-error surprise for physically implausible events; we ask whether covariance weighting improves this signal. PlaNet and Dreamer include learned stochastic latent dynamics and inference (Hafner et al., 2019; 2021). Here, the encoder and deterministic predictor are fixed before a filter is added.

Filtering learned dynamics. Kalman variational autoencoders (Fraccaro et al., 2017), deep Kalman filters (Krishnan et al., 2015), and recurrent Kalman networks (Becker et al., 2019) combine learned representations or dynamics with state estimation. We study inference after training and compare scoring rules on the same innovations. Innovation-based noise estimation is also classical (Mehra, 1970). We use a fitted residual scale and assess its consequences empirically; this does not identify observation noise separately from model error.

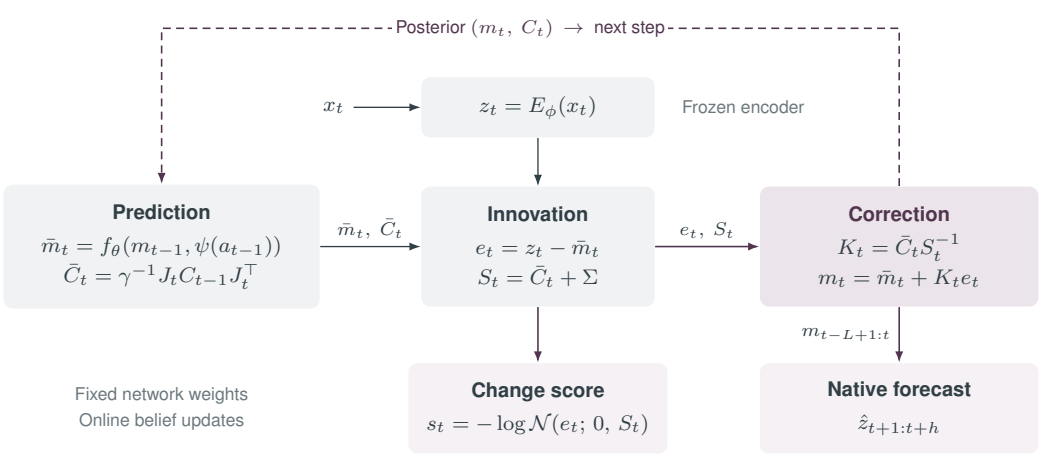

Figure 2: **One step of KalWM.** Frozen dynamics propagate the previous belief; the encoded observation forms the innovation; and the EKF corrects the latent state. The posterior feeds the next step through the dashed path. Detection scores the innovation before correction. For forecasting, L filtered posterior means initialize the frozen model's native rollout ($L = 3$ in the visual experiments). All network weights remain fixed. Algorithm 1 specifies the Jacobian guards and covariance update; Eq. (3) defines the Gaussian score.

Covariance approximations. The unscented filter propagates sigma points (Julier & Uhlmann, 2004); ensemble filters represent covariance through samples (Evensen, 1994; 2003); and square-root ensemble methods supply low-rank recursions (Tippett et al., 2003). These established alternatives let us examine whether observed differences follow covariance structure or the use of local linearization. Our results concern their behavior with the tested frozen predictors.

Disturbance estimation and adaptation. Gradient-based test-time adaptation updates model parameters (Wang et al., 2021; Sun et al., 2020). Augmented-state estimation instead updates a disturbance estimate while keeping dynamics parameters fixed. Bias estimation (Friedland, 1969), unknown-input estimation (Hou & Müller, 1992; Darouach & Zasadzinski, 1997; Gillijns & De Moor, 2007), and offset-free predictive control (Pannocchia & Rawlings, 2003) already provide this construction. Our supplementary Panda study illustrates that principle and does not claim an advantage over those methods.

3 Filtering and Covariance-Weighted Scores

3.1 Setting and outputs

An encoder maps an observation x_t to $z_t = E_\phi(x_t) \in \mathbb{R}^k$, and a predictor advances the latent under action embedding $\psi(a_t)$. The encoder, action encoder, and predictor remain fixed during filtering. We use a one-step transition $f_\theta(z_t, \psi(a_t))$; for a windowed visual predictor, this is its one-step interface rather than its native history-dependent rollout. This restriction is part of the evaluated wrapper.

The filter returns a posterior mean m_t and an innovation score (Figure 2). We use the posterior in two ways: to estimate the current latent, and to initialize the frozen model's future rollout. For visual forecasting, KalWM filters a three-frame history before passing the posterior means to LeWM's native windowed predictor. This preserves the native forecasting architecture; only the observed history is filtered. Detection instead uses the recursive filter's one-step prior and covariance. All visual errors are measured against frozen encodings of clean images, not ground-truth physical states.

3.2 The post-hoc EKF

Figure 2 summarizes the method, which is the classical Kalman recursion (Kalman, 1960) applied to a frozen learned latent. We treat the k-dimensional encoded latent as a noisy measurement with a fitted

diagonal observation covariance $\Sigma = \mathrm{diag}(\sigma^2_{\text{enc}})$. The predictor must have a tractable state Jacobian. The controlled studies use a residual MLP $f_\theta(z, a) = z + g_\theta([z, \psi(a)])$ with SiLU activations; the visual study uses the released LeWM predictor. From $m_0 = z_0,\ C_0 = \Sigma$, each step forms a prior by pushing the posterior through the predictor, $\bar{m}_t = f_\theta(m_{t-1}, \psi(a_{t-1}))$, and linearizes the frozen predictor by reverse-mode automatic differentiation, flooring the Jacobian's singular values so the propagated covariance cannot collapse along a contracted direction:

$$J_t = U_t \max(\Lambda_t,\ s_{\min} I)\, V_t^\top, \tag{1}$$

where $U_t \Lambda_t V_t^\top$ is the SVD of $\partial f_\theta / \partial z\,|_{m_{t-1}}$ and $s_{\min} = 0.1$ throughout (a jitter ϵI, ϵ=10^{-6}, likewise guards the posterior covariance). A residual architecture does not ensure that the floor is inactive. SiLU gives a continuous predictor Jacobian; we do not evaluate alternative activation functions. Writing $\bar{C}_t$ for the prior (predicted) covariance and S_t for the innovation covariance, the filter propagates

$$\bar{C}_t = \tfrac{1}{\gamma} J_t C_{t-1} J_t^\top, \qquad S_t = \bar{C}_t + \Sigma, \tag{2}$$

with discount $\gamma \in (0, 1]$ inflating the propagated covariance (selected on validation data in RQ1 and RQ2; fixed at 0.1 in RQ3). With innovation $e_t = z_t - \bar{m}_t$ and gain $K_t = \bar{C}_t S_t^{-1}$ the posterior is $m_t = \bar{m}_t + K_t e_t$ and C_t follows in Joseph form. The filter returns the per-step Gaussian log-likelihood

$$\ell_t = -\tfrac{1}{2}\left(k \log 2\pi + \log\det S_t + e_t^\top S_t^{-1} e_t\right), \tag{3}$$

Its negation $s_t = -\ell_t$ is the Gaussian innovation score, which we also call surprise. After training, we fit $\sigma^2_{\text{enc},j} = \max\{\mathrm{Var}_t(e_{t,j}), 10^{-6}\}$ from innovations, keeping all network weights fixed. We call this wrapper KalWM. Appendix G gives a numerical example.

Noise fitting and calibration. Innovation variance contains both observation noise and prediction error, so the fitted Σ does not identify measurement noise alone. Adding the propagated covariance can overstate uncertainty (Appendix G.2). We therefore use s_t as a ranking score and fit empirical alarm thresholds, rather than treating its Gaussian form as evidence of calibrated probabilities.

When covariance weighting can help. For a fixed positive-definite covariance $S = U\,\mathrm{diag}(\lambda_i)U^\top$, the quadratic part of surprise is

$$e^\top S^{-1} e = \sum_i \frac{(u_i^\top e)^2}{\lambda_i}. \tag{4}$$

Consider an idealized change from a zero-mean innovation e to $e + \delta$, keeping its covariance and the scoring matrix S fixed. The expected increase in surprise is $\frac{1}{2}\delta^\top S^{-1}\delta$, whereas the expected increase in squared residual is $\|\delta\|^2$. Thus weighting emphasizes changes along low-variance directions and attenuates those along high-variance directions. If $S = cI$ is fixed, surprise and squared residual give identical rankings. These elementary identities describe score response, not detection power: AUROC also depends on the score distributions. In the actual filter, S_t and the innovations vary with time, and the log-determinant term also contributes. RQ2 compares fixed whitening and the quadratic term alone. These comparisons isolate scoring choices; establishing the directional mechanism in RQ3 additionally requires measuring disturbance alignment.

Well-posedness. With finite predictor outputs and Jacobians, positive measurement noise, and the stated numerical guards, the dense identity-observation recursion has $\epsilon I \preceq C_t \preceq \Sigma + \epsilon I$ for $t \geq 1$ in exact arithmetic (Proposition 1). This bounds posterior covariance, not state-estimation error or closed-loop stability. We fix the guard values in these experiments rather than tune them; γ controls covariance inflation.

Fitting and selection. RQ1 and RQ2 use disjoint data for residual-scale fitting, discount selection, alarm calibration, and evaluation. Each discount candidate starts from the same initial scale. The visual study (RQ1) selects by error against clean image encodings; the controlled detection study (RQ2) selects by healthy validation NLL. Both assume a known sensor-noise regime. RQ3 fits scales on training trajectories and thresholds on evaluated pre-change segments, a limitation of that benchmark.

3.3 COMPUTATIONAL COST

The wrapper adds k fitted noise scalars and no trainable network weights. Each step computes a frozen prediction, a $k \times k$ Jacobian ($O(k)$ reverse-mode passes), and dense Kalman algebra: $O(k^3)$ time and $O(k^2)$ memory. The primary studies use $k = 4$ or 192; a separate timing sweep evaluates larger widths. Dense covariance, including the Jacobian SVD and matrix solves, is the main scaling constraint.

3.4 STRUCTURED COVARIANCE

Dense covariance algebra is costly at large k. We consider two structured approximations for the residual-MLP predictor (hidden width h): **diagonal**, where for the residual skeleton the Jacobian factors exactly as $J = I + \Pi\Gamma$, so the exact diagonal of $JCJ^\top$ costs $O(kh^2)$, linear in k with no autodiff and no SVD; and a **low-rank square root** $C_t \approx L_t L_t^\top$ propagated through the same identity and inverted by Woodbury, which retains at most rank ρ without full-rank jitter and reproduces the dense filter at $\rho{=}k$ when jitter is zero and the spectral floor is inactive. Appendix E gives the derivations and discusses an unevaluated block-diagonal extension.

Section 4.3 compares them on synthetic high-dimensional latents against the dense filter and two derivative-free alternatives.

4 EXPERIMENTS

The experiments separate three questions: whether filtering improves the frozen predictor (RQ1), whether covariance scoring improves detection for a given filter (RQ2), and how covariance structure changes this tradeoff (RQ3). RQ1 compares native LeWM and KalWM on the same visual episodes. RQ2 holds the innovations fixed to isolate the scoring rule. RQ3 uses a synthetic representation to compare covariance approximations.

Visual results use one released checkpoint and 128 paired test episodes per task; confidence intervals describe episode variability conditional on that checkpoint and the selected filter. Controlled results report variation across twenty training seeds. Bold identifies the best value within each condition, including displayed ties, rather than statistical significance. In tables that compare the two methods directly, the better entry is shaded: green when KalWM improves on LeWM, red when LeWM is better.

4.1 RQ1: FILTERING FROZEN VISUAL WORLD MODELS

Evaluation setup. We use released LeWM checkpoints for Push-T, TwoRoom, and OGBench cube-single (Maes et al., 2026; Park et al., 2025), keeping all encoder and predictor weights frozen. Each model produces 192-dimensional image embeddings. We add Gaussian image noise with standard deviation $\sigma \in \{0, 0.03, 0.1\}$ in $[0, 1]$ pixel units. For each task, disjoint sets of 48, 48, 64, and 128 recorded episodes serve residual-scale fitting, discount selection, alarm calibration, and testing. Validation denoising error selects $\gamma \in \{0.1, 0.4, 0.7\}$ for each known noise level; forecasting inherits this choice. Episodes supply 12-frame clips sampled every five environment steps. The release datasets may overlap checkpoint pretraining. Appendix C specifies the protocol; Appendix D reports the complete results.

Filtering improves noisy forecasts but can harm clean predictions. Both methods use the native three-frame LeWM rollout and the same recorded future actions. LeWM receives raw encoded observations; KalWM first filters those same three observations. The filter resets at each forecast origin, so both arms receive the same observed history and neither receives future images. We measure squared distance to the clean encoding at horizons of one, three, and six model steps, averaging four origins within each episode before paired bootstrap resampling.

Table 1 shows the central tradeoff. At $\sigma = 0.1$, KalWM lowers prediction error at every evaluated horizon on all three tasks. Six-step reductions are 2.2% on Push-T, 31.0% on TwoRoom, and 1.5% on cube-single; all three paired 95% difference intervals exclude zero (Figure 4). The largest benefit is therefore on TwoRoom, with smaller gains on the two manipulation tasks. At $\sigma = 0.03$, TwoRoom

Table 1: Frozen visual prediction: LeWM versus the same checkpoint with KalWM. Summed squared latent error (↓) uses 128 test episodes per task. Both arms receive the same three observed frames and recorded future actions. One and six model steps span 5 and 30 environment steps. Bold marks the lower error within each horizon. Figure 4 shows all horizons with paired relative-error intervals.

Benchmark	Image noise σ	LeWM ($h = 1$)	KalWM ($h = 1$)	LeWM ($h = 6$)	KalWM ($h = 6$)
Push-T	0	**1.451**	1.535	**18.923**	19.361
	0.03	4.521	**4.139**	33.021	**32.161**
	0.1	120.746	**117.001**	256.116	**250.382**
TwoRoom	0	**12.904**	13.482	35.042	**32.668**
	0.03	16.043	**15.223**	38.109	**34.256**
	0.1	124.666	**79.063**	171.990	**118.669**
Cube-single	0	**0.828**	0.882	**7.639**	7.866
	0.03	2.510	**2.439**	**9.508**	9.702
	0.1	125.202	**118.069**	170.723	**168.130**

improves at all horizons, while the longer-horizon intervals include zero on Push-T and cube-single. On clean inputs, one-step error increases on every task. Filtering helps in the tested high-noise regime, but is not a uniformly beneficial preprocessing step.

Forecasting gains do not predict detection gains. We increase image-noise sd from 0.03 to 0.1 at frame six while keeping recorded actions and dynamics unchanged. Native LeWM is scored by its three-frame predictive residual. KalWM uses either its raw recursive innovation or its Gaussian score. This comparison includes changes in state estimation and effective history; unlike the forecast experiment, the recursive filter can retain earlier observations. Matched-time AUROC compares changed and healthy clips at the same post-onset times, avoiding a comparison of early healthy frames with later corrupted frames.

Table 2: Visual sensor-change detection: matched-time AUROC (↑) on 128 paired episodes. Image-noise sd rises from 0.03 to 0.1. LeWM uses its native three-frame residual; KalWM raw and Gaussian scores share the recursive filter innovations. Brackets are paired episode-bootstrap 95% intervals for Gaussian minus LeWM, without multiplicity adjustment. Table 7 adds the one-frame LeWM baseline and independent false-alarm rates.

Benchmark	LeWM	KalWM raw	KalWM Gaussian	Difference [95% CI]
Push-T	0.9510	0.9876	**0.9906**	$+0.0397$ $[+0.0293, +0.0507]$
TwoRoom	**0.9984**	0.9917	0.9935	-0.0048 $[-0.0157, +0.0004]$
Cube-single	0.9786	0.9804	**0.9877**	$+0.0091$ $[+0.0048, +0.0137]$

KalWM's Gaussian score raises AUROC from 0.9510 to 0.9906 on Push-T and from 0.9786 to 0.9877 on cube-single (Table 2). Both paired difference intervals exclude zero. On TwoRoom, native LeWM already reaches 0.9984; KalWM scores lower, and its difference interval includes zero. Thus the task with the largest forecasting gain has no demonstrated detection gain.

Detection ranking and alarm calibration are evaluated separately. Each detector uses its own threshold fitted on the 64 healthy calibration episodes. KalWM's healthy test false-alarm rates are 5.56%, 5.47%, and 3.39% per step for Push-T, TwoRoom, and cube-single. These rates do not bound the probability of an alarm over an entire episode. Appendix D includes the one-frame LeWM baseline and the full alarm results.

4.2 RQ2: Isolating covariance weighting

The contribution of covariance scoring differs across tasks. Table 2 separates the move from native LeWM to KalWM's raw innovation from the subsequent change in scoring. On Push-T, AUROC rises from 0.9510 to 0.9876 before covariance scoring, then to 0.9906 with the Gaussian score. On cube-single, the first increase is smaller (0.9786 to 0.9804) and the scoring increase is larger (0.9804 to 0.9877). The method comparison therefore reflects different sources of improvement on

the two tasks; the first increase combines state estimation and effective history, whereas the second holds innovations fixed.

In the identical-innovation ablation, Gaussian-minus-raw differences are 0.0031 on Push-T (95% CI $[0.0011, 0.0053]$), 0.0018 on TwoRoom ($[0.0001, 0.0042]$), and 0.0073 on cube-single ($[0.0040, 0.0108]$). Covariance scoring improves the filter's detector on all three tasks, but this alone does not imply improvement over native LeWM, as TwoRoom illustrates.

Scoring gains are modest for additive dynamics changes. To test beyond visual sensor noise, we train and freeze a residual MLP on a four-dimensional point mass. For each of twenty seeds and each noise level, four disjoint sets of 256 trajectories fit the residual scale, select discount by healthy validation NLL, calibrate alarms, and evaluate detection. Candidates start from the same scale. We test additive steps, ramps, sign reversals, and reduced actuator gain. Six scores share the same filter innovations: raw squared residual, fixed diagonal whitening, fixed full whitening with 10% covariance shrinkage, the dynamic quadratic term, the full Gaussian score, and its log-determinant term alone.

Table 3: Matched-time point-mass AUROC at noise sd 0.4 (mean $\pm$ sd over twenty seeds). Scores on changed and paired healthy trajectories are compared only at the same post-onset times. This diagnostic removes differences in the time ranges used for the two classes. All scores use the same filter innovations; larger values indicate change.

Score	Step	Ramp	Reversal	Gain
Raw innovation	0.958 ± 0.002	0.778 ± 0.004	0.879 ± 0.003	0.503 ± 0.001
Fixed diagonal whitening	0.959 ± 0.002	0.780 ± 0.004	0.881 ± 0.003	$\mathbf{0.504 \pm 0.001}$
Fixed full whitening	0.959 ± 0.002	0.779 ± 0.004	0.881 ± 0.003	$\mathbf{0.504 \pm 0.001}$
Dynamic quadratic	$\mathbf{0.962 \pm 0.002}$	$\mathbf{0.786 \pm 0.004}$	$\mathbf{0.886 \pm 0.003}$	$\mathbf{0.504 \pm 0.002}$
Gaussian score	$\mathbf{0.962 \pm 0.002}$	$\mathbf{0.786 \pm 0.004}$	$\mathbf{0.886 \pm 0.003}$	$\mathbf{0.504 \pm 0.002}$
Log-determinant	0.506 ± 0.033	0.501 ± 0.008	0.504 ± 0.028	0.500 ± 0.000

At observation-noise sd 0.4, Gaussian scoring improves matched-time ramp AUROC from 0.778 to 0.786, while gain reduction remains near chance (0.504). Under independently calibrated thresholds, ramp delay falls from 7.09 to 6.72 steps (paired difference -0.37, 95% CI $[-0.44, -0.29]$); healthy per-step exceedance is approximately 5.1%. Thus covariance weighting provides modest benefits for these additive changes, without making surprise a reliable detector of every dynamics change. Appendix B reports all noise levels, thresholds, and temporal controls.

Temporal controls distinguish change response from score drift. A score that changes over time can separate pre-change from post-change frames even without responding to a disturbance. On healthy point-mass trajectories, assigning early and late frames the same pre/post labels gives Gaussian AUROC 0.469 at noise sd 0.4; the log-determinant score shows much stronger temporal drift (Appendix B). Our primary metric therefore compares changed and healthy trajectories over the same post-onset times. This choice matters empirically: on cube-single, quadratic scoring ranks above Gaussian scoring under pre/post AUROC (0.991 versus 0.990), but below it at matched times (0.983 versus 0.988). A paired healthy control changes the conclusion about which score is preferable.

4.3 RQ3: Covariance structure, likelihood, and cost

We lift the point-mass state through a fixed random nonlinear map into 192 dimensions, train a residual predictor, and freeze it. Dense, diagonal, and rank-16 filters share the fitted residual scale and discount $\gamma = 0.1$. An unscented filter (UKF) and an ensemble filter (EnKF, $N = 16$) provide derivative-free comparisons. This supporting benchmark has intrinsic dimension four and uses the original pre/post protocol with thresholds fitted on evaluated pre-change scores; it does not establish visual performance or held-out alarm calibration.

Better likelihood need not mean better detection. Dense EKF and UKF have nearly identical detection and likelihood, so replacing local linearization with sigma points does not help here. Covariance rank has a much larger association with performance: low-rank filtering reaches AUROC

Table 4: Filter comparison on a synthetic lifted point mass (k=192, σ=0.1). Delay uses thresholds fitted to the evaluated pre-change scores at the 95th percentile; the nominal 5% false-alarm rate is in sample. NLL is per step and dimension. Runtime uses one CPU thread. Filters share the frozen predictor, fitted Σ, and discount γ, but differ in covariance approximation and numerical guards. The low-rank filter uses $\rho = 16$; the EnKF uses $N = 16$ members (covariance rank at most $N - 1$). Best per column in bold.

Filter	AUROC ↑	delay ↓	NLL/dim ↓	ms/step
dense EKF	0.61 ± 0.03	17.9 ± 0.8	$\mathbf{-0.43 \pm 0.01}$	2.49
UKF	0.61 ± 0.03	17.9 ± 0.8	$\mathbf{-0.43 \pm 0.01}$	1.36
EnKF (N=16)	0.96 ± 0.01	3.8 ± 0.9	-0.28 ± 0.06	**0.16**
low-rank	$\mathbf{0.97 \pm 0.01}$	3.4 ± 0.7	-0.37 ± 0.05	0.21
diagonal	0.83 ± 0.04	18.9 ± 1.1	-0.36 ± 0.01	0.17
raw residual	0.84 ± 0.05	$\mathbf{2.9 \pm 1.8}$	–	–

0.97, versus 0.61 for dense EKF and 0.84 for the raw residual. Yet its stationary Gaussian NLL is 0.064 nats/dimension worse than dense filtering (paired 95% CI $[0.044, 0.084]$), and its detection delay exceeds that of the raw residual. Likelihood, discrimination, and thresholded delay therefore favor different choices.

Structured covariance reduces cost in this implementation. At width 192, a dense step takes 2.49 ms versus 0.21 ms for rank 16; at width 3072, dense filtering takes 2.3 s while the structured variants take 0.9–3.7 ms. These are implementation-specific CPU timings. Equation (4) suggests that covariance can mask changes along high-variance directions, but the experiment does not measure disturbance alignment or isolate all numerical differences between filters. Rank restriction is an empirical result here, not a general prescription for visual models.

5 Discussion and Limitations

The practical finding is that post-training filtering should be assessed for its intended use. A denoising validation criterion produces forecast gains under stronger noise, but also clean-input costs. Detection requires a separate comparison: TwoRoom has the largest forecast gain yet no demonstrated advantage over native LeWM for sensor-change detection. Even when detection improves, the contribution of covariance scoring differs substantially between Push-T and cube-single. Current-frame denoising, forecasting, and change detection therefore provide complementary evidence; none establishes closed-loop task success.

The evaluation also limits what can be inferred about uncertainty. Fitted innovation variance mixes observation noise and model error; a Gaussian score and an empirical alarm threshold do not establish calibrated covariance. Visual selection assumes a known noise regime and access to paired clean images on validation episodes. Intervals condition on one checkpoint and selected filter settings, without multiplicity adjustment. The experiments cover Gaussian corruption of release-data trajectories, not unseen scenes, changed visual dynamics, or closed-loop control. Post-training use avoids network optimization but still incurs Jacobian and covariance costs. The separate Panda study illustrates matched constant-bias compensation and lacks independent disturbance-adaptive control baselines.

6 Conclusion

KalWM adds classical state estimation to a frozen world model without retraining its networks. Direct LeWM comparisons show improved forecasts under stronger sensor noise and degraded clean one-step predictions. Detection gains depend on both the task and the scoring rule, while controlled experiments show that likelihood and detection can favor different covariances. The contribution is this separation of effects: filtering changes the prediction stream, covariance changes how its errors are ranked, and neither benefit follows automatically from the other.

AI USE STATEMENT

Generative AI tools assisted with manuscript grammar review and code generation. The authors are responsible for the final content of the submission.

ETHICS STATEMENT

The reported experiments concern synthetic dynamics, simulated robot control, and released visual benchmark datasets. The anomaly scores and control results do not certify safe operation of a physical robot; transfer to hardware requires separate validation and appropriate control safeguards.

REFERENCES

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/becker19a.html.

Mohamed Darouach and Michel Zasadzinski. Unbiased minimum variance estimation for systems with unknown exogenous inputs. *Automatica*, 33(4):717–719, 1997. URL https://doi.org/10.1016/S0005-1098(96)00217-8.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5): 10143–10162, 1994. URL https://doi.org/10.1029/94JC00572.

Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003. URL https://doi.org/10.1007/s10236-003-0036-9.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7b7a53e239400a13bd6be6c91c4f6c4e-Abstract.html.

Bernard Friedland. Treatment of bias in recursive filtering. *IEEE Transactions on Automatic Control*, 14(4):359–367, 1969. URL https://doi.org/10.1109/TAC.1969.1099223.

Steven Gillijns and Bart De Moor. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica*, 43(1):111–116, 2007. URL https://doi.org/10.1016/j.automatica.2006.08.002.

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*, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.02193.

Ming Hou and Peter C Müller. Design of observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, 37(6):871–875, 1992. URL https://doi.org/10.1109/9.256351.

Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004. URL https://doi.org/10.1109/JPROC.2003.823141.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. URL https://doi.org/10.1115/1.3662552.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015. URL https://arxiv.org/abs/1511.05121.

Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. URL https://arxiv.org/abs/2603.19312.

Raman Mehra. On the identification of variances and adaptive kalman filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1970. URL https://doi.org/10.1109/TAC.1970.1099422.

Gabriele Pannocchia and James B Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437, 2003. URL https://doi.org/10.1002/aic.690490213.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. URL https://seohong.me/projects/ogbench/.

Reuven Y Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999. URL https://doi.org/10.1023/A:1010091220143.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020. URL https://proceedings.mlr.press/v119/sun20b.html.

Michael K Tippett, Jeffrey L Anderson, Craig H Bishop, Thomas M Hamill, and Jeffrey S Whitaker. Ensemble square root filters. *Monthly Weather Review*, 131(7):1485–1490, 2003. URL https://doi.org/10.1175/1520-0493(2003)131<1485:ESRF>2.0.CO;2.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, pp. 79115–79135, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

Appendix guide. Appendix A specifies the filter. Appendices C–D document the main visual comparison; Appendix B contains controlled detection results. Appendices E–G give covariance details, guarantees, and a worked example. The remaining appendices contain supporting state-based and Panda studies with distinct models and protocols. Calibration diagnostics are not ranked in bold.

A FULL ALGORITHM

Algorithm 1 Post-hoc EKF on a frozen world model (KalWM). The encoder E_ϕ and predictor f_θ are frozen; the noise scale is fitted once before deployment, after which only the belief changes.

Input: frozen E_ϕ, f_θ; fitted observation covariance Σ; discount γ; floor $s_{\min}$; jitter ϵ; stream $\{x_t, a_t\}$
Output: filtered latents m_t, covariance-normalized surprise s_t
Scale fitting: first run the recursion on the designated fitting windows with the initial Σ, set $\sigma^2_{\text{enc},j} \leftarrow \max\{\text{Var}(e_{t,j}), 10^{-6}\}$, then reset before deployment.

$m_0 \leftarrow E_\phi(x_0)$; $\quad C_0 \leftarrow \Sigma$ {initialize from the encoder}
for $t = 1, 2, \ldots$ **do**
  $z_t \leftarrow E_\phi(x_t)$ {encode (frozen)}
  $\bar{m}_t \leftarrow f_\theta(m_{t-1}, \psi(a_{t-1}))$ {prior mean (frozen f_θ)}
  $J_t \leftarrow U_t \max(\Lambda_t, s_{\min} I) V_t^\top$ {linearize + floor (Eq. 1)}
  $\bar{C}_t \leftarrow \frac{1}{\gamma} J_t C_{t-1} J_t^\top$; $\quad S_t \leftarrow \bar{C}_t + \Sigma$ {prior, innovation cov.}
  $e_t \leftarrow z_t - \bar{m}_t$ {innovation}
  $K_t \leftarrow \bar{C}_t S_t^{-1}$
  $m_t \leftarrow \bar{m}_t + K_t e_t$ {posterior mean}
  $C_t \leftarrow (I - K_t)\bar{C}_t (I - K_t)^\top + K_t \Sigma K_t^\top + \epsilon I$ {Joseph form + jitter}
  $s_t \leftarrow \frac{1}{2}\big(k \log 2\pi + \log\det S_t + e_t^\top S_t^{-1} e_t\big)$ {surprise $= -\ell_t$}
end for

B CONTROLLED DETECTION: PROTOCOL AND COMPLETE RESULTS

This appendix supports RQ2. Table 5 reports the full noise and disturbance grid under the pre/post metric; Table 6 diagnoses temporal drift on healthy trajectories. The main text uses matched-time AUROC (Table 3) to avoid confounding that drift with disturbance response.

Data and calibration. The point-mass study trains one residual MLP per seed on $1{,}500$ trajectories with 600 updates, then freezes it. Separate generated trajectories fit its residual scale, select the discount by healthy validation NLL, fit 95th-percentile alarm thresholds, and evaluate the detectors. Each set contains 256 trajectories of 40 transitions. We exclude frames zero through four. The change starts with transition 20, so the first affected observation is frame 21 and has delay zero. Misses receive the 20-step post-change horizon. We also measure detection probability, delay conditional on detection, and the probability of at least one alarm per healthy episode. Fitting and thresholds are noise-specific, assuming the observation-noise regime is known.

Disturbances and scores. The tested disturbances are a constant state increment $(0, 0, 0.3, -0.3)$, a linear ramp to that increment, a reversal of its sign halfway through the affected segment, and an action multiplier of 0.4. The same test initial states, commanded actions, and noise realizations pair the healthy and changed conditions. Fixed whitening uses the fitting-set innovation covariance with 10% shrinkage toward an isotropic covariance of equal trace; diagonal whitening uses its diagonal. Scores use the uncentered innovations, with no fitted mean subtraction. All scores use the same filter innovations. Larger scores trigger alarms throughout, including conditions where the change instead reduces the score. Paired intervals resample twenty training-seed differences $10{,}000$ times and do not adjust for multiple comparisons.

No-change temporal control. For each healthy paired test trajectory, we label the scores before and after the nominal onset time and compute AUROC without injecting a disturbance. Departures from 0.5 reveal temporal variation in the healthy score distribution. At $\sigma = 0.4$, raw and Gaussian values are 0.478 and 0.469; at $\sigma = 0.1$ they are 0.366 and 0.364. The log-determinant term is particularly

Table 5: Held-out point-mass detection AUROC (mean ± sd over twenty training seeds). Each noise level has independent scale-fitting, healthy validation, alarm-calibration, and test trajectories. Discount is selected by healthy validation NLL. FAR is measured per step on independent healthy test trajectories (%). Gain multiplies actions by 0.4 after onset. Scores are oriented so larger values trigger alarms; the low gain-change AUROCs are failures of that detector convention, not inverted post hoc.

σ	Score	Step	Ramp	Reversal	Gain	FAR
0.1	Raw innovation	0.994 ± 0.001	0.860 ± 0.008	0.985 ± 0.008	0.398 ± 0.025	5.25
0.1	Fixed diagonal whitening	0.994 ± 0.002	0.858 ± 0.008	0.986 ± 0.009	0.405 ± 0.023	5.17
0.1	Fixed full whitening	0.994 ± 0.002	0.859 ± 0.008	$\mathbf{0.987 \pm 0.009}$	0.409 ± 0.022	5.19
0.1	Dynamic quadratic	$\mathbf{0.995 \pm 0.002}$	$\mathbf{0.864 \pm 0.008}$	0.986 ± 0.009	$\mathbf{0.412 \pm 0.021}$	5.21
0.1	Gaussian score	0.994 ± 0.002	0.860 ± 0.008	0.986 ± 0.008	0.396 ± 0.023	5.20
0.1	Log-determinant	0.069 ± 0.212	0.051 ± 0.156	0.059 ± 0.182	0.036 ± 0.110	4.99
0.2	Raw innovation	0.983 ± 0.001	0.832 ± 0.006	0.953 ± 0.003	0.450 ± 0.011	5.06
0.2	Fixed diagonal whitening	0.983 ± 0.001	0.831 ± 0.006	0.953 ± 0.003	0.451 ± 0.011	5.09
0.2	Fixed full whitening	0.983 ± 0.001	0.831 ± 0.006	0.953 ± 0.003	0.452 ± 0.011	5.03
0.2	Dynamic quadratic	$\mathbf{0.985 \pm 0.001}$	$\mathbf{0.839 \pm 0.006}$	$\mathbf{0.955 \pm 0.002}$	$\mathbf{0.460 \pm 0.012}$	5.07
0.2	Gaussian score	0.984 ± 0.001	0.834 ± 0.006	0.953 ± 0.003	0.442 ± 0.011	5.07
0.2	Log-determinant	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	4.98
0.4	Raw innovation	0.955 ± 0.002	0.765 ± 0.005	0.871 ± 0.004	0.482 ± 0.008	5.06
0.4	Fixed diagonal whitening	0.956 ± 0.002	0.767 ± 0.005	0.872 ± 0.004	0.481 ± 0.008	5.07
0.4	Fixed full whitening	0.956 ± 0.002	0.766 ± 0.005	0.872 ± 0.004	0.481 ± 0.008	5.05
0.4	Dynamic quadratic	$\mathbf{0.960 \pm 0.002}$	$\mathbf{0.779 \pm 0.004}$	$\mathbf{0.882 \pm 0.003}$	$\mathbf{0.492 \pm 0.008}$	5.09
0.4	Gaussian score	0.959 ± 0.002	0.771 ± 0.004	0.877 ± 0.003	0.474 ± 0.008	5.10
0.4	Log-determinant	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	4.96

time-dependent, with healthy-control AUROC near 0.001 at the two higher noise levels. Its similarly low changed-trajectory AUROC should therefore not be interpreted as a useful inverted disturbance detector. These controls qualify the pre/post discrimination metric; the independently calibrated healthy alarm rates remain measured directly on healthy test trajectories.

Table 6: No-change temporal control (mean ± sd over twenty seeds). Healthy trajectories are assigned the same pre/post time labels as changed trajectories, although no disturbance occurs. AUROC different from 0.5 indicates time-dependent score distributions; it is not disturbance discrimination.

Score	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
Raw innovation	0.366 ± 0.030	0.438 ± 0.010	0.478 ± 0.008
Fixed diagonal whitening	0.377 ± 0.027	0.439 ± 0.010	0.478 ± 0.008
Fixed full whitening	0.381 ± 0.026	0.440 ± 0.010	0.478 ± 0.008
Dynamic quadratic	0.381 ± 0.025	0.447 ± 0.011	0.489 ± 0.008
Gaussian score	0.364 ± 0.028	0.428 ± 0.010	0.469 ± 0.008
Log-determinant	0.037 ± 0.110	0.001 ± 0.001	0.001 ± 0.001

Matched-time discrimination. As a second diagnostic, we compare scores on changed and paired healthy trajectories at the same post-onset times. This keeps the time range identical for both classes. At $\sigma = 0.4$, Gaussian step and ramp AUROCs are 0.962 and 0.786, versus 0.958 and 0.778 for raw innovations. Reduced gain remains near chance. The quadratic and full Gaussian scores are now nearly tied, illustrating why a pre/post ranking alone does not isolate disturbance response.

Table 3 in the main text summarizes the matched-time comparison at noise sd 0.4.

C VISUAL EVALUATION PROTOCOL

Checkpoints and data. We use the released LeWM checkpoints and corresponding datasets for Push-T, TwoRoom, and OGBench cube-single (Maes et al., 2026; Park et al., 2025), using the expert demonstrations for Push-T and cube-single. Each task uses one 12-frame clip from each of 288 distinct episodes, partitioned $48/48/64/128$ for fitting, validation, alarm calibration, and testing.

These partitions are disjoint for filter evaluation; they do not establish a holdout from checkpoint pretraining.

Images and actions. Frames are resized to 224×224 RGB. Independent Gaussian noise with sd 0, 0.03, or 0.1 is added in $[0, 1]$ pixel units and clipped to that range before ImageNet normalization. Clean encodings supply validation and diagnostic targets, never test-time filter inputs. A model step spans five environment steps. The action encoder receives concatenated blocks of five actions: width 10 for Push-T and TwoRoom, and 25 for cube-single. Standardization uses fixed dataset-wide action statistics, following the released evaluation protocol.

Fitting and selection. For each noise level and discount $\gamma \in \{0.1, 0.4, 0.7\}$, a fresh filter estimates diagonal innovation variance on fitting episodes. Validation clean-encoding MSE selects the discount. Forecasting reuses this selection without tuning on forecast error. The selected discounts for noise sd $(0, 0.03, 0.1)$ are $(0.1, 0.4, 0.4)$ on Push-T and $(0.1, 0.1, 0.1)$ on both other tasks. Detection fixes the filter selected at healthy noise sd 0.03 throughout the sensor change. Each detector fits its own 95th-percentile threshold on the 64 separate healthy calibration episodes.

Evaluation windows. Forecast origins are zero-indexed model frames $2, 3, 4, 5$, with horizons $1, 3, 6$. Both methods receive the same three observed frames and future actions; KalWM resets and filters that window before the native rollout. Direct detection scores frames 3–11 to provide LeWM's full three-frame context. KalWM detection carries its posterior through the clip and can retain earlier information. Denoising and identical-innovation score ablations use full-sequence filtering and score frames 2–11. The sensor change begins at frame six, increasing noise sd from 0.03 to 0.1 without changing actions or recorded dynamics. These different warm-up conventions explain why direct-comparison and score-ablation alarm rates need not coincide.

Metrics and intervals. Prediction and denoising errors are summed squared distances to the corresponding clean image encoding. Forecast origins are averaged within each episode. Matched-time AUROC compares changed and paired healthy scores only at post-onset times; pre/post AUROC instead labels early and late scores within changed clips. A healthy temporal control uses pre/post labels without any disturbance. Larger scores always trigger alarms. Delay is zero at the first affected frame; misses receive the six-frame post-change horizon. Error intervals use $10{,}000$ paired episode-bootstrap resamples; AUROC differences use $2{,}000$, both with seed zero. Figures 1 and 4 use the same episode resampling to compute relative forecast-error reduction, $100(1 - \overline{L}_{\mathrm{KalWM}}/\overline{L}_{\mathrm{LeWM}})$, recomputing both mean errors within each resample. Intervals condition on the checkpoint and selected filter, without refitting or multiplicity adjustment.

Validation and computational cost. All encoder, action-encoder, and predictor parameters remained unchanged. Forecasts exactly matched native LeWM rollouts on two observation windows per checkpoint; we also verified action alignment and exclusion of future observations. Sequential filtering agrees with full-sequence filtering in the tested linear systems. Results from earlier runs that reset covariance at every step are excluded. With image embeddings precomputed, the direct comparisons took 10.5, 8.9, and 8.7 minutes for Push-T, TwoRoom, and cube-single on CPU allocations with two threads and batch size four. The earlier TwoRoom and cube runs, including image encoding and discount selection, took 36.9 and 40.8 minutes. These are experiment times, not deployment latencies.

D Complete visual comparisons and diagnostics

The visual results answer three distinct questions: whether filtered histories improve future prediction (Figure 4), whether the complete KalWM detector improves on native LeWM (Table 7), and whether covariance scoring improves a fixed innovation stream (Table 9). Current-frame denoising is a separate diagnostic (Table 8). All results use the protocol in Appendix C; entries repeated from the main text are the same measurements. Figure 3 shows the image corruption evaluated.

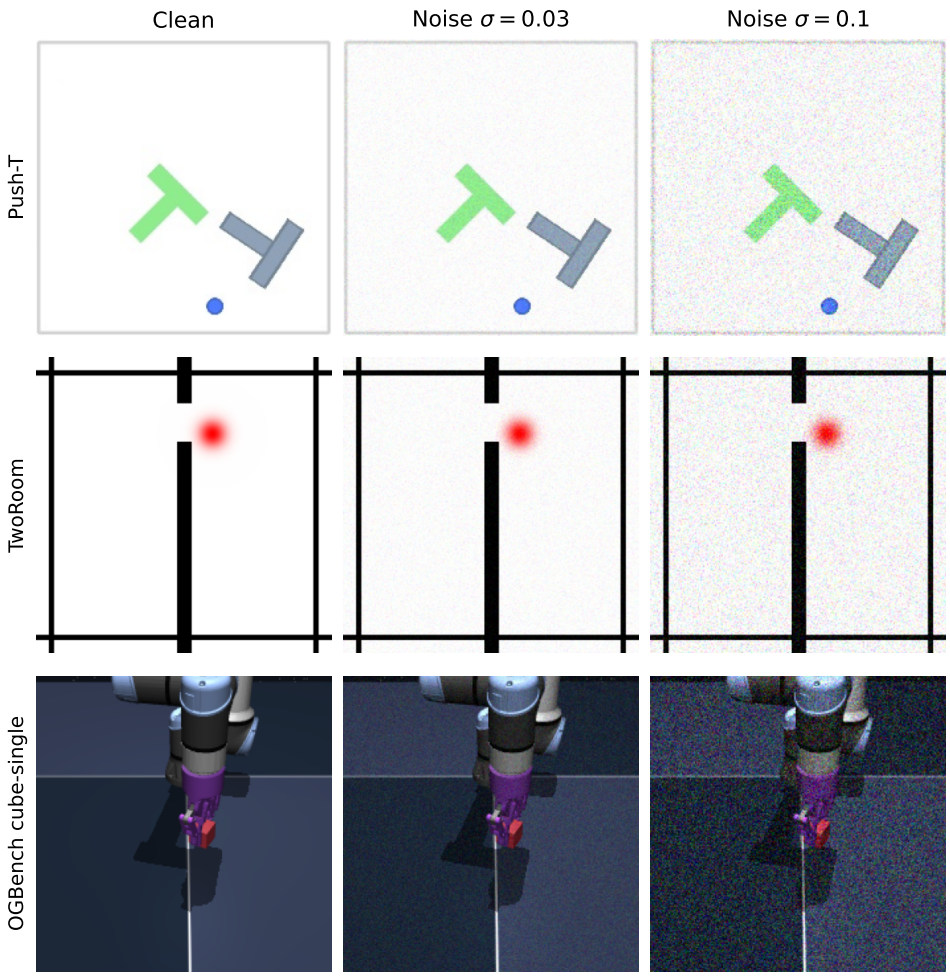

Figure 3: Inputs from the first test episode of each benchmark at model frame six, before encoder normalization. Filtering and all prediction-error measurements operate in the frozen latent space.

D.1 Prediction from matched observation histories

Figure 4 shows the full forecast comparison across noise levels and horizons. Error reductions are relative to native LeWM; positive values favor KalWM. Here KalWM is the same LeWM checkpoint with its three observed embeddings filtered before the native rollout; no separate predictor is trained. The intervals distinguish a consistent benefit at noise sd 0.1 from uncertain longer-horizon differences under milder noise. Clean one-step degradation occurs on every task, and clean cube-single degradation occurs at every horizon.

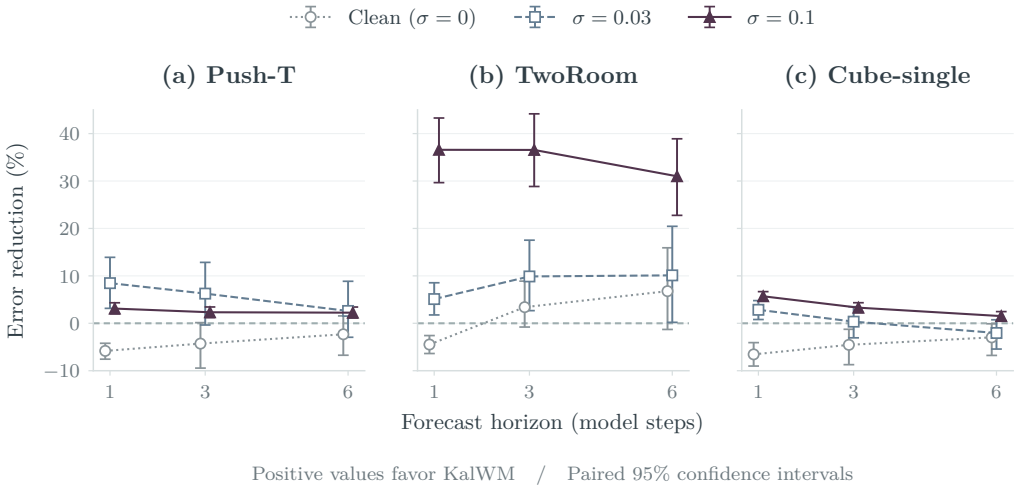

Figure 4: **Prediction from matched observation histories.** Both methods use the same three observed frames, frozen checkpoint, native rollout, and recorded future actions. Each panel reports all noise levels and forecast horizons for one task. Points show $100(1 - \overline{L}_{\mathrm{KalWM}}/\overline{L}_{\mathrm{LeWM}})$; bars are paired 95% episode-bootstrap intervals from 10,000 resamples of 128 test episodes, conditional on the checkpoint and selected filter, without multiplicity adjustment. Noise levels are slightly offset horizontally for readability. One model step spans five environment steps.

D.2 Direct LeWM–KalWM detection comparison

Table 7 adds the one-frame LeWM interface used as the EKF transition. Native three-frame LeWM retains its trained context; recursive KalWM can retain older observations. This comparison therefore evaluates complete estimators. Covariance weighting alone is isolated in the following ablation. The Gaussian-versus-native-LeWM difference interval excludes zero on Push-T and cube-single, but includes zero on TwoRoom. False-alarm rates are measured after independent threshold calibration and are reported separately from AUROC.

Table 7: Complete direct sensor-change comparison on 128 paired test episodes. LeWM uses unfiltered histories; KalWM uses recursive filter innovations. Matched AUROC (↑) compares changed and healthy scores at the same post-onset times. FAR is healthy per-step exceedance with thresholds fitted on 64 separate episodes. Differences and paired 95% intervals compare against native three-frame LeWM, without multiplicity adjustment.

Benchmark	Method	Matched AUROC	FAR (%)	Difference [95% CI]
Push-T	LeWM (3-frame)	0.9510	5.90	–
	LeWM (1-frame)	0.9483	5.90	−0.00266 [−0.00409, −0.00140]
	KalWM raw	0.9876	6.94	+0.03660 [+0.02720, +0.04658]
	KalWM Gaussian	**0.9906**	5.56	+0.03966 [+0.02925, +0.05074]
TwoRoom	LeWM (3-frame)	**0.9984**	4.25	–
	LeWM (1-frame)	0.9976	4.51	−0.00078 [−0.00182, −0.00002]
	KalWM raw	0.9917	3.47	−0.00665 [−0.02075, +0.00015]
	KalWM Gaussian	0.9935	5.47	−0.00485 [−0.01574, +0.00039]
Cube-single	LeWM (3-frame)	0.9786	2.34	–
	LeWM (1-frame)	0.9766	2.52	−0.00198 [−0.00414, −0.00012]
	KalWM raw	0.9804	2.43	+0.00182 [−0.00128, +0.00481]
	KalWM Gaussian	**0.9877**	3.39	+0.00914 [+0.00479, +0.01373]

D.3 Current-frame denoising

Table 8 measures current-frame denoising, a different target and history protocol from the forecast experiment. At noise sd 0.1, error falls by 6.0%, 63.1%, and 5.2% on Push-T, TwoRoom, and cube-single. At sd 0.03, the TwoRoom difference interval includes zero and cube-single error increases by 9.5%. On clean images the raw error is zero by construction; filtering introduces error. The target is the current clean encoding, so these reductions do not measure future prediction or physical-state accuracy.

Table 8: Post-training denoising on three frozen visual LeWM checkpoints. Each benchmark uses 48 fitting, 48 validation, 64 alarm-calibration, and 128 test episodes, with one 12-frame clip per episode. Error is summed squared latent distance to the clean encoding, averaged over ten scored frames. Brackets are paired episode-bootstrap 95% intervals for filtered minus raw error; negative values favor filtering. Latent error scales differ between checkpoints and should not be compared across tasks.

Benchmark	σ	γ	Raw error	Filtered error	Difference [95% CI]
Push-T	0	0.1	**0.000**	0.416	+0.416 [+0.385, +0.451]
	0.03	0.4	2.718	**2.366**	−0.352 [−0.583, −0.123]
	0.1	0.4	115.783	**108.830**	−6.953 [−9.912, −4.115]
TwoRoom	0	0.1	**0.000**	10.237	+10.237 [+9.823, +10.746]
	0.03	0.1	13.508	**12.825**	−0.682 [−2.268, +2.019]
	0.1	0.1	156.415	**57.758**	−98.656 [−110.021, −86.082]
Cube-single	0	0.1	**0.000**	0.317	+0.317 [+0.295, +0.352]
	0.03	0.1	**1.649**	1.805	+0.157 [+0.125, +0.198]
	0.1	0.1	123.910	**117.450**	−6.461 [−7.884, −5.095]

D.4 Covariance scoring with fixed innovations

Table 9 holds the filter innovations fixed and varies only their score. Fixed full whitening uses 10% shrinkage of the fitting-set innovation covariance; diagonal whitening uses its diagonal. The table reports pre/post, matched-time, and healthy temporal-control AUROC. The quadratic/Gaussian ordering reverses on cube-single when times are matched. Low log-determinant pre/post AUROC also occurs without a change, revealing temporal drift rather than evidence for an inverted disturbance detector. The paired healthy comparison and independently measured alarms qualify these scoring results.

Table 9: Sensor-change detection across the three benchmarks. Image-noise sd changes from 0.03 to 0.1 at frame six. Pre/post AUROC compares frames before and after the change. Matched AUROC compares changed with healthy scores at the same post-onset times. Control assigns pre/post time labels to healthy trajectories and diagnoses temporal drift. FAR is healthy per-step exceedance; episode FA is any alarm within ten healthy scored frames. Thresholds use 64 separate episodes. Larger scores trigger alarms throughout.

Benchmark	Score	Pre/post	Matched	Control	FAR (%)	Episode FA (%)
Push-T	Raw	0.992	0.988	0.554	5.94	35.2
	Fixed diagonal	0.992	0.988	0.554	6.48	37.5
	Fixed full	0.991	0.988	0.538	7.58	38.3
	Quadratic	**0.996**	0.990	0.660	5.39	34.4
	Gaussian	**0.996**	**0.991**	0.568	5.31	34.4
	Log-determinant	0.112	0.628	0.078	4.92	49.2
TwoRoom	Raw	0.999	0.992	0.470	3.67	23.4
	Fixed diagonal	0.999	0.992	0.470	3.83	24.2
	Fixed full	**1.000**	0.992	0.463	3.52	25.0
	Quadratic	**1.000**	**0.994**	0.523	5.23	31.2
	Gaussian	**1.000**	**0.994**	0.509	5.31	31.2
	Log-determinant	0.188	0.498	0.189	5.47	53.1
Cube-single	Raw	0.978	0.980	0.493	3.20	23.4
	Fixed diagonal	0.979	0.981	0.495	2.89	21.1
	Fixed full	0.985	0.985	0.505	2.73	20.3
	Quadratic	**0.991**	0.983	0.656	3.59	24.2
	Gaussian	0.990	**0.988**	0.442	3.12	21.1
	Log-determinant	0.286	0.570	0.249	5.00	50.0

E STRUCTURED COVARIANCE FAMILIES

Structured covariances approximate the dense recursion and need not inherit its guard-based bounds. **(i) Block-diagonal:** for P patch tokens of width d_p, per-token blocks (optionally pooling the gain across tokens) cost $O(Pd_p^3)$ instead of $O(P^3d_p^3)$ for the block covariance algebra. Exact decoupling additionally requires compatible block dynamics and block-diagonal initial and noise covariances. **(ii) Diagonal:** for the residual skeleton $f_\theta(z,a) = z + g_\theta([z,\psi(a)])$ with MLP hidden width h, the Jacobian factors *exactly* as $J = I + \Pi\Gamma$, where Γ is the first-layer block acting on z and $\Pi = W_3D_2W_2D_1$ collects the later layers (W_ℓ the MLP weights, D_ℓ the diagonal activation derivatives), so the exact diagonal of $JCJ^\top$ costs $O(kh^2)$, linear in k with no autodiff and no SVD. Eq. (1) becomes the per-dimension floor $\operatorname{diag}(JCJ^\top) \geq s_{\min}^2 \operatorname{diag}(C)$, the rest of the recursion is $O(k)$, and a single pooled scalar gain is the degenerate case. **(iii) Low-rank square root:** $C_t \approx L_tL_t^\top$ with $L_t \in \mathbb{R}^{k\times\rho}$ propagates through the same identity, giving the prior factor $\bar{L}_t = \gamma^{-1/2}J_tL_{t-1}$ and $\bar{C}_t = \bar{L}_t\bar{L}_t^\top$. The innovation covariance $S_t = \Sigma + \bar{L}_t\bar{L}_t^\top$ inverts by Woodbury and its log-determinant follows from the determinant lemma, so the posterior collapses to $C_t = \bar{L}_t(I_\rho + \bar{L}_t^\top\Sigma^{-1}\bar{L}_t)^{-1}\bar{L}_t^\top$. Rank is therefore at most ρ with no re-truncation (the ensemble square-root family; Tippett et al., 2003), at $O(k(h^2+\rho^2)+\rho^3)$ per step, and at $\rho{=}k$ the filter reproduces the dense one when jitter is zero and the spectral floor is inactive (the conditions of our unit test). Nonzero full-rank jitter would destroy exact rank preservation.

F WELL-POSEDNESS OF THE RECURSION

The following statement concerns the dense filter with identity observations in exact arithmetic. It bounds the reported posterior covariance, not the accuracy of the posterior mean, the validity of its uncertainty estimates, or stability of a controller using it.

Proposition 1 (Uniform well-posedness)**.** *Run Eqs.* (2)–(3) *with* $\Sigma \succ 0$*, jitter* $\epsilon > 0$*, spectral floor* $s_{\min} > 0$ *and discount* $\gamma \in (0,1]$*. Assume finite predictor outputs and Jacobians at every evaluated state. Set* $c_* = \min\{\lambda_{\min}(\Sigma), \epsilon\}$ *to account for* $C_0 = \Sigma$*. Then for every* $t \geq 1$*,*

$$\epsilon I \preceq C_t \preceq \Sigma + \epsilon I, \qquad S_t \succeq \Sigma + \tfrac{s_{\min}^2 c_*}{\gamma} I \succ 0. \tag{5}$$

Hence C_t can neither collapse nor diverge, S_t is invertible with $\|S_t^{-1}\| \leq \lambda_{\min}(\Sigma)^{-1}$, and ℓ_t is finite.

Proof sketch. The initialization is positive definite and the singular-value floor makes J_t invertible, hence $\bar{C}_t \succ 0$. The Joseph update before jitter equals $(\bar{C}_t^{-1} + \Sigma^{-1})^{-1} \preceq \Sigma$. Adding ϵI proves both posterior bounds. Initially $C_0 \succeq c_* I$ and subsequently $C_t \succeq \epsilon I \succeq c_* I$. Thus

$$J_t C_{t-1} J_t^\top \succeq c_* J_t J_t^\top \succeq c_* s_{\min}^2 I.$$

Adding Σ to the discounted prior proves the bound on S_t. Finite inputs and Jacobians ensure finite quantities at each finite step. This argument uses a scalar eigenvalue bound; in general $JCJ^\top \succeq s_{\min}^2 C$ does not follow from a singular-value floor. □

The posterior upper bound follows from identity observations and positive measurement noise. Measurement noise alone already ensures $S_t \succeq \Sigma \succ 0$; the jitter and spectral floor additionally prevent degeneration of the propagated posterior covariance. Their values can affect estimates if they become active. The worked example has singular values 1.02 and 0.88, above the chosen floor 0.1, but this is not a guarantee for every learned predictor.

Equation (5) holds in exact arithmetic; finite precision can weaken the measured bound. The Joseph form realizes it as $(I-K_t)\bar{C}_t(I-K_t)^\top + K_t \Sigma K_t^\top$, in which $I - K_t = \Sigma S_t^{-1}$ becomes small exactly as $\bar{C}_t$ becomes large, so the product is a small factor against a large one. Once $\|\bar{C}_t\|/\|\Sigma\|$ grows past roughly 10^7 the cancellation exceeds single precision and the measured upper bound degrades; we verified that the same recursion in double precision satisfies Eq. (5) to 2×10^{-11} at $\|\bar{C}_t\| \approx 1.5 \times 10^8$, confirming the loss is numerical. Jitter therefore does not remove all finite-precision limitations. Extreme covariance scales may require double precision or square-root implementations.

G A WORKED NUMERICAL EXAMPLE

This example illustrates every step of Algorithm 1 on a two-dimensional linear system. The numerical values were independently checked against the analytical Kalman recursion.

Setup. The frozen predictor is one axis of the linear point mass of Sec. 4, $z = (p, v)$ and $f_\theta(z, a) = Az + Ba$ with

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & d \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.9 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}, \tag{6}$$

which provides an exactly specified reference predictor. An exactly linear f_θ makes the linearization exact, $J_t = A$ at every step, so each matrix below follows from one 2×2 product. Observations are $z_t + \varepsilon_t$, $\varepsilon_t \sim \mathcal{N}(0, \sigma^2 I)$, $\sigma = 0.1$; the numerics are those used throughout, $\gamma = 0.1$, $s_{\min} = 0.1$, $\epsilon = 10^{-6}$. Actions are zero in §G.1–G.2 so the numbers isolate the filter.

G.1 One step of the recursion

Starting from $m_0 = z_0 = (1.1541, -0.0293)$ and $C_0 = \Sigma = \mathrm{diag}(0.01, 0.01)$, and receiving $z_1 = (0.7821, 0.0568)$, the step proceeds as follows.

$$\bar{m}_1 = Am_0 = (1.1512,\ -0.0264)$$
$$\Lambda_1 = (1.0220,\ 0.8806) \ \Rightarrow\ J_1 = A \quad \text{(floor inactive)}$$
$$\bar{C}_1 = \tfrac{1}{\gamma} AC_0A^\top = \begin{bmatrix} 0.1010 & 0.0090 \\ 0.0090 & 0.0810 \end{bmatrix}$$
$$S_1 = \bar{C}_1 + \Sigma = \begin{bmatrix} 0.1110 & 0.0090 \\ 0.0090 & 0.0910 \end{bmatrix}$$
$$e_1 = z_1 - \bar{m}_1 = (-0.3690,\ 0.0833)$$
$$K_1 = \bar{C}_1 S_1^{-1} = \begin{bmatrix} 0.9092 & 0.0090 \\ 0.0090 & 0.8892 \end{bmatrix}$$
$$m_1 = \bar{m}_1 + K_1 e_1 = (0.8164,\ 0.0443)$$
$$C_1 = \begin{bmatrix} 0.0091 & 0.0001 \\ 0.0001 & 0.0089 \end{bmatrix}, \qquad s_1 = 0.2207 \text{ nats.}$$

Figure 5(a) shows the same recursion over 40 steps: it lowers position RMSE from 0.092 to 0.085, an 8% reduction. Two features of the method explain why the gain is modest. The singular values of J_1 are 1.02 and 0.88, far above $s_{\min} = 0.1$, so the spectral floor of Eq. (1) never binds, as expected for this particular linear predictor. And the gain is near the identity, $K_1 \approx 0.91I$: since $\gamma = 0.1$ inflates the propagated covariance tenfold, the posterior sits close to the current measurement and the filter acts as a light denoiser. This gain is consistent with the modest denoising effect in the example; it does not establish how filtering behaves for other predictors or noise levels.

G.2 Post-hoc noise-scale fitting

The post-hoc noise-scale fit can correct a poor initial scale in this example. Deploying the filter with a deliberately wrong noise scale $\sigma^2_{\text{enc}} = (1, 1)$, two orders of magnitude too large, gives a mean surprise of 4.037 nats/step over a 400-step stationary stream. Setting $\sigma^2_{\text{enc},j} = \mathrm{Var}_t(e_{t,j})$ from the innovations that run produced yields $(0.0178, 0.0185)$ and a mean surprise of 0.137 nats/step. Re-running with the fitted value reproduces the same innovation variances to four decimals, so this particular example is approximately scale-invariant; a general fixed-point guarantee does not follow.

The fitted innovation variance, approximately 0.018, exceeds the true measurement variance 0.01 because innovations contain prior prediction error as well as observation noise. Adding the propagated prior to this fitted scale can therefore overstate predictive variance. Over 4000 steps, the marginal normalized innovations $e_{t,j}/\sqrt{S_{t,jj}}$ have sample standard deviation 0.32, and 99.8% fall within one nominal standard deviation, compared with 68.3% under a standard normal (Figure 5(b)). The normalized innovation squared $e_t^\top S_t^{-1} e_t$ averages $0.105k$, rather than k. These diagnostics indicate substantial over-dispersion in this example. Neither marginal scale nor mean normalized innovation squared establishes Gaussianity, correct correlations, or temporal independence. Changing covariance parameters can also change score rankings, so detector robustness must be measured separately.

G.3 Estimating a constant disturbance online

Now observe position only, $H = [1 \quad 0]$, hide velocity, and drive the system with an unknown constant input $d = 0.5$ that the model never sees. The observability test of Eq. (9) holds with equality, $\mathrm{rank} \left[\begin{smallmatrix} I-A & -B \\ H & 0 \end{smallmatrix}\right] = 3 = k + n_d$, so Proposition 2 applies with k=2 and n_d=1. Proposition 2 is a statement about $\mathbb{E}[\hat{d}_t]$, so we report the mean and standard deviation of the estimate over 512 independent observation-noise realizations of the same trajectory ($q_d = 0.1$; Figure 5(c)):

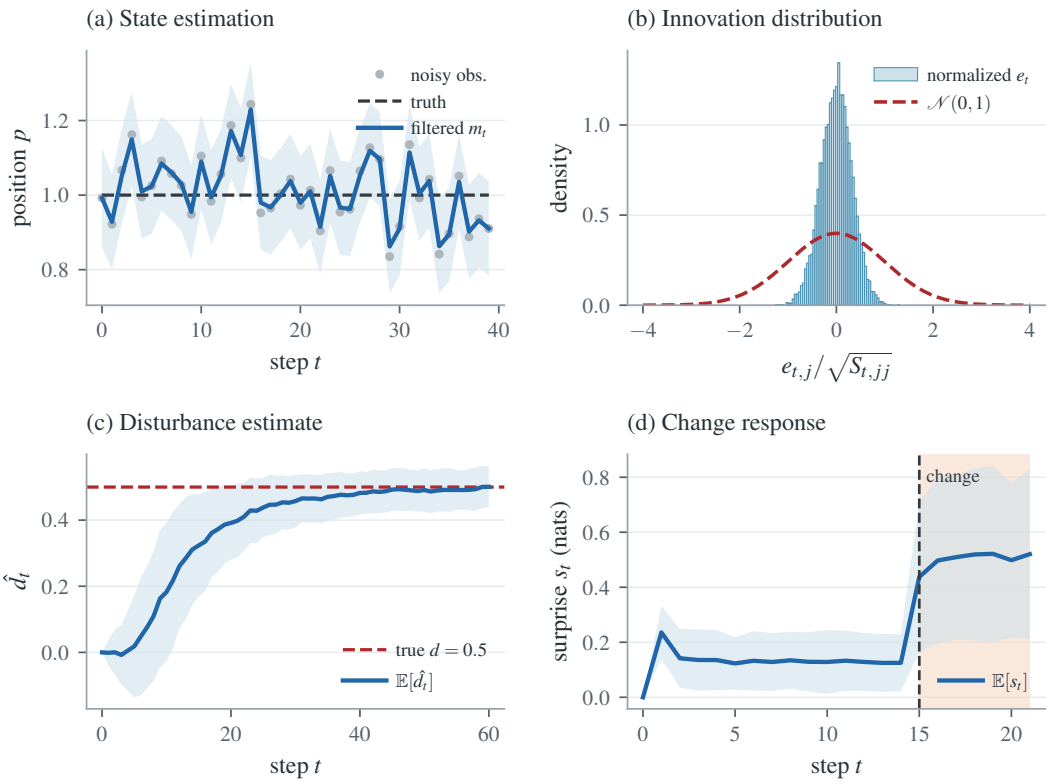

Figure 5: Verification of the worked example. **(a)** the filtered posterior against truth and the noisy stream (band: the filter's own ± 1 sd); **(b)** innovations normalized by the predictive covariance they are scored against, versus the standard normal they would follow if the Gaussian model were calibrated; **(c)** the augmented estimate $\hat{d}_t$ converging to the true disturbance; **(d)** per-step surprise across the unmodeled change at t_c=15 (shaded: post-change). Panels (c) and (d) show the mean over 512 observation-noise realizations with ± 1 sd bands.

t	0	4	8	12	20	30	40	60
$\mathbb{E}[\hat{d}_t]$	0.000	0.005	0.110	0.261	0.392	0.457	0.482	0.501
$\mathrm{sd}(\hat{d}_t)$	0.000	0.121	0.197	0.188	0.111	0.080	0.065	0.062

The estimate is asymptotically unbiased as claimed, reaching 0.501 ± 0.062 by t=60, but far more slowly than the ~ 10 control steps we report on the Panda. The reason is instructive: d enters through B, which touches only the velocity channel, while the measurement sees position alone, so the disturbance is observable yet visible only at second order. Convergence time is governed by how directly the disturbance reaches the measured coordinates, not by q_d alone.

G.4 Surprise at an unmodeled regime change

Finally, fit $\sigma^2_{\text{enc}} = (0.0178, 0.0185)$ on the stationary stream as above, then inject an unmodeled constant acceleration $\delta = 0.3$ at $t_c = 15$. Mean per-step surprise over 512 noise realizations (Figure 5(d)):

t	12	13	14	15	16	17	19
$\mathbb{E}[s_t]$	0.128	0.125	0.125	**0.437**	0.497	0.509	0.521
$\mathrm{sd}(s_t)$	0.106	0.105	0.104	0.272	0.302	0.301	0.320

The signal is flat at ≈ 0.125 nats while the dynamics match the model and steps to ≈ 0.5 within one step of the change. This illustrates a change in the mean score, but neither compares against raw residuals nor determines first-crossing delays. Ratios of Gaussian NLL values are not informative because NLL includes an additive normalization term.

H Supporting prediction and control studies

H.1 Models and comparison scope

These supporting studies use state observations and different trained predictors from the released visual checkpoints. The fully observed methods share an identity encoder, planner, and one-step MSE objective; the partially observed Kalman model uses likelihood training. Here **KalWM** denotes our wrapper around a residual-MLP predictor f_θ whose latent is denoised by the post-hoc EKF of Algorithm 1; its filter-off ablation is the bare predictor. **KalWM-XL** substitutes a deeper residual predictor matched in parameter count to the transformer baseline, separating capacity from architecture. In the partially observed regime we instead learn a linear-Gaussian state-space model, **LinearKalmanWM**, with a non-identity measurement matrix H fit end-to-end by maximizing $\sum_t \ell_t$ (a differentiable Kalman filter, not a wrapper on a frozen predictor). Under distribution shift its **augmented** variant estimates the disturbance $\hat{d}$ online (Eq. 7), with the non-augmented filter as the matched ablation.

The external baseline is **LeWM** (Maes et al., 2026), a windowed autoregressive transformer implementation adapted to state observations that fuses the noisy history through attention and carries no inference-time filter; **LeWM (windowed)** is its native form and our reference point predictor. To test a second predictor family, we expose the trained state-based LeWM predictor as a one-step Markov map (**LeWM (1-step)**) and wrap it in the *same* EKF (**LeWM + EKF**). The clean filtering ablation is therefore LeWM (1-step) with the filter off versus on, not a comparison against the windowed model.

Every Panda experiment uses the LinearKalmanWM family, fit to bias-free rollouts with joint positions observed and velocities hidden (k=14) and then frozen. The three controllers are that frozen model (**Kalman (frozen)**), its **augmented** counterpart, and LeWM. The augmentation is post hoc and no weights move, but the underlying filter is learned rather than wrapped around a frozen nonlinear predictor, so the Panda results speak to the augmentation and not to the wrapper. Table 10 records the distinction. In the supplementary figures, colour marks the model family (blue for Kalman/KalWM, red for LeWM), EKF-on is solid and EKF-off dashed, LeWM's windowed rollout dotted, and the best entry per column bold.

Table 10: Which model each study exercises. The first group tests the post-hoc EKF wrapper (Algorithm 1, a frozen *nonlinear* predictor filtered afterwards); the Panda studies apply post-hoc *augmentation* to a learned filter, and the partial-observation study trains its filter end to end.

Study	**Model**	**Alg. 1**	**Frozen**
Visual comparison (§4.1)	LeWM with KalWM	✓	encoder + predictor
Held-out detection (§4.2)	residual MLP + KalWM	✓	predictor
Prediction/planning (App. H.2)	KalWM, LeWM + EKF	✓	predictor
Regime change (App. H.3)	KalWM	✓	predictor
Filter comparison (§4.3)	KalWM	✓	predictor
Partial observation (App. K)	LinearKalmanWM	×	none
Panda adaptation (App. H.4)	LinearKalmanWM + aug.	×	filter
Panda control (App. H.5)	LinearKalmanWM + aug.	×	filter
Panda faults (App. H.3, App. M)	LinearKalmanWM	×	filter

Table 11: Open-loop prediction error (↓), mean ± std over seeds. Nonlinear dynamics, heteroscedastic observation noise (velocity noise 6σ). Best per noise level in bold.

Method	$\sigma = 0$	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.4$
KalWM + EKF	1.114 ± 0.062	1.143 ± 0.064	1.225 ± 0.067	1.551 ± 0.081	2.870 ± 0.147
KalWM	1.041 ± 0.057	1.082 ± 0.058	1.202 ± 0.062	1.682 ± 0.083	3.656 ± 0.180
KalWM-XL + EKF	1.093 ± 0.108	1.118 ± 0.109	1.193 ± 0.110	1.496 ± 0.116	2.826 ± 0.165
KalWM-XL	1.043 ± 0.105	1.077 ± 0.105	1.182 ± 0.106	1.646 ± 0.117	3.801 ± 0.201
LeWM (windowed)	$\mathbf{0.244 \pm 0.070}$	$\mathbf{0.263 \pm 0.071}$	$\mathbf{0.325 \pm 0.074}$	$\mathbf{0.603 \pm 0.099}$	$\mathbf{1.865 \pm 0.250}$
LeWM (1-step)	1.109 ± 0.126	1.145 ± 0.124	1.253 ± 0.120	1.745 ± 0.134	3.863 ± 0.314
LeWM + EKF	1.186 ± 0.136	1.212 ± 0.134	1.288 ± 0.133	1.595 ± 0.138	2.902 ± 0.246

Planning objective. The state-based prediction studies use CEM (Rubinstein, 1999) to minimize $\|\hat{z}_T - z_g\|_2^2$ over a horizon-T action sequence. The Panda controller instead minimizes mean joint-target error over its planning horizon. These controller studies are separate from the frozen visual prediction and sensor-change comparisons.

H.2 Prediction, planning, and the EKF ablation

Filtering improves prediction at higher observation noise, with a substantial dependence on the discount. We measure eight-step open-loop prediction error and CEM-planning terminal error versus σ (Figure 6, Table 11). In the matched ablation, holding the predictor and training fixed and toggling only the filter, the EKF becomes beneficial for all three predictor families (the small residual MLP, a capacity-matched deep variant, and LeWM's one-step map) at the higher noise levels, and its benefit grows with noise (e.g. $3.66 \to 2.87$ at σ=0.4 in the nonlinear regime). At $\sigma \leq 0.1$ it is neutral or slightly harmful, where there is little noise to remove.

The head-to-head against the external baseline is a separate question. In these rows LeWM's native windowed rollout is the most accurate *point predictor* at every noise level. Matching predictor parameter counts (KalWM-XL) makes a pure capacity explanation unlikely, but does not control for architecture or optimization. The comparison is also sensitive to the operating point: these rows fix the discount at γ=0.1, a setting with strong covariance inflation. Appendix J sweeps it and finds that at γ=0.55, an operating point with approximately unit marginal normalized-innovation scale, the wrapper overtakes the windowed transformer at σ=0.4 in the nonlinear regime (1.751 vs. 1.865; paired difference -0.115, 95% CI $[-0.216, -0.013]$ over twenty seeds), while falling further behind at σ=0.1. So when every state component is measured, the filter is a noise-dependent add-on rather than a uniformly better world model. This advantage is restricted to the tested high-noise setting and discount sweep.

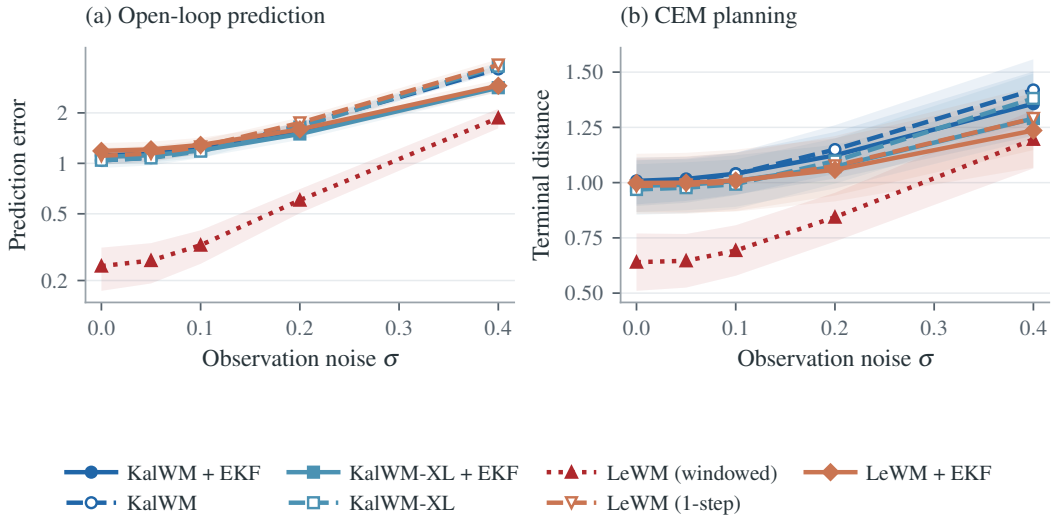

Figure 6: Prediction and planning error vs. noise (nonlinear, heteroscedastic regime; velocity noise is 6σ). Filter-on is solid and filter-off dashed. Filtering helps at higher noise, while LeWM's native windowed rollout is the strongest point predictor throughout.

H.3 Original regime-change protocol

Covariance weighting reduces delay under the in-sample threshold protocol, but does not uniformly improve AUROC. We compare surprise with the squared innovation from the same filter, isolating the scoring rule from the predictor and posterior updates. The LeWM residual uses its one-step Markov map, not the native windowed rollout. On trajectories that switch to perturbed dynamics at a known change point, we score detection by AUROC and detection delay (Figure 7); each signal's threshold is fitted to the pooled pre-change scores of the evaluated trajectories at the 95th percentile. This matches an in-sample per-step alarm rate, not an independently validated false-alarm rate. Delay is the mean first post-change crossing, with non-crossings assigned the remaining horizon. Surprise and the raw residual are effectively tied at $\sigma \leq 0.05$; from $\sigma{=}0.1$ onward, surprise has lower delay than both residual baselines. It also has the best AUROC at $\sigma{=}0.1$ and 0.2, while LeWM's residual is best at $\sigma{=}0.4$.

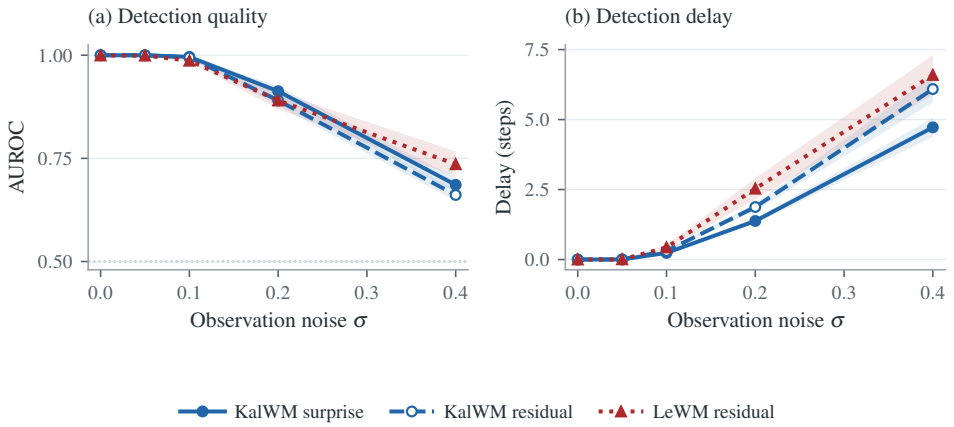

Figure 7: Regime-change detection with thresholds fitted on the evaluated pre-change segments. EKF surprise gives the lowest delay from $\sigma{=}0.1$ onward and the best AUROC at moderate noise; LeWM's residual has the highest AUROC at $\sigma{=}0.4$.

H.4 Online adaptation to unmodeled dynamics

Disturbance augmentation reduces prediction error under a matched constant input bias. Trained on stationary dynamics, the model meets a constant unmodeled disturbance at deployment. The augmented filter estimates it online while the non-augmented model retains a systematic error (Figure 8). On the point mass, adaptation cuts post-shift prediction error by $\sim 40\%$ at the largest disturbance ($0.060 \rightarrow 0.036$), at the price of a higher error when there is no shift to estimate ($0.009 \rightarrow 0.017$). On the simulated Panda, with joint positions observed and an actuator calibration bias δ introduced at test time, the augmented filter's estimate $\hat{d}$ converges to the true bias within ~ 10 control steps and cuts post-bias joint-prediction error by roughly 70–80% at biases of 0.1 rad and above (e.g. $0.118 \rightarrow 0.025\,\text{rad}^2$ at 0.15 rad bias), while the errors of the frozen model and LeWM grow with the bias.

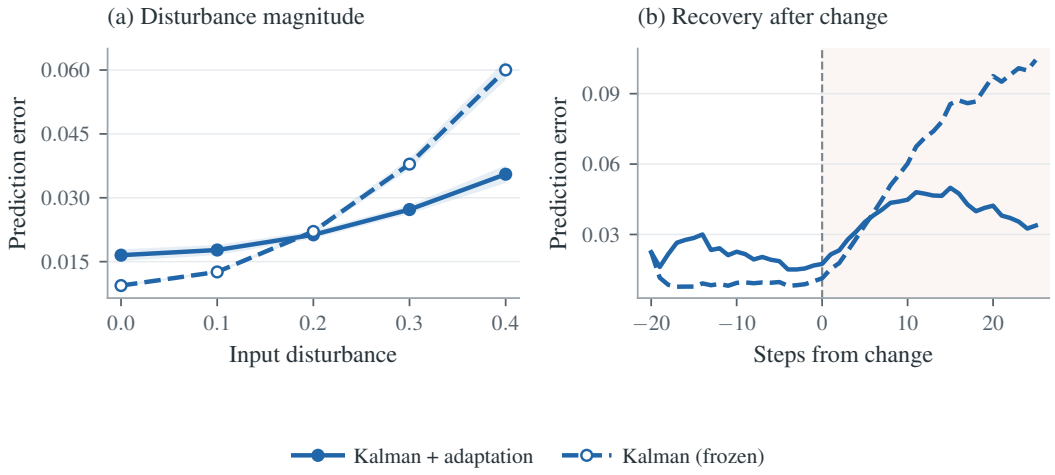

Figure 8: Online adaptation to an unmodeled disturbance (point mass). The augmented filter recovers prediction accuracy as the disturbance grows; the recovery trace (right) shows a smaller post-change error for the augmented filter.

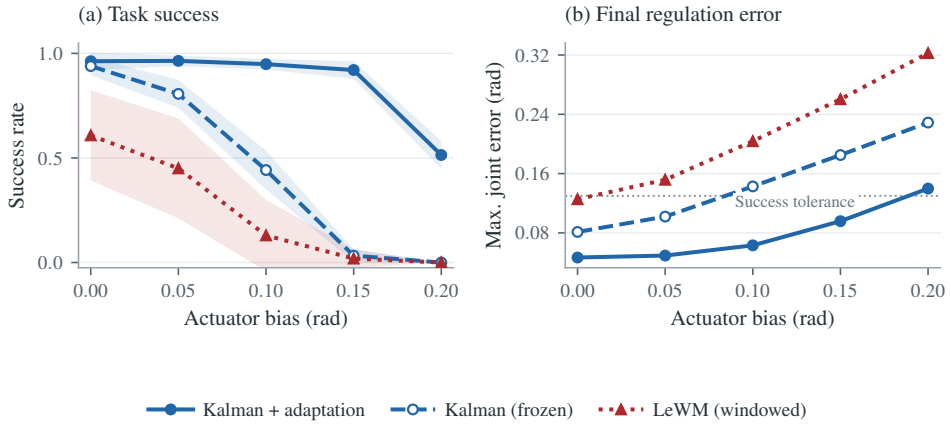

Figure 9: Closed-loop precision control on the Panda: task success (left) and final regulation error (right) vs. actuator bias. Non-adapting controllers collapse as the bias grows; the augmented filter holds near-nominal success.

H.5 Closed-loop precision control

The matched augmented-versus-non-augmented comparison supports constant-bias compensation. A simulated Panda is driven by MPC over the frozen world model to a joint target under an actuator calibration bias (Figure 10). Success means final maximum absolute joint error below 0.13 rad after 25 control steps, over 32 episodes per seed. CEM uses horizon 6, 64 candidates, and 3 iterations; the same bias is added to all seven commanded joint targets. Proposition 2 explains the offset-rejection mechanism under linear assumptions, without guaranteeing performance in this simulator. At zero bias both Kalman controllers are competent (0.96 and 0.94) while LeWM is lower and far more variable (0.61 ± 0.22), so we do not read the control gap as a clean architecture comparison. As the bias grows the frozen model and LeWM fall to zero, while the augmented filter holds 0.92 through 0.15 rad before falling to 0.51 at 0.2 rad, with no retraining (Figure 9). The disturbance is an equal constant offset on all seven joint commands, matching the augmentation. No independent disturbance-observer or offset-free MPC baseline is evaluated, so the result does not establish an advantage over those established approaches.

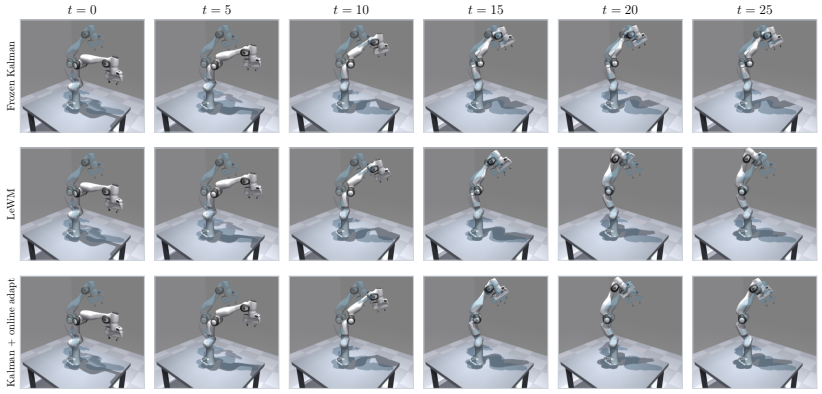

Figure 10: Simulated Panda control under constant actuator bias (translucent blue: goal). The augmented linear-Gaussian model (bottom) estimates the bias online; the frozen non-augmented model (top) and LeWM (middle) retain an offset. This illustrates disturbance compensation on joint observations, not filtering of a frozen nonlinear visual model.

I PROTOCOLS FOR SUPPORTING STATE-BASED STUDIES

The fully observed point-mass configuration uses $1{,}500$ training trajectories, 600 training updates, a history of 4 observations, and 512 evaluation trajectories. Prediction error is the sum of squared coordinate errors, averaged over the 8 rollout steps and evaluation trajectories against clean state. Training uses Adam with learning rate 10^{-3} and batches of 256 sampled windows. The residual MLP has two width-128 SiLU hidden layers and a 32-dimensional action embedding. LeWM uses two transformer layers, width 64, four attention heads, and MLP width 128. Observation noise is added to the state; in the nonlinear experiment the velocity channels have six times the position noise scale. Residual-scale fitting reuses the training trajectories. The discount sweep trains one residual predictor per seed and refits its noise scale for each discount, with separate generated trajectories at $\sigma = 0.1$ for marginal scale diagnostics. Paired 95% intervals in the discount sweep use percentile bootstrap resampling of the twenty seed differences ($10{,}000$ resamples; NumPy PCG64 seed 0), without multiplicity correction. The structured-filter intervals use the same procedure; statistical metrics use twenty seeds, while runtime measurements have three recorded repeats per width. The sweep and original prediction table use separate scale-fitting runs, so their $\gamma = 0.1$ rows need not coincide exactly. The sweep refits noise sequentially across discounts, which may introduce dependence on the preceding scale fit.

The Panda control experiment trains on $1{,}200$ bias-free trajectories of length 30, with 500 training updates. Joint observations have Gaussian noise sd 0.01 rad; velocities are hidden. Goals are sampled within ± 0.4 rad of the home pose per joint. Each evaluation runs 32 goals per seed for 25 control steps with ten simulator steps per control step. CEM minimizes mean squared joint-target error over its six-step horizon, with 64 candidates, 3 iterations, and initial exploration sd 0.15 rad. Success tests only the final maximum joint error against 0.13 rad, not sustained goal holding. Models and bias conditions use the same goals and seeded evaluation noise. The augmented filter uses $q_d = 0.03$.

Regime-change delay is the mean first threshold crossing after onset; trajectories without a crossing receive the remaining post-change horizon. Thresholds pool the evaluated pre-change scores and target 5% in-sample per-step exceedance. This protocol does not measure independent deployment false alarms or average run length. Seed variation and evaluation-episode variation are distinct; the reported bands summarize training-seed variation.

J WHAT THE DISCOUNT γ CONTROLS

Appendix G shows a Kalman gain near the identity and substantial predictive over-dispersion at $\gamma = 0.1$. We therefore vary γ while keeping one trained predictor fixed per seed and refitting Σ from innovations for each setting. The filter-off reference is independent of γ. Table 12 and Figure 11 report this exploratory sweep. The noise scale is refitted sequentially across discounts, with each fit initialized from the preceding estimate; this dependence on the sweep order is a further limitation.

γ trades low noise against high noise. Raising γ inflates the propagated covariance less, so the posterior leans further on the model and less on the current measurement. In these experiments, this improves high-noise prediction and worsens low-noise prediction. At σ=0.4, moving from γ=0.1 to γ=0.7 cuts eight-step error by 44% in the linear regime ($0.397 \to 0.222$) and by 39% in the nonlinear one ($2.817 \to 1.711$), to roughly half the filter-off error in both (0.472 and 3.656). At σ=0.1 the same move is a loss: $0.092 \to 0.122$ linear and $1.229 \to 1.394$ nonlinear, against filter-off values of 0.092 and 1.202. Figure 11(a) shows the two effects crossing. There is therefore no dominant γ. The original setting, $\gamma = 0.1$, gives smaller gains at high noise and smaller losses at low noise in this sweep.

At $\sigma = 0.4$ in the nonlinear regime, the wrapper reaches error 1.751 at $\gamma = 0.55$ and 1.711 at $\gamma = 0.7$, against LeWM's 1.865. The paired differences over twenty seeds are -0.115 (95% CI $[-0.216, -0.013]$) and -0.155 ($[-0.258, -0.052]$), respectively. At $\gamma = 0.4$, the difference is $+0.001$ ($[-0.099, +0.102]$). These intervals describe individual evaluated settings.

At $\gamma = 0.55$, marginal normalized-innovation standard deviations are 0.978 (linear) and 1.106 (nonlinear). This approximately matches marginal scale, but does not establish Gaussianity, temporal independence, joint calibration, or calibrated behavior at other noise levels. The same setting performs poorly against LeWM at $\sigma = 0.1$ (1.361 versus 0.325). These are exploratory sweep results, not an

Table 12: Post-hoc sweep of the EKF discount γ over a single frozen predictor per seed (mean over twenty seeds; σ^2_{enc} re-fit at every γ). Left block: 8-step open-loop prediction MSE with the filter on, for both regimes. Right block: the sample sd of $e_{t,j}/\sqrt{S_{t,jj}}$, whose population value is 1 under correct calibration (necessary, not sufficient). The filter-off ablation carries no γ and sits at linear 0.092 ($\sigma = 0.1$), 0.472 ($\sigma = 0.4$); nonlinear 1.202 ($\sigma = 0.1$), 3.656 ($\sigma = 0.4$). The original prediction comparison fixes γ=0.1; RQ1 and RQ2 select it on validation data. Bold marks the lowest prediction errors among the evaluated discounts.

	linear		nonlinear		sd of $e/\sqrt{S}$	
γ	$\sigma = 0.1$	$\sigma = 0.4$	$\sigma = 0.1$	$\sigma = 0.4$	lin.	non.
0.05	**0.092**	0.432	**1.214**	3.187	0.266	0.318
0.1	**0.092**	0.397	1.229	2.817	0.376	0.466
0.2	0.094	0.339	1.256	2.322	0.535	0.649
0.4	0.103	0.263	1.320	1.867	0.787	0.911
0.55	0.112	0.234	1.361	1.751	0.978	1.106
0.7	0.122	**0.222**	1.394	**1.711**	1.156	1.316
1	0.138	0.225	1.449	1.735	1.392	1.774

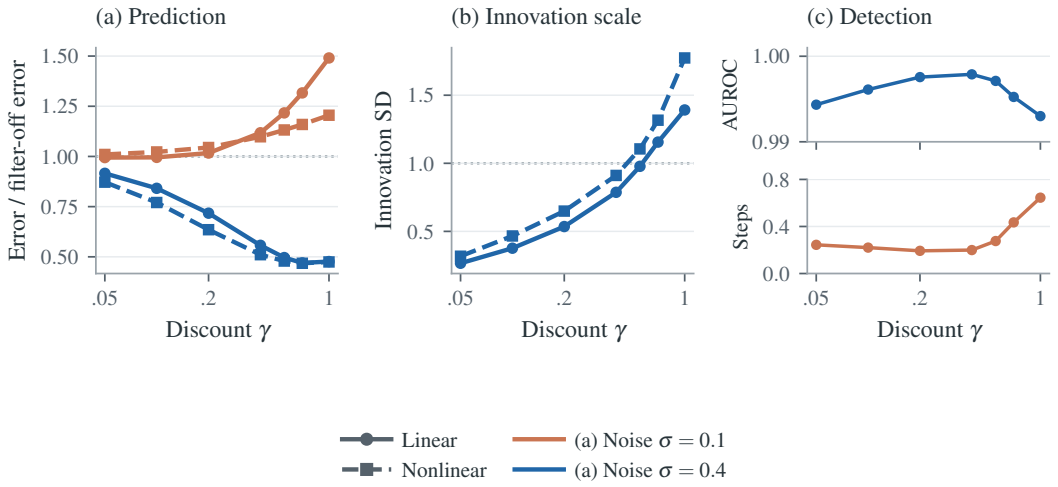

Figure 11: The discount γ as an operating point. **(a)** prediction error relative to the filter-off ablation, so 1 means equal error and the two regimes' very different error scales are comparable; **(b)** the sample sd of the covariance-normalized innovations, whose population value is 1 under correct calibration, a necessary but insufficient diagnostic; **(c)** AUROC (top) and detection delay (bottom) on the linear change-point protocol. Solid: linear regime. Dashed: nonlinear, heteroscedastic regime.

independently validated discount-selection procedure; paired intervals describe the evaluated settings and do not adjust for selecting among discounts.

Marginal scale matching has a prediction cost. The sd of $e_{t,j}/\sqrt{S_{t,jj}}$ rises with γ and crosses 1 near 0.5 in both regimes (Figure 11(b)). This corrects a marginal dispersion diagnostic, at the cost of worse low-noise prediction, but does not by itself justify calling the predictive distribution calibrated. The original prediction comparisons retain $\gamma = 0.1$; RQ1 and RQ2 use independent validation selection. The adaptation and control experiments use a separate learned linear-Gaussian filter without this discount.

Detection changes little in the tested high-separability setting. Across the tested discounts, AUROC on the linear change-point protocol remains within 0.993–0.998, while mean delay changes from 0.19 to 0.65 steps. This is empirical robustness in a specific high-separability regime, not a general invariance to γ. Changing γ affects both the innovation and its covariance. Detection is

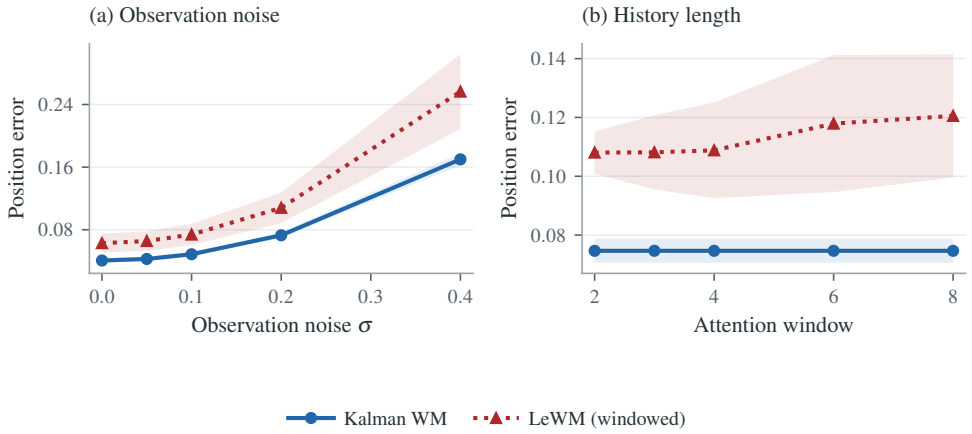

Figure 12: Partial observation (position seen, velocity hidden). The Kalman world model (full history) beats LeWM at every noise level and every attention window.

evaluated only with matching linear training dynamics; the nonlinear sweep omits this column rather than score its predictor on a mismatched linear protocol.

K Partial observability

We observe position only, so prediction requires inference of hidden velocity from the measurement history. LinearKalmanWM and a windowed LeWM are trained on the same data (Figure 12). LinearKalmanWM is the more accurate predictor at every noise level ($\sim$34% lower at σ=0.4, 0.170 versus 0.256) and, because its recursive posterior summarizes the full supplied history, is independent of the attention-window sweep: LeWM trails it at every window tested. This is a comparison of two trained architectures, not a post-hoc filter-on/off ablation; differences in architecture and training objective prevent attributing the full gap to filtering.

L Offset-free estimation: proof

When the latent is only partially observed, $y_t = Hz_t + \varepsilon_t$ with H non-square, the same recursion estimates the hidden coordinates through the dynamics (Eq. (2) with the innovation covariance $S_t = H\bar{C}_t H^\top + \Sigma$ and gain $K_t = \bar{C}_t H^\top S_t^{-1}$). In that setting we learn a linear-Gaussian state-space model end-to-end by maximizing $\sum_t \ell_t$ (a differentiable Kalman filter). For deployment-time *distribution shift*, we augment the state with an unknown input disturbance d under a slow random-walk prior, so predictions become $z' = Fz + G(a + \hat{d})$:

$$\begin{bmatrix} z_{t+1} \\ d_{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} F & G \\ 0 & I \end{bmatrix}}_{\tilde{F}} \begin{bmatrix} z_t \\ d_t \end{bmatrix} + \underbrace{\begin{bmatrix} G \\ 0 \end{bmatrix}}_{\tilde{G}} a_t + w_t, \tag{7}$$

$$y_t = \underbrace{\begin{bmatrix} H & 0 \end{bmatrix}}_{\tilde{H}} \begin{bmatrix} z_t \\ d_t \end{bmatrix} + \varepsilon_t, \tag{8}$$

with $w_t \sim \mathcal{N}\big(0, \mathrm{blkdiag}(Q, q_d^2 I)\big)$. Here Q is the learned process noise of the state block and the $q_d^2 I$ block is the random walk $d_{t+1} = d_t + \eta_t$ that lets $\hat{d}$ move slowly at test time (q_d=0.1 on the point mass, 0.03 on the Panda). The filter estimates $\hat{d}$ online from innovations and the controller includes it in its predictions. This uses classical disturbance estimation (Pannocchia & Rawlings, 2003), applied post hoc to a pretrained model, and toggling the augmentation is a clean ablation. The following classical argument states conditions for asymptotically unbiased augmented estimates:

Proposition 2 (Classical constant-disturbance estimation)**.** *Consider the augmented model of Eqs.* (7)–(8)*, with $\tilde{F}$ and $\tilde{H}$ as defined there, with correctly specified F, G, H, zero-mean process and mea-*

surement noise, and an unknown constant disturbance $d_t \equiv d$. Suppose (i) (F, H) is detectable; (ii) the disturbance modes at eigenvalue one satisfy

$$\operatorname{rank} \begin{bmatrix} I - F & -G \\ H & 0 \end{bmatrix} = k + n_d, \tag{9}$$

which requires the disturbance dimension n_d to be at most the number of measurements; and (iii) the steady-state filter is stable. Then the augmented estimate is asymptotically unbiased, $\mathbb{E}[\hat{d}_t] \to d$ and $\mathbb{E}[m_t - z_t] \to 0$. For a feasible reference and a stable linear nominal regulation loop with unconstrained disturbance cancellation, the mean regulation error consequently tends to zero. Without augmentation or another offset-rejection mechanism, a steady-state bias can remain; a nonzero bias is not inevitable for every controller or disturbance. The Panda experiment evaluates the same compensation principle with learned approximate dynamics; the proposition does not guarantee its performance.

Proof sketch. The only additional potentially unobservable modes of $(\tilde{F}, \tilde{H})$ occur at eigenvalue one. The rank condition excludes those modes, while detectability of (F, H) handles the remaining unstable modes. Let L be the stable steady-state update gain and let b_t denote mean posterior estimation error. Correctly specified mean dynamics and zero-mean noise give $b_t = (I - L\tilde{H})\tilde{F} b_{t-1}$, which tends to zero by assumption (iii). A positive design value of q_d affects the gain and variance, but introduces no mean forcing for constant d. Thus $\mathbb{E}[\hat{d}_t] \to d$. Cancelling $\hat{d}_t$ leaves a mean input error tending to zero; stability and reference feasibility then yield zero mean regulation error under the stated linear control assumptions (Pannocchia & Rawlings, 2003). This result does not guarantee nonlinear constrained MPC performance on the Panda. □

Scope beyond constant disturbances. Proposition 2 covers constant disturbances in a correctly specified linear model. A random-walk disturbance prior may also track slowly varying inputs, with q_d trading response speed against estimation variance, but neither tracking accuracy nor graceful failure under state-dependent or rapidly varying disturbances follows from the proposition. Unexplained disturbances may increase surprise, although covariance inflation can also mask them. More expressive disturbance bases and adaptive noise models are possible extensions (Pannocchia & Rawlings, 2003; Mehra, 1970); they are not evaluated here.

M ONLINE FAULT DETECTION ON THE PANDA

The same frozen filter also flags simulated stuck-joint faults. Partway through a trajectory a subset of joints *sticks* (severity α: 0 none, 1 fully stuck) while the model still sees the commanded action; we detect the fault from the per-step surprise $s_t = -\ell_t$ and score AUROC over steps before versus after the nominal onset (Figure 13). AUROC rises monotonically with severity to 0.80 at a full lock, where the thresholded detector fires within ~ 4 control steps. At $\alpha{=}0$ there is no physical fault; the observed AUROC of 0.554 instead reveals mild temporal drift between the nominal pre- and post-onset segments and should be read as a negative control, not as a fault-detection result. AUROC is threshold-free, but latency requires an alarm threshold: we set it to the 95th percentile of the pooled pre-onset scores, targeting a 5% in-sample false-alarm rate, and report the median first crossing after onset. LeWM remains weak throughout (≤ 0.58). Both predictors condition on observations and actions; their different fault sensitivity may reflect learned dynamics, history use, and training, rather than an absence of commanded-action information in LeWM. The gain here is the model-based prediction, not covariance normalization: over twenty seeds the raw same-model residual and the covariance-normalized surprise are indistinguishable at full lock (0.802 ± 0.020 versus 0.802 ± 0.021), and the residual is marginally ahead at every lower severity.

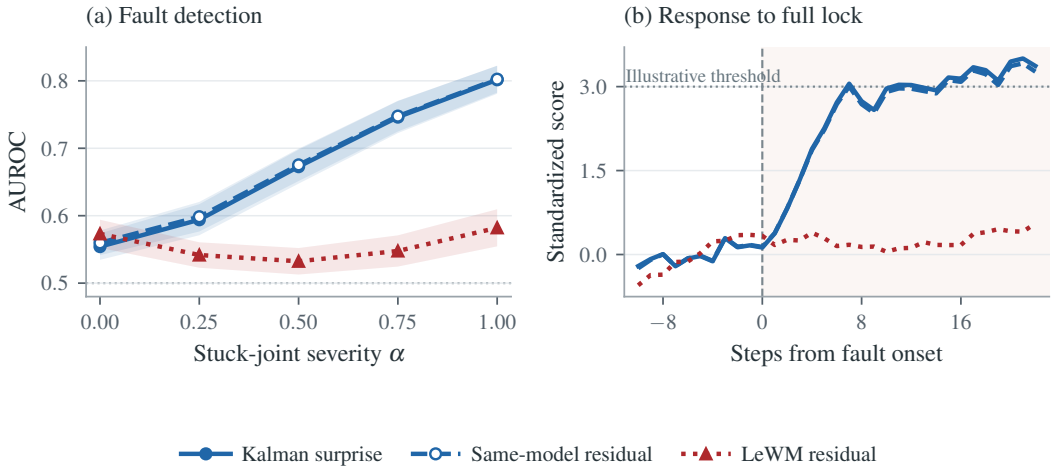

Figure 13: Online fault detection on the Panda. **Left:** AUROC vs. stuck-joint severity: the model-based filter rises to 0.80 while LeWM remains weak. **Right:** surprise vs. steps since onset at a full lock (fault shaded); the filter rises after onset while LeWM remains below the illustrative three-standard-deviation threshold. This display threshold differs from the percentile threshold used for latency metrics.