FILTER BEFORE ADAPTING LATENT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation helps latent world models remain effective under deployment shift. However, direct adaptation treats every prediction–observation mismatch as a learning signal, even when the discrepancy arises from a transiently unreliable observation. Such erroneous updates can persist after the observation recovers, degrading subsequent prediction and planning. We formulate this problem as latent residual attribution, which seeks to recover the clean-target prediction-side residual from an observed discrepancy that also contains observation-side error. We introduce BayesJEPA, which estimates prediction-side and observation-side uncertainty and uses their relative reliability to filter the residual before adapting the predictor. Under second-moment assumptions, this uncertainty-weighted rule is optimal among linear attribution rules; under Gaussian uncertainty, it becomes the posterior mean with a Kalman-style gain and induces the posterior-optimal estimate of the corresponding clean-target adaptation gradient. On PointMaze, BayesJEPA improves recovery of the clean-target residual, reduces harmful updates and parameter drift under transient visual corruption, and preserves useful adaptation under persistent dynamics shifts. Under mixed dynamics and observation shifts, it further improves planning over direct adaptation. Together, our theoretical and empirical results establish filtering before adapting as a principled and effective approach to robust test-time adaptation of latent world models.

1 INTRODUCTION

Latent predictive world models are becoming an increasingly practical basis for planning and control. Rather than reconstructing every detail of high-dimensional observations, Joint-Embedding Predictive Architectures (JEPAs) learn to predict directly in representation space (LeCun et al., 2022; Assran et al., 2023; Bardes et al., 2024). Recent work has shown that such latent models can support reward-free offline planning, zero-shot planning with pretrained visual features, and planning on real robotic systems (Sobal et al., 2025; Zhou et al., 2025; Assran et al., 2025), while representation geometry can itself be shaped to improve planning (Wang et al., 2026b). Deployment, however, can expose a world model to sensing conditions and dynamics that differ from those seen during training. A fixed model cannot correct such mismatch once deployed. AdaJEPA addresses this by direct test-time adaption (TTA) on the predictor from newly observed transitions before replanning (Wang et al., 2026a). In this setting, the discrepancy between what the model predicts and what it subsequently observes becomes the signal for adaptation. This raises a fundamental question:

What should a latent world model learn from a prediction–observation mismatch?

A prediction–observation mismatch can have different origins. It may arise from the prediction side, in which case adaptation can be beneficial, or from an unreliable current observation, in which case learning from the full mismatch can be harmful. Direct adaptation, as in AdaJEPA (Wang et al., 2026a), effectively attributes the entire mismatch to the prediction side and therefore adapts to the full residual. For latent world models, this can turn a transient observation error into longer-lasting model degradation, since the resulting parameter updates affect subsequent predictions and planning rollouts. Figure 1 illustrates this effect: after a short corruption interval, AdaJEPA can remain below FrozenJEPA (Wang et al., 2026b) even after the observation disturbance has ended. This motivates a latent residual attribution view of TTA: determining how much of the observed mismatch should be assigned to the prediction side and used for adaptation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

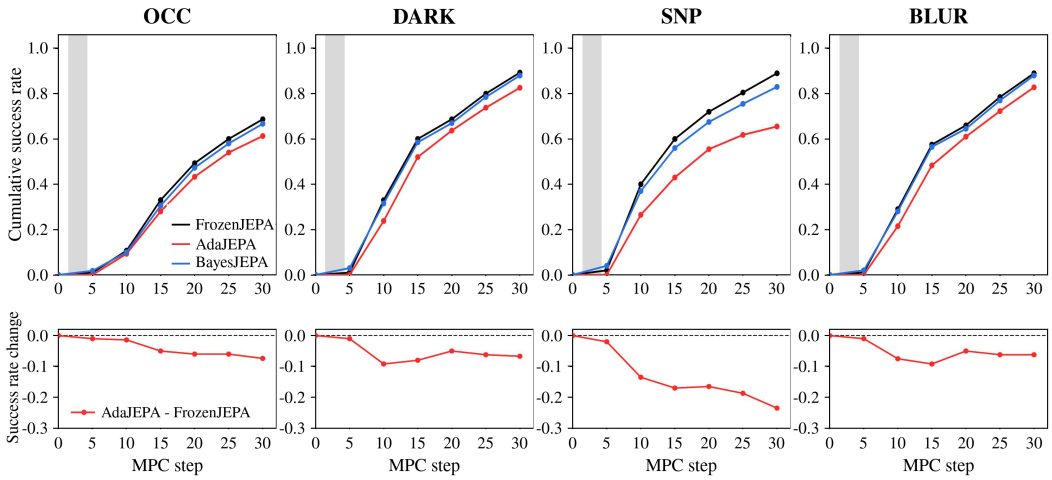

Figure 1: Transient observation corruption can leave a lasting performance gap under direct adaptation. The top row shows cumulative planning success under three-step occlusion, darkening, salt-and-pepper noise, and blur corruptions. Although the visual disturbance is brief, direct AdaJEPA (Wang et al., 2026a) remains below FrozenJEPA (Wang et al., 2026b) after the corruption has ended, producing a persistent post-corruption performance gap. BayesJEPA stays substantially closer to the frozen baseline across all four corruptions. The bottom row reports the success-rate difference between AdaJEPA and FrozenJEPA, highlighting the lasting deficit induced by direct adaptation.

The residual-attribution problem suggests a natural connection to Bayesian filtering, where uncertain predictions and observations are combined according to their relative reliability (Kalman, 1960; Revach et al., 2022). Following this principle, we introduce BayesJEPA: rather than learning from the full prediction–observation mismatch, it estimates prediction-side and observation-side uncertainty and uses their relative reliability to determine how much of the residual should be attributed to the prediction side. This leads to a simple rule: *filtering before adapting*. When the current observation is reliable, BayesJEPA remains close to direct adaptation and preserves responsiveness to prediction-side mismatch; when the observation is unreliable, its contribution to the learning signal is suppressed before it can drive a harmful update. Figure 2 illustrates this mechanism.

We formalize latent residual attribution as estimating the inaccessible clean-target prediction residual while conditioning on the predictor's actual input history. Under our probabilistic residual model, uncertainty-weighted filtering is the optimal estimate of this signal and induces the corresponding posterior-optimal estimate of the oracle adaptation gradient. This result motivates three empirical questions: does filtering better recover the clean-target adaptation signal when the current observation is unreliable; does improved attribution reduce harmful updates and lasting model degradation under transient corruption; and does BayesJEPA preserve useful adaptation when observations are reliable but the environment dynamics have shifted? We study these questions in PointMaze (Fu et al., 2020) under transient visual corruptions, persistent dynamics shifts, and their combinations, evaluating both the quality of the adaptation signal and its downstream effect on closed-loop planning.

Our main contributions are:

- Latent residual attribution and theory. We formulate test-time world-model adaptation as estimating the prediction-side residual from the prediction–observation mismatch, and derive a posterior-optimal uncertainty-weighted estimator with its oracle-gradient interpretation.
- BayesJEPA: filtering before adapting. We introduce a practical uncertainty-aware adaptation rule that filters the latent residual according to the relative reliability of prediction and observation before updating the world model.
- Theory-aligned empirical validation. We show that BayesJEPA better recovers the adaptation signal, reduces harmful updates and model degradation under transient visual corruption, and preserves useful adaptation under persistent dynamics mismatch and mixed shifts.

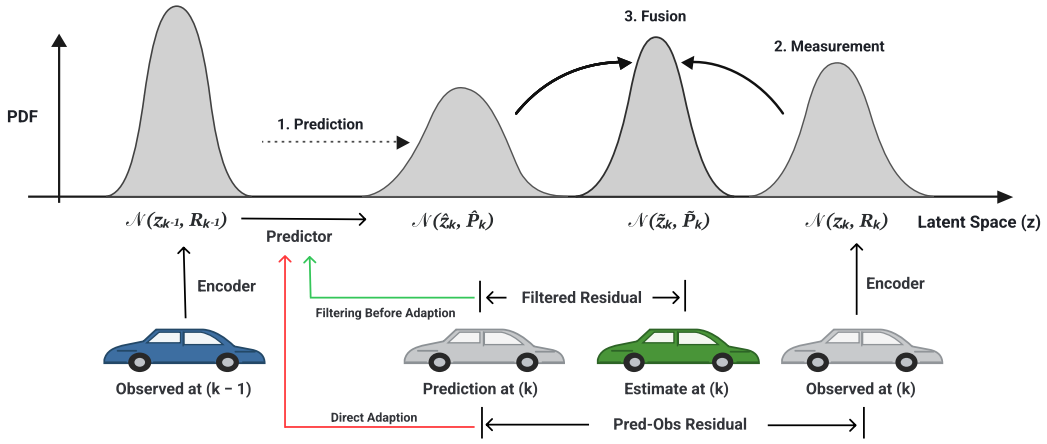

Figure 2: BayesJEPA filters before adapting. Direct adaptation (Wang et al., 2026a) treats the full prediction–observation discrepancy as a learning signal. BayesJEPA instead uses the relative uncertainty of prediction and observation to estimate the prediction-side component of that discrepancy, and adapts the world model using the filtered signal. Reliable observations preserve useful adaptation, whereas unreliable observations are prevented from dominating the update.

2 Problem Setup

We study test-time adaptation of a latent world model deployed in closed-loop planning. After each executed action, the model observes the resulting transition and may update its predictor before replanning. Under distribution shift, a prediction–observation discrepancy may arise because the prediction is inaccurate, the current observation is unreliable, or both. The central question is therefore *which part of the observed residual should drive adaptation*.

JEPA Latent World Models. Let $\mathcal{O}$ and $\mathcal{A}$ denote the observation and action spaces, with $o_k \in \mathcal{O}$ and $a_k \in \mathcal{A}$ the observation and action at interaction step k. A JEPA-style latent world model consists of an observation encoder E_ϕ, an action encoder A_ψ, and a latent predictor F_θ, parameterized by ϕ, ψ, and θ, respectively. We write $z_k = E_\phi(o_k)$ and $u_{k-1} = A_\psi(a_{k-1})$, with prediction $\hat{z}_k = F_\theta(z_{k-1}, u_{k-1})$. The same notation covers predictors conditioned on a short history of encoded observations and actions.

The predictor is trained with the squared latent prediction loss $\mathcal{L}_{\text{pred}} = \frac{1}{2}\|\hat{z}_k - \text{sg}(z_k)\|_2^2$, where $\text{sg}(\cdot)$ denotes stop-gradient. At deployment, the learned latent dynamics are rolled forward under candidate action sequences for goal-conditioned planning (Zhou et al., 2025; Wang et al., 2026a). Throughout this work, both encoders remain fixed and only the predictor parameters θ are adapted.

Direct Test-Time Adaptation. A frozen world model keeps θ fixed during deployment, whereas AdaJEPA updates the predictor using transitions encountered during closed-loop interaction (Wang et al., 2026a). For a newly observed transition, define the prediction–observation residual as $e_k := z_k - \hat{z}_k$. Under the corresponding single-transition loss $\mathcal{L}_{\text{dir},k} := \frac{1}{2}\|\hat{z}_k - \text{sg}(z_k)\|_2^2$, the latent gradient is $\nabla_{\hat{z}_k}\mathcal{L}_{\text{dir},k} = \hat{z}_k - z_k = -e_k$. Thus, direct adaptation uses the entire observed residual e_k as the adaptation signal. This is appropriate when z_k is a reliable target and the discrepancy reflects prediction error. When the observation is unreliable, however, the same update can adapt the predictor toward observation corruption. The residual e_k alone does not recognize two sources of error.

The Residual Attribution Problem. To formalize this ambiguity, we introduce a conceptual clean latent $z_k^\star$, defined as the representation that the fixed encoder would produce from a clean observation of the same realized environment state. This quantity is used only for analysis and evaluation and is unavailable at test time. Let $\mathcal{H}_{k-1}$ denote the information available before o_k arrives, including the observation and action history used by the predictor, its current parameters, and its adaptation state. Conditioning on $\mathcal{H}_{k-1}$ fixes the prediction $\hat{z}_k$ and its predictor Jacobian. We define the clean-target prediction residual $\delta_k := z_k^\star - \hat{z}_k$ and the current observation-side error $\varepsilon_k := z_k - z_k^\star$. Hence

$$e_k = z_k - \hat{z}_k = (z_k^\star - \hat{z}_k) + (z_k - z_k^\star) = \delta_k + \varepsilon_k. \tag{1}$$

This decomposition is local to the realized transition. Because $\hat{z}_k$ is produced from the actual predictor input/history, δ_k measures the correction from the prediction actually made to the clean current target. Thus, any effect already induced by corrupted source frames remains on the prediction side. In contrast, ε_k isolates the additional error introduced by the current observation.

If $z_k^\star$ were available, clean-target adaptation would use $\mathcal{L}_k^\star = \frac{1}{2}\|\hat{z}_k - z_k^\star\|_2^2$, whose negative latent gradient is δ_k. We therefore treat δ_k as the oracle adaptation signal while retaining the predictor input/history actually available at that transition.

At deployment, however, only $e_k = \delta_k + \varepsilon_k$ is available at test-time. We therefore formulate test-time adaptation as a residual attribution problem: estimate the inaccessible clean-target prediction residual δ_k from the observed residual before updating the predictor. Section 3 shows that relative prediction and observation uncertainty determine the optimal linear attribution, motivating the central principle of BayesJEPA: *filtering the residual before adapting*.

3 Theoretical Insight

Section 2 formulates test-time adaptation as estimating the clean-target prediction residual δ_k from the observed residual $e_k = \delta_k + \varepsilon_k$. We derive the optimal affine attribution rule under second-moment assumptions, show that it becomes the posterior mean and MMSE estimator under a Gaussian model, propagate it to oracle-gradient estimation, and compare its attribution risk with direct adaptation and freezing. Throughout, expectations and covariances are conditioned on the pre-observation history $\mathcal{H}_{k-1}$ unless stated otherwise.

3.1 Optimal Residual Attribution

From Equation 1, we first characterize the optimal estimator of δ_k from the observed residual e_k.

Assumption 1 (Conditional residual moments)**.** *The residual components have finite conditional second moments, with means $\mu_{\delta,k} := \mathbb{E}[\delta_k]$ and $\mu_{\varepsilon,k} := \mathbb{E}[\varepsilon_k]$, covariances $\hat{P}_k := \mathrm{Cov}(\delta_k)$ and $R_k := \mathrm{Cov}(\varepsilon_k)$, and zero centered cross-covariance $\mathrm{Cov}(\delta_k, \varepsilon_k) = 0$. Hence $\mu_{e,k} := \mathbb{E}[e_k] = \mu_{\delta,k} + \mu_{\varepsilon,k}$ and $S_k := \mathrm{Cov}(e_k) = \hat{P}_k + R_k \succ 0$.*

Let d be the latent dimension. For an attribution rule g, define the conditional squared-error risk $\mathcal{R}_k(g) := \mathbb{E}[\|\delta_k - g(e_k)\|_2^2 \mid \mathcal{H}_{k-1}]$. We first consider affine rules $g_{A,b}(e_k) = Ae_k + b$, where $A \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$.

Theorem 1 (Optimal affine residual attribution)**.** *Under Assumption 1, the unique minimum-risk affine rule is $g_k^\star(e_k) = \mu_{\delta,k} + K_k\big(e_k - \mu_{e,k}\big),\ K_k := \hat{P}_k(\hat{P}_k + R_k)^{-1} = \hat{P}_k S_k^{-1}$. Equivalently, the optimal affine coefficients are $A^\star = K_k$ and $b^\star = \mu_{\delta,k} - K_k\mu_{e,k}$.*

Corollary 1 (Gaussian posterior attribution)**.** *If, in addition, $(\delta_k, \varepsilon_k)$ is conditionally jointly Gaussian, then $\delta_k \mid e_k, \mathcal{H}_{k-1} \sim \mathcal{N}\Big(\mu_{\delta,k} + K_k(e_k - \mu_{e,k}), \tilde{P}_k\Big),\ \tilde{P}_k := \hat{P}_k - \hat{P}_k S_k^{-1}\hat{P}_k$. Therefore $g_k^\star(e_k) = \mathbb{E}[\delta_k \mid e_k, \mathcal{H}_{k-1}]$. Since the conditional mean minimizes mean-squared error over all measurable estimators, the optimal affine rule is also the* minimum mean-squared-error *(MMSE) estimator.*

Corollary 2 (Zero-mean uncertainty-weighted attribution)**.** *If $\mu_{\delta,k} = \mu_{\varepsilon,k} = 0$, then the optimal affine rule reduces to the linear uncertainty-weighted estimator $\rho_k^{\mathrm{Bayes}} = K_k e_k,\ K_k = \hat{P}_k(\hat{P}_k + R_k)^{-1}$.*

Under Assumption 1, $K_k e_k$ is the unique minimum-risk linear attribution rule. Under the additional Gaussian assumption of Corollary 1, it is also the posterior mean and unrestricted MMSE estimator.

The proofs are given in Appendix A.1. Theorem 1 shows that the covariance gain K_k is unchanged by nonzero means: the means contribute an affine correction, while relative uncertainty determines how the centered residual is weighted. The main BayesJEPA method uses the zero-mean specialization in Corollary 2; the nonzero-mean implementation is studied as an extension in Appendix A.5.

Under the Gaussian model, since $z_k^\star = \hat{z}_k + \delta_k$, the corresponding posterior estimate of the clean current latent is $\tilde{z}_k := \hat{z}_k + g_k^\star(e_k)$, which reduces in the zero-mean case to $\tilde{z}_k = \hat{z}_k + K_k e_k$.

3.2 From Residual Attribution to Oracle-Gradient Estimation

Residual attribution determines not only which latent discrepancy should be learned from, but also the parameter gradient used for adaptation. Using the same predictor input/history as the deployed model, define the inaccessible clean-target loss $\mathcal{L}_k^\star(\theta) := \frac{1}{2}\|F_\theta(z_{k-1}, u_{k-1}) - z_k^\star\|_2^2$. Let $J_k := \partial F_\theta(z_{k-1}, u_{k-1})/\partial\theta|_{\theta=\theta_k}$ denote the predictor Jacobian at the current parameters. Since $\delta_k = z_k^\star - \hat{z}_k$, the corresponding clean-target oracle gradient is $\gamma_k^\star := \nabla_\theta \mathcal{L}_k^\star(\theta_k) = -J_k^\top \delta_k$. Under the Gaussian setting of Corollary 1, the posterior attribution rule from Theorem 1 induces $\gamma_k^{\mathrm{post}} := -J_k^\top g_k^\star(e_k) = -J_k^\top \left[\mu_{\delta,k} + K_k(e_k - \mu_{e,k})\right]$.

Theorem 2 (Posterior-optimal oracle-gradient estimation)**.** *Under Corollary 1, $\gamma_k^{\mathrm{post}} = \mathbb{E}[\gamma_k^\star \mid e_k, \mathcal{H}_{k-1}]$. Consequently, γ_k^{post} is the minimum mean-squared-error (MMSE) estimator of the clean-target oracle gradient among all estimators measurable from the observed residual and history.*

Corollary 3 (Zero-mean BayesJEPA gradient)**.** *Under the zero-mean setting of Corollary 2, $\gamma_k^{\mathrm{Bayes}} = -J_k^\top K_k e_k$. If the residuals are additionally conditionally Gaussian, then $\gamma_k^{\mathrm{Bayes}} = \mathbb{E}[\gamma_k^\star \mid e_k, \mathcal{H}_{k-1}]$ and is therefore the unrestricted MMSE estimate of the clean-target oracle gradient.*

The proof is given in Appendix A.2. The oracle is a *clean-current-target* oracle: it replaces only the current target with $z_k^\star$ while retaining the predictor input/history actually available at that transition. Thus, any effect of corrupted source frames already present in the predictor input remains on the prediction side. The result concerns the gradient at the current parameters; optimizer momentum, replay, and repeated updates can alter the eventual parameter displacement.

3.3 Attribution Risk and Limiting Regimes

We next compare optimal residual attribution with two limiting deployment strategies: freezing, $g_{\mathrm{frz}}(e_k) = 0$, and direct adaptation, $g_{\mathrm{dir}}(e_k) = e_k$. Let $g_k^\star$ denote the optimal affine rule.

Theorem 3 (Attribution-risk dominance)**.** *Under Assumption 1, suppose $\hat{P}_k, R_k \succ 0$. The optimal affine attribution rule satisfies $\mathcal{R}_k(g_k^\star) = \mathrm{tr}(\tilde{P}_k)$, $\tilde{P}_k = (\hat{P}_k^{-1} + R_k^{-1})^{-1}$, while $\mathcal{R}_k(g_{\mathrm{frz}}) = \mathrm{tr}(\hat{P}_k) + \|\mu_{\delta,k}\|_2^2$ and $\mathcal{R}_k(g_{\mathrm{dir}}) = \mathrm{tr}(R_k) + \|\mu_{\varepsilon,k}\|_2^2$. Moreover, $\tilde{P}_k \prec \hat{P}_k$ and $\tilde{P}_k \prec R_k$, and hence $\mathcal{R}_k(g_k^\star) < \min\{\mathcal{R}_k(g_{\mathrm{frz}}), \mathcal{R}_k(g_{\mathrm{dir}})\}$.*

The proof is given in Appendix A.3. Here $\tilde{P}_k$ is the estimation-error covariance of the optimal affine rule; under the Gaussian model of Corollary 1, it also coincides with the posterior covariance. Appendix A.4 further characterizes the excess risk induced by approximate means and uncertainty estimates. These are residual-attribution guarantees under the local statistical model, not guarantees of closed-loop task performance.

4 Filtering Before Adapting Latent World Models

Section 3 shows that, under the centered residual model, $K_k e_k$ is the minimum-risk linear estimate of the clean-target prediction residual and, under Gaussian uncertainty, its posterior mean and MMSE estimate. BayesJEPA instantiates this principle online by estimating prediction- and observation-side uncertainty and adapting only to the filtered residual. The filter acts on the adaptation signal rather than the latent state used for planning, leaving the planning pipeline unchanged and isolating the effect of uncertainty-aware residual attribution. Figure 2 illustrates BayesJEPA pipeline.

4.1 Uncertainty-Weighted Residual Filtering

For transition (o_{k-1}, a_{k-1}, o_k), the fixed observation and action encoders produce $z_{k-1} = E_\phi(o_{k-1})$, $u_{k-1} = A_\psi(a_{k-1})$, and $z_k = E_\phi(o_k)$. Before the current update, the predictor with parameters θ_k gives $\hat{z}_k = F_{\theta_k}(z_{k-1}, u_{k-1})$. The observed residual is $e_k := z_k - \hat{z}_k \in \mathbb{R}^d$, where d is the latent dimension. As established in Section 2, $e_k = \delta_k + \varepsilon_k$ combines the clean-target prediction residual δ_k and current observation-side error ε_k.

Algorithm 1 BayesJEPA: filtering before adaptation

Require: Fixed E_ϕ, A_ψ, predictor F_θ, uncertainty estimators $\mathcal{U}_P, \mathcal{U}_R^{\text{cal}}$, learning rate η

for each transition (o_{k-1}, a_{k-1}, o_k) **do**
  Add the transition to replay and select $\mathcal{B}_k$
  for each $(o_{i-1}, a_{i-1}, o_i) \in \mathcal{B}_k$ **do**
    Encode z_{i-1}, z_i, u_{i-1} and compute $\hat{z}_i = F_\theta(z_{i-1}, u_{i-1})$, $e_i = z_i - \hat{z}_i$
    Estimate $p_i = \mathcal{U}_P(z_{i-1}, u_{i-1}; \theta)$ and $r_i = \mathcal{U}_R^{\text{cal}}(o_i)$
    Set $\kappa_i = p_i/(p_i + r_i + \epsilon_{\text{num}})$ and $t_i = \text{sg}(\hat{z}_i + \kappa_i e_i)$
  end for
  Update $\theta \leftarrow \theta - \eta \nabla_\theta |\mathcal{B}_k|^{-1} \sum_{i \in \mathcal{B}_k} \frac{1}{2} \|F_\theta(z_{i-1}, u_{i-1}) - t_i\|_2^2$
  Replan with the adapted predictor
end for

Let $\hat{P}_k, R_k \in \mathbb{R}^{d \times d}$ denote the prediction- and observation-side uncertainty quantities used by the filter. Following Corollary 2, BayesJEPA computes

$$K_k = \hat{P}_k(\hat{P}_k + R_k)^{-1}, \qquad \rho_k = K_k e_k, \qquad \tilde{z}_k = \hat{z}_k + \rho_k, \tag{2}$$

where K_k is the uncertainty-weighting gain, ρ_k the filtered adaptation residual, and $\tilde{z}_k$ the corresponding adaptation target. Under the ideal zero-mean Gaussian model, $\rho_k = \mathbb{E}[\delta_k \mid e_k, \mathcal{H}_{k-1}]$; with estimated uncertainties, Equation 2 is a plug-in approximation to this posterior rule.

Our default implementation is isotropic, with $\hat{P}_k = p_k I_d$ and $R_k = r_k I_d$, where I_d is the $d \times d$ identity and $p_k, r_k > 0$ are scalar uncertainty estimates. The stabilized gain is therefore $K_k = \kappa_k I_d$, where $\kappa_k = p_k/(p_k + r_k + \epsilon_{\text{num}})$, $\epsilon_{\text{num}} > 0$, and $\rho_k = \kappa_k e_k$. Ignoring the numerical stabilizer, the gain depends only on relative uncertainty: as $r_k \to 0$, $\kappa_k \to 1$ and BayesJEPA approaches direct adaptation, whereas $r_k/p_k \to \infty$ gives $\kappa_k \to 0$ and suppresses adaptation to an unreliable current observation. Thus filtering is transition dependent rather than a fixed reduction of every update. A coordinate-wise diagonal variant is described in Appendix B.1.

4.2 Filtering Before Adaptation

BayesJEPA uses $\tilde{z}_k$ only as a detached adaptation target: $\mathcal{L}_{\text{Bayes},k} = \frac{1}{2}\|\hat{z}_k - \text{sg}(\tilde{z}_k)\|_2^2$, $\nabla_\theta \mathcal{L}_{\text{Bayes},k} = -J_k^\top \rho_k$, where $\text{sg}(\cdot)$ denotes stop-gradient and $J_k = \partial \hat{z}_k/\partial\theta$ is evaluated at θ_k. Equivalently, $\nabla_{\hat{z}_k} \mathcal{L}_{\text{Bayes},k} = -\rho_k$. Under the ideal zero-mean Gaussian model, Corollary 3 identifies $-J_k^\top \rho_k$ as the posterior mean and MMSE estimate of the clean-target oracle gradient. Practical BayesJEPA instead uses learned uncertainty estimates and is therefore a plug-in approximation to this ideal rule.

The filtered target is used only for adaptation: it neither replaces the latent observation used by the planner nor introduces a recursive filtered state. Only the predictor parameters are adapted; both encoders and the uncertainty estimators remain fixed.

Let $\mathcal{B}_k$ denote the transitions selected at interaction step k by the same replay rule used for direct adaptation. Each selected transition is re-evaluated under the current predictor, its filtered target is recomputed, and the predictor is updated using the mean BayesJEPA loss over $\mathcal{B}_k$. We denote the calibrated prediction- and observation-uncertainty estimators by $\mathcal{U}_P$ and $\mathcal{U}_R^{\text{cal}}$, respectively. Algorithm 1 summarizes the scalar implementation; replay and calibration details are in Appendix B.

4.3 Practical Uncertainty Estimation

Prediction uncertainty. The nominal prediction $\hat{z}_k$ is deterministic, with dropout disabled. BayesJEPA uses $M > 1$ MC-dropout predictions $\{\hat{z}_k^{(m)}\}_{m=1}^M$ only to estimate the per-dimension variance $v_k^P = [d(M-1)]^{-1} \sum_m \|\hat{z}_k^{(m)} - \bar{z}_k\|_2^2$, where $\bar{z}_k = M^{-1} \sum_m \hat{z}_k^{(m)}$ (Gal & Ghahramani, 2016). Held-out calibration gives $p_k = \alpha_P v_k^P + \beta_P$, with $\alpha_P \geq 0$, $\beta_P > 0$, and we set $\hat{P}_k = p_k I_d$.

Observation uncertainty. A frozen reliability network $\mathcal{U}_R$, trained offline on paired clean and perturbed observations, estimates current-target latent error. Its calibrated output is $r_k = \alpha_R[\text{softplus}(\mathcal{U}_R(o_k)) + r_{\min}] + \beta_R$, with $r_{\min} > 0$, $\alpha_R \geq 0$, and $\beta_R > 0$, and we set $R_k = r_k I_d$.

At deployment, it receives only o_k, with no corruption type, severity, timing, indicator, or paired clean observation. Training and calibration details are given in Appendix B.1.

The resulting p_k and r_k are calibrated second-moment proxies rather than exact conditional covariances, so BayesJEPA implements a learned plug-in approximation to the ideal posterior rule. Diagnostics are reported in Appendix B.6, and the nonzero-mean extension in Appendix A.5.

5 EXPERIMENTS

We evaluate our BayesJEPA on PointMaze (Fu et al., 2020), largely following the evaluation protocol of AdaJEPA (Wang et al., 2026a), which serves as our primary direct adaptation baseline. Planning and test-time adaptation settings are matched across methods wherever applicable, allowing us to isolate the effect of residual filtering. We organize the experiments around three questions: (1) Does uncertainty-weighted filtering improve the adaptation signal? (2) Does it reduce harmful adaptation under transient visual corruption? (3) Does it preserve useful adaptation under persistent model shift when observations remain reliable?

5.1 EXPERIMENTAL SETUP

Protocol and adaptation. We evaluate on D4RL PointMaze (Fu et al., 2020), largely following AdaJEPA (Wang et al., 2026a). We use the Medium-Maze world model with the pretrained Temporal Straightening representation (Wang et al., 2026b): a fixed 384-dimensional visual encoder and an action-conditioned predictor with three-frame history and frameskip five. Following AdaJEPA, start–goal pairs have Euclidean separation greater than three cell units, and each condition uses 50 episodes per seed across three seeds. All adaptive methods start from the same checkpoint, update only the predictor, and use one SGD step per MPC replanning step with learning rate 5×10^{-4} and a recent-5 replay buffer. AdaJEPA adapts to $e_k = z_k - \hat{z}_k$, whereas BayesJEPA uses $\rho_k = \kappa_k e_k$ with $\kappa_k = p_k/(p_k + r_k + \varepsilon)$; p_k is estimated by MC dropout (Gal & Ghahramani, 2016) and r_k by an offline-trained reliability network. Estimator training and calibration are detailed in Appendix B.1.

Shifts, baselines, and metrics. We study transient visual corruption (OCC, with SNP, DARK, and BLUR as additional conditions), persistent dynamics shifts (mass $\times 0.2$ and damping $\times 20$), and their combinations; representative observations are shown in Appendix B.2). We compare FROZENJEPA (Wang et al., 2026b), ADAJEPA (Wang et al., 2026a), and BAYESJEPA; diagonal and mean-aware variants are evaluated in Section 5.5. For residual-attribution analysis, the evaluator renders a clean current observation $o_k^\star$ at the same simulator state and forms $\delta_k^\star = E_\phi(o_k^\star) - \hat{z}_k$ while retaining the actual predictor history. We report per-dimension residual MSE $d^{-1}\|\rho_k - \delta_k^\star\|_2^2$ and closed-loop Success@T, with Success@30 as the primary planning metric.

5.2 DOES FILTERING IMPROVE THE ADAPTATION SIGNAL?

We first isolate adaptation-signal quality from downstream planning. On held-out transitions, we fix the predictor history, parameters, and prediction $\hat{z}_k$, and render a clean counterpart observation $o_k^\star$ at the same simulator state. This defines the evaluator-only reference residual $\delta_k^\star = E_\phi(o_k^\star) - \hat{z}_k$ while retaining the actual predictor inputs. We compare AdaJEPA's raw residual $e_k = z_k - \hat{z}_k$ and BayesJEPA's filtered residual $\rho_k = \kappa_k e_k$ using their per-dimension squared errors relative to $\delta_k^\star$.

Figure 3(a) reports the mean residual errors over 202 held-out transitions from 10 trajectories per condition. BayesJEPA reduces residual MSE by 83.7% under occlusion, 84.3% under salt-and-pepper noise, 61.9% under darkening, and 43.4% under blur. The improvement is strongest when visual corruption contributes substantially to the observed residual, while blur yields a smaller gain because its absolute latent error is already small. Overall, the results indicate that uncertainty-weighted filtering better isolates the component of the residual that should drive adaptation.

5.3 DOES FILTERING REDUCE HARMFUL ADAPTATION TO TRANSIENT CORRUPTION?

We next examine whether improved signal quality reduces harmful updates under transient corruption. With dynamics fixed, each episode begins with clean observations, receives a contiguous occlusion

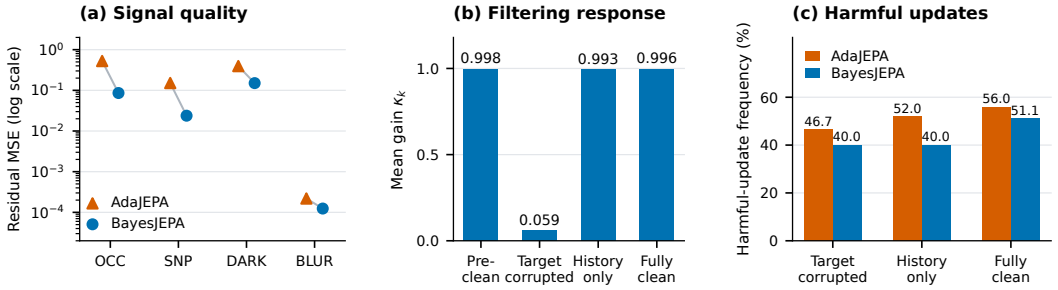

Figure 3: Filtering improves adaptation signals and reduces harmful updates. (a) Per-dimension residual MSE relative to the evaluator-only reference; lower is better. (b) Mean BayesJEPA gain and (c) harmful-update frequency during a three-step occlusion burst and recovery. "History only" indicates a clean current target with contaminated predictor history.

burst of three MPC steps, and then returns to clean observations. Corruption affects the observations used for both planning and adaptation. We distinguish transitions with a corrupted current target, a clean target but contaminated predictor history, and a clean target and history.

For each actual adaptation step, we evaluate the squared prediction loss $L_k^\star(\theta)$ on the realized transition using a clean current target and the actual predictor inputs. An update is harmful when $\Delta L_k^\star = L_k^\star(\theta_{k+1}) - L_k^\star(\theta_k) > 0$, where θ_k and θ_{k+1} are the parameters immediately before and after the step. The inputs and clean target remain fixed across both loss evaluations. Harmful-update frequency is the fraction of valid adaptation steps satisfying this condition.

Figure 3(b) shows that the mean BayesJEPA gain falls from 0.998 before occlusion to 0.059 during the burst. It returns to 0.993 once the current target becomes clean, even while predictor history remains contaminated. This response is consistent with estimating current-target reliability without explicitly tracking history corruption. BayesJEPA exhibits lower harmful-update frequency in all three phases (Figure 3(c)). The reduction is most directly attributable to filtering during target corruption, when the gain drops sharply. The advantage persists after the current observation becomes clean despite the gain returning near one, consistent with less harmful adaptation accumulated during the corruption interval. Supporting this interpretation, AdaJEPA retains substantially larger parameter drift from the pre-burst predictor even after both target and history are clean (Appendix B.5), showing that transient corruption can leave lasting parameter changes under direct adaptation.

The downstream results in Table 1 are aligned with this pattern: BayesJEPA achieves 66.7% Success@30, compared with 61.3% (AdaJEPA) and 68.7% (FrozenJEPA). Together, these results show fewer harmful updates, reduced lasting parameter drift, and better planning under transient occlusion. Additional uncertainty and residual-norm diagnostics appear in Appendix B.5.

5.4 Does Filtering Preserve Useful Adaptation under Persistent Model Shift?

The previous experiment shows that filtering reduces harmful adaptation to transient corruption. We now test the complementary requirement: filtering should preserve adaptation when observations are reliable but the model is persistently mismatched. We therefore keep observations clean and modify the environment dynamics throughout the episode using low mass ($\times 0.2$) and high damping ($\times 20$), without providing BayesJEPA any shift label.

BayesJEPA maintains mean gains of 0.93 under low mass and 0.88 under high damping, so most of the observed residual continues to drive adaptation. Consistent with this, BayesJEPA nearly matches AdaJEPA under persistent-only shifts: Success@30 is 62.7% versus 63.3% under low mass and 42.7% versus 43.3% under high damping. Both adaptive methods improve over FrozenJEPA, indicating that filtering preserves useful adaptation when the mismatch is persistent.

We next combine each dynamics shift with intermittent OCC. This setting tests selective adaptation: the same method must retain updates driven by model mismatch while suppressing unreliable observation-side residuals, without receiving a condition switch or corruption indicator. Under mass$\times 0.2$+OCC, BayesJEPA improves Success@30 from 50.7% to 59.3% over AdaJEPA; un-

Table 1: **PointMaze planning under transient, persistent, and mixed shifts.** Success@30 (%) over 50 episodes per seed across three seeds. All methods use matched start–goal pairs and identical planning and adaptation budgets. BayesJEPA uses the same uncertainty estimators in every condition.

	Transient	Persistent		Mixed	
Method	Occlusion	Mass $\times 0.2$	Damping $\times 20$	Mass $\times 0.2$ + OCC	Damping $\times 20$ + OCC
FrozenJEPA	68.7	58.7	40.7	57.3	20.7
AdaJEPA	61.3	63.3	43.3	50.7	22.0
BayesJEPA	66.7 (↑ 5.4)	62.7 (↓ 0.6)	42.7 (↓ 0.6)	59.3 (↑ 8.6)	29.3 (↑ 7.3)

Table 2: **Ablations of the BayesJEPA filtering recipe.** (a) Default scalar, diagonal, and mean-aware filtering. Mixed Success@30 averages the two mixed-shift conditions; residual MSE is averaged over the corresponding residual-attribution conditions. (b) Observation-error decomposition on paired held-out observations. Bias fraction is the squared empirical mean divided by total latent error.

(a) Filtering variants

Methods	OCC S@30 ↑	Mixed S@30 ↑	Residual MSE ↓
BayesJEPA-D	64.7	40.7	0.384
BayesJEPA-MA	66.0	**45.3**	0.340
BayesJEPA	**66.7**	44.3	**0.340**

(b) Observation-error bias

Condition	Total error	Bias fraction
OCC	7.900	24.9%
SNP	0.946	7.6%
DARK	5.076	6.3%
BLUR	0.00162	23.3%

der damping$\times$20+OCC, it improves from 22.0% to 29.3%. Importantly, BayesJEPA maintains high gains on reliable transitions while reducing them when the current observation is corrupted, showing transition-dependent rather than uniformly conservative adaptation (Appendix B.7). Full Success@10/20/30 profiles and gain statistics are reported there, with a representative mixed-shift rollout shown in Appendix B.8.

5.5 Ablations

We ablate the two simplifications used in the default BayesJEPA recipe: a scalar rather than coordinate-wise gain, and a centered rather than mean-aware residual model.

Is coordinate-wise filtering necessary? BayesJEPA uses the scalar gain $K_k = \kappa_k I$. We compare it with BayesJEPA-D, which estimates a separate gain for each latent coordinate. Table 2(a) shows no benefit from diagonal filtering: the scalar rule achieves higher OCC and mixed-shift Success@30 and lower residual MSE. This suggests that the useful uncertainty variation is largely shared across latent dimensions in our setting, while coordinate-wise estimation adds unnecessary complexity. We therefore use the scalar rule by default.

Is mean-aware filtering necessary? The default method uses the centered rule $\rho_k = K_k e_k$. We compare it with BayesJEPA-MA, which additionally tracks a running residual mean and applies $\rho_k = m_k + K_k(e_k - m_k)$. Mean-aware filtering gives a small improvement in mixed-shift Success@30 (45.3% versus 44.3%), but slightly lower OCC performance and essentially unchanged residual MSE. Table 2(b) helps explain the limited benefit: centered variation accounts for most observation-side error across all four corruptions. We therefore retain the centered rule in the main method and treat mean tracking as an optional extension.

6 Conclusion

We formulate test-time adaptation of latent world models as a residual-attribution problem and introduce BayesJEPA, which filters prediction–observation residuals according to relative uncertainty before adaptation. Under our probabilistic model, the resulting signal is optimal for clean-target residual and oracle-gradient estimation. Experiments show that BayesJEPA improves adaptation-signal quality, reduces harmful updates under transient corruption, and preserves useful adaptation under persistent and mixed shifts. Overall, our results support *filtering before adapting* as a principled approach to robust test-time adaptation.

AI USE STATEMENT

In this work, we used generative AI tools to edit and polish the manuscript for clarity and concision, and to assist in checking mathematical derivations and proofs. We did not use generative AI to develop the conceptual framework, formulate mathematical claims, design or implement the methodology or experiments, analyze data, or interpret results. All AI-assisted content was reviewed and independently verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG Van Sloun, and Yonina C Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 43905–43941. Curran Associates, Inc., 2025. doi: 10.52202/085713-1465.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. Adajepa: An adaptive latent world model. *arXiv preprint arXiv:2606.32026*, 2026a.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=Ik1mKtUYlZ.

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

A Additional Theory

This section provides proofs and extensions for Section 3. Unless stated otherwise, all expectations and covariances are conditioned on the pre-observation history $\mathcal{H}_{k-1}$. Under this conditioning, the current predictor parameters, inputs, nominal prediction, and Jacobian are fixed.

A.1 Proof of Optimal Residual Attribution

We prove Theorem 1 and Corollaries 1–2.

For an affine rule $g_{A,b}(e_k) = Ae_k + b$, define $\bar{\delta}_k := \delta_k - \mu_{\delta,k}$ and $\bar{e}_k := e_k - \mu_{e,k}$. Under Assumption 1,

$$\mathbb{E}[\bar{\delta}_k \bar{e}_k^\top] = \hat{P}_k, \qquad \mathbb{E}[\bar{e}_k \bar{e}_k^\top] = S_k.$$

The estimation error is

$$\delta_k - g_{A,b}(e_k) = \bar{\delta}_k - A\bar{e}_k + \big(\mu_{\delta,k} - A\mu_{e,k} - b\big).$$

Since the centered term has zero mean,

$$\mathcal{R}_k(g_{A,b}) = \operatorname{tr}\Big(\hat{P}_k - A\hat{P}_k - \hat{P}_k A^\top + AS_kA^\top\Big) + \|\mu_{\delta,k} - A\mu_{e,k} - b\|_2^2 \tag{3}$$

$$= \operatorname{tr}\Big(\hat{P}_k - \hat{P}_k S_k^{-1}\hat{P}_k\Big) + \operatorname{tr}\big[(A - K_k)S_k(A - K_k)^\top\big] + \|\mu_{\delta,k} - A\mu_{e,k} - b\|_2^2, \tag{4}$$

where $K_k = \hat{P}_k S_k^{-1}$. Since $S_k \succ 0$, the second term is nonnegative and vanishes uniquely at $A = K_k$. For this choice, the final term vanishes uniquely at $b = \mu_{\delta,k} - K_k\mu_{e,k}$. Hence the unique minimum-risk affine rule is

$$g_k^\star(e_k) = \mu_{\delta,k} + K_k(e_k - \mu_{e,k}),$$

which proves Theorem 1.

For Corollary 1, suppose additionally that $(\delta_k, \varepsilon_k)$ is conditionally jointly Gaussian. Since $e_k = \delta_k + \varepsilon_k$, the pair (δ_k, e_k) is jointly Gaussian with $\operatorname{Cov}(\delta_k, e_k) = \hat{P}_k$ and $\operatorname{Cov}(e_k) = S_k$. Gaussian conditioning gives

$$\delta_k \mid e_k, \mathcal{H}_{k-1} \sim \mathcal{N}\Big(\mu_{\delta,k} + \hat{P}_k S_k^{-1}(e_k - \mu_{e,k}), \hat{P}_k - \hat{P}_k S_k^{-1}\hat{P}_k\Big). \tag{5}$$

Therefore

$$\mathbb{E}[\delta_k \mid e_k, \mathcal{H}_{k-1}] = g_k^\star(e_k).$$

Because conditional expectation minimizes mean-squared error over all measurable estimators, $g_k^\star$ is also the unrestricted MMSE estimator. If $\mu_{\delta,k} = \mu_{\varepsilon,k} = 0$, then $\mu_{e,k} = 0$ and $g_k^\star(e_k) = K_k e_k$, proving Corollary 2.

A.2 Proof of Posterior-Optimal Oracle-Gradient Estimation

Conditioning on $\mathcal{H}_{k-1}$ fixes the predictor Jacobian J_k. Under the Gaussian setting of Corollary 1,

$$\mathbb{E}[\gamma_k^\star \mid e_k, \mathcal{H}_{k-1}] = -J_k^\top \mathbb{E}[\delta_k \mid e_k, \mathcal{H}_{k-1}] \tag{6}$$

$$= -J_k^\top \left[\mu_{\delta,k} + K_k(e_k - \mu_{e,k})\right] = \gamma_k^{\text{post}}. \tag{7}$$

Thus γ_k^{post} is the posterior mean of the clean-target oracle gradient and therefore its MMSE estimator. In the zero-mean setting, this reduces to

$$\gamma_k^{\text{Bayes}} = -J_k^\top K_k e_k.$$

The oracle is local to the realized transition: it replaces only the current target by its clean counterpart while retaining the predictor input/history available at deployment. Effects already induced by corrupted source-history frames therefore remain on the prediction side.

A.3 Proof of the Attribution-Risk Comparison

For freezing, $g_{\mathrm{frz}}(e_k) = 0$, so

$$\mathcal{R}_k(g_{\mathrm{frz}}) = \mathbb{E}\|\delta_k\|_2^2 = \mathrm{tr}(\hat{P}_k) + \|\mu_{\delta,k}\|_2^2.$$

For direct adaptation, $g_{\mathrm{dir}}(e_k) = e_k$ and $\delta_k - e_k = -\varepsilon_k$, hence

$$\mathcal{R}_k(g_{\mathrm{dir}}) = \mathrm{tr}(R_k) + \|\mu_{\varepsilon,k}\|_2^2.$$

For the optimal affine rule, the estimation error has zero mean and covariance

$$\tilde{P}_k = \hat{P}_k - \hat{P}_k S_k^{-1} \hat{P}_k.$$

If $\hat{P}_k, R_k \succ 0$, the parallel-sum identity gives

$$\tilde{P}_k = (\hat{P}_k^{-1} + R_k^{-1})^{-1}, \qquad 0 \prec \tilde{P}_k \prec \hat{P}_k, \qquad 0 \prec \tilde{P}_k \prec R_k. \tag{8}$$

Therefore

$$\mathcal{R}_k(g_k^\star) = \mathrm{tr}(\tilde{P}_k) < \min\{\mathcal{R}_k(g_{\mathrm{frz}}), \mathcal{R}_k(g_{\mathrm{dir}})\},$$

which proves Theorem 3. In the scalar zero-mean case, $\mathcal{R}_k(g_k^\star) = p_k r_k/(p_k + r_k)$.

A.4 Sensitivity to Approximate Moments

The main theory assumes exact conditional moments. Let $\bar{\mu}_{\delta,k}$, $\bar{\mu}_{\varepsilon,k}$, and $\bar{K}_k$ be approximate quantities, and define

$$\Delta_{\delta,k} := \mu_{\delta,k} - \bar{\mu}_{\delta,k}, \qquad \Delta_{\varepsilon,k} := \mu_{\varepsilon,k} - \bar{\mu}_{\varepsilon,k}.$$

For the plug-in affine rule

$$\bar{g}(e_k) = \bar{\mu}_{\delta,k} + \bar{K}_k\big(e_k - \bar{\mu}_{\delta,k} - \bar{\mu}_{\varepsilon,k}\big),$$

Equation 4 yields

$$\begin{aligned}\mathcal{R}_k(\bar{g}) - \mathcal{R}_k(g_k^\star) = \ & \mathrm{tr}\big[(\bar{K}_k - K_k) S_k (\bar{K}_k - K_k)^\top\big] \\ & + \big\|\Delta_{\delta,k} - \bar{K}_k(\Delta_{\delta,k} + \Delta_{\varepsilon,k})\big\|_2^2 .\end{aligned} \tag{9}$$

Thus gain error and mean-estimation error contribute separate nonnegative penalties.

In the zero-mean case, if $\bar{K}_k$ is fixed under the stated history conditioning,

$$\mathcal{R}_k(g_{\bar{K}_k}) - \mathcal{R}_k(g_{K_k}) = \mathrm{tr}\big[(\bar{K}_k - K_k) S_k (\bar{K}_k - K_k)^\top\big] = \|(\bar{K}_k - K_k) S_k^{1/2}\|_F^2. \tag{10}$$

This identity gives a quadratic penalty around the ideal gain. A learned gain predicted from the current observation can be statistically coupled to the current residual, so the fixed-coefficient identity is not asserted to hold exactly for the deployable estimator.

A.5 Nonzero-Mean Affine Extension

The optimal affine rule from Theorem 1 is

$$g_k^\star(e_k) = \mu_{\delta,k} + K_k(e_k - \mu_{e,k}), \qquad \mu_{e,k} = \mu_{\delta,k} + \mu_{\varepsilon,k}.$$

The main BayesJEPA method uses the centered specialization $\rho_k = K_k e_k$. A practical mean-aware extension maintains a causal estimate m_k of the prediction-side mean and uses

$$\rho_k^{\mathrm{aff}} = m_k + K_k\big(e_k - m_k - \hat{\mu}_{\varepsilon,k}\big). \tag{11}$$

Our implementation sets $\hat{\mu}_{\varepsilon,k} = 0$ and updates m_k only from past transitions, for example

$$m_{k+1} = (1 - \lambda) m_k + \lambda\ \mathrm{sg}(\rho_k^{\mathrm{aff}}).$$

Thus the current observation does not directly determine the prior mean used for its own attribution. The ablation in the main paper shows only limited benefit from explicit mean tracking, so the centered rule remains the default.

B ADDITIONAL EXPERIMENTS

This section provides implementation details, evaluation definitions, and supporting analyses for Section 5. Unless stated otherwise, all methods use the same PointMaze environment, matched start–goal pairs, predictor-only adaptation, and matched optimization and planning budgets as described in Section 5.1. Here d denotes the latent dimension and $o_k^\star$ denotes an evaluator-only clean rendering of the realized state, with $z_k^\star = E_\phi(o_k^\star)$.

B.1 IMPLEMENTATION AND UNCERTAINTY ESTIMATION

Data splits. Source trajectories are partitioned into disjoint training, calibration, and test splits before observation pairs are constructed. The training split is used to fit the observation-reliability model $\mathcal{U}_R$, a separate calibration split is used to fit global output-scale parameters for prediction and observation uncertainty, and the test split contains all reported closed-loop evaluation episodes and held-out diagnostic trajectories. All renderings and perturbations derived from one source trajectory remain in the same split.

Prediction uncertainty. All methods compute the nominal prediction deterministically with dropout disabled. BayesJEPA separately obtains $M > 1$ stochastic MC-dropout predictions $\{\hat{z}_k^{(m)}\}_{m=1}^M$ from the same predictor input/history and computes

$$\bar{z}_k = \frac{1}{M}\sum_{m=1}^{M} \hat{z}_k^{(m)}, \qquad v_k^P = \frac{1}{d(M-1)}\sum_{m=1}^{M} \|\hat{z}_k^{(m)} - \bar{z}_k\|_2^2.$$

A held-out affine calibration maps this score to $p_k = \alpha_P v_k^P + \beta_P$, with $\alpha_P \geq 0$ and $\beta_P > 0$, and the default scalar implementation sets $\hat{P}_k = p_k I$. Calibration targets the clean-target prediction-error second moment on clean calibration transitions. This is used as a practical uncertainty-scale proxy and is not asserted to equal the exact conditional covariance when the clean-target residual has nonzero mean.

Observation uncertainty. For each offline pair $(o_i^\star, o_i^{\mathrm{pert}})$, define

$$y_i = \frac{1}{d}\left\|E_\phi(o_i^{\mathrm{pert}}) - E_\phi(o_i^\star)\right\|_2^2. \tag{12}$$

The reliability network receives only the perturbed observation and outputs

$$r_i^{\mathrm{raw}} = \mathrm{softplus}\big(\mathcal{U}_R(o_i^{\mathrm{pert}})\big) + r_{\min}, \qquad r_{\min} > 0.$$

It is trained with

$$\mathcal{L}_R^{\mathrm{train}} = \frac{1}{2|\mathcal{D}_{\mathrm{train}}|}\sum_{i\in\mathcal{D}_{\mathrm{train}}}\left[\frac{y_i}{r_i^{\mathrm{raw}}} + \log r_i^{\mathrm{raw}}\right]. \tag{13}$$

Clean examples are included by setting $o_i^{\mathrm{pert}} = o_i^\star$.

After training, $\mathcal{U}_R$ is frozen and a global calibration

$$r_i = \alpha_R r_i^{\mathrm{raw}} + \beta_R, \qquad \alpha_R \geq 0,\ \beta_R > 0,$$

is fitted on the calibration split using the same loss. At deployment,

$$r_k = \alpha_R\left[\mathrm{softplus}(\mathcal{U}_R(o_k)) + r_{\min}\right] + \beta_R, \qquad R_k = r_k I.$$

The estimator receives no corruption type, severity, timing, indicator, or paired clean observation. Since y_i is a latent-error second moment, the learned r_k is interpreted as a practical error-scale proxy rather than an exact covariance when the signed observation error has nonzero mean.

Scalar and diagonal gains. The default scalar gain is

$$\kappa_k = \frac{p_k}{p_k + r_k + \epsilon_{\mathrm{num}}}, \qquad K_k = \kappa_k I.$$

For BAYESJEPA-D, prediction and observation scales are estimated per latent coordinate, giving

$$[K_k]_{jj} = \frac{[\hat{P}_k]_{jj}}{[\hat{P}_k]_{jj} + [R_k]_{jj} + \epsilon_{\mathrm{num}}}. \tag{14}$$

No condition-specific parameters are used.



Figure 4: **Representative observation corruptions used in PointMaze.** From left to right: clean observation, occlusion, darkening, salt-and-pepper noise, and blur.

Table 3: **Local update harm and parameter drift under transient occlusion.** Harmful-update frequency is the percentage of valid adaptation steps. Drift is measured relative to the pre-burst predictor.

	f_H (%)		Drift	
Phase	AdaJEPA	BayesJEPA	AdaJEPA	BayesJEPA
Target corrupted	46.7	40.0	6.98×10^{-4}	7.53×10^{-5}
History only	52.0	40.0	1.80×10^{-3}	1.49×10^{-4}
Fully clean	56.0	51.1	3.90×10^{-3}	4.58×10^{-4}

B.2 Representative Observation Corruptions

Figure 4 shows representative observations under the visual corruptions used in our experiments. We consider occlusion (OCC), darkening (DARK), salt-and-pepper noise (SNP), and blur (BLUR), together with the corresponding clean observation. These perturbations affect the observation available to both planning and adaptation. At deployment, BayesJEPA receives only the current observation and is not provided with the corruption type, severity, timing, or an explicit corruption indicator.

B.3 Replay and Closed-Loop Adaptation

Let $\mathcal{B}_k$ denote the transitions selected at interaction step k by the same replay rule used for direct adaptation. Stored transitions, rather than previously constructed filtered targets, are replayed. For each selected transition, BayesJEPA recomputes the nominal prediction, residual, uncertainty estimates, gain, and detached filtered target under the current predictor parameters. The predictor is then updated using the mean BayesJEPA loss over $\mathcal{B}_k$, after which the agent replans. Only the predictor parameters are updated; the observation encoder, action encoder, and uncertainty estimators remain fixed throughout deployment.

B.4 Evaluation Metrics

Clean-target residual. For evaluator-only analysis, a clean current observation $o_k^\star$ is rendered at the same simulator state while retaining the predictor input/history actually encountered during deployment. This defines

$$\delta_k^\star = E_\phi(o_k^\star) - \hat{z}_k.$$

The residual-attribution error is

$$\mathcal{E}_{\text{res}} = d^{-1}\|\rho_k - \delta_k^\star\|_2^2,$$

with ρ_k replaced by the raw residual e_k for AdaJEPA.

Harmful updates. For each actual adaptation step, let $L_k^\star(\theta)$ denote the squared prediction loss on the realized transition with only the current target rendered cleanly. Predictor inputs and the clean

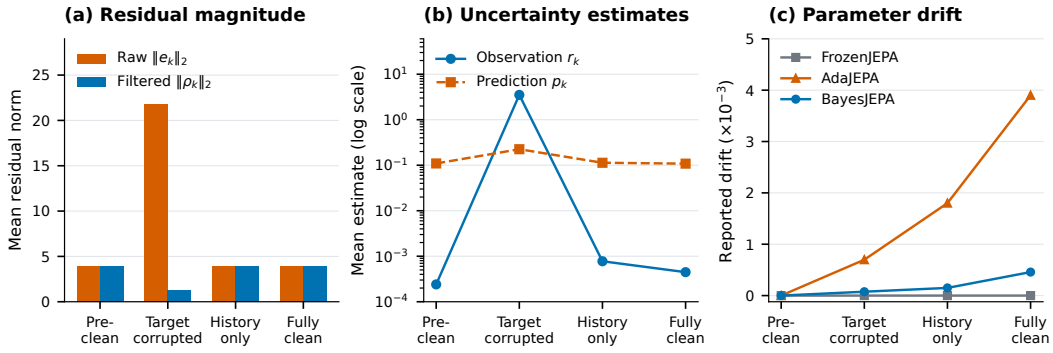

Figure 5: **Additional diagnostics across the occlusion and recovery phases.** (a) Mean raw and filtered residual norms for BayesJEPA. (b) Mean observation and prediction uncertainty on a logarithmic scale. (c) Parameter drift relative to the pre-burst predictor. "History only" denotes a clean current target with contaminated predictor history.

target are fixed across the before/after evaluations. Define

$$\Delta L_k^\star = L_k^\star(\theta_{k+1}) - L_k^\star(\theta_k), \qquad f_H = \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} \mathbf{1}\{\Delta L_k^\star > 0\}, \tag{15}$$

where $\mathcal{I}$ is the set of valid adaptive updates. An update is called harmful when $\Delta L_k^\star > 0$.

Parameter drift. Let θ_{pre} denote the predictor immediately before the corruption burst. We measure accumulated parameter change as

$$D_\theta(k) = \|\theta_k - \theta_{\text{pre}}\|_2.$$

Parameter drift measures displacement from the pre-burst predictor and does not by itself imply improved or degraded prediction quality.

B.5 Transient-Corruption Diagnostics

This subsection supports Section 5.3 and Figure 3. The transient intervention is a three-step occlusion burst with fixed dynamics.

For evaluator-only stratification, we distinguish three post-onset phases: *target corrupted*, where the current observation is corrupted; *history only*, where the current target is clean but at least one source frame used by the predictor is corrupted; and *fully clean*, where both the current target and predictor history are clean. These labels are not provided to BayesJEPA.

During occlusion, mean observation uncertainty rises from 2.40×10^{-4} to 3.515, while mean prediction uncertainty rises from 0.109 to 0.225. The mean gain decreases from 0.998 to 0.059, and the mean filtered residual norm is 1.256, compared with a raw residual norm of 21.775. Once the current target becomes clean, the gain returns close to one even while predictor history remains contaminated. This is consistent with BayesJEPA estimating current-target reliability rather than maintaining an explicit history-corruption state.

These diagnostics measure local update harm and accumulated parameter change. They do not by themselves establish persistent prediction degradation on a common clean population, which would require evaluating matched predictor checkpoints on a fixed held-out clean probe set.

B.6 Observation-Uncertainty Diagnostics

We next evaluate whether the learned observation score tracks latent current-target error on held-out trajectories. The evaluator computes

$$y_k = d^{-1} \|E_\phi(o_k) - E_\phi(o_k^\star)\|_2^2,$$

Table 4: **Observation bias versus variation.** Per-dimension empirical squared bias, centered error, and total latent error on paired held-out observations.

Condition	Squared bias	Centered error	Total error
Clean	0.000000	0.000000	0.000000
Occlusion	1.967424	5.932437	7.899861
Salt-and-Pepper Noise	0.072219	0.873439	0.945658
Darkening	0.320120	4.755628	5.075748
Blur	0.000378	0.001246	0.001624

Table 5: **Observation reliability on held-out trajectories.** Empirical error is y_k; predicted r_k uses only the current observation. One learned estimator and one global calibration are used in all rows.

Condition	Empirical latent error	Predicted r_k	Mean κ_k
Clean	0.000000	0.000319	0.997152
Occlusion (mild)	5.830125	3.516026	0.250075
Occlusion (severe)	9.969598	8.531948	0.030308
Salt-and-Pepper Noise	0.945658	1.402770	0.264983
Darkening	5.075748	4.892003	0.030548
Blur	0.001624	0.000379	0.996652

whereas only o_k is passed to the deployable observation-uncertainty estimator.

The centered component accounts for most of the total observation-side error for all four corruptions, which helps explain why the centered BayesJEPA rule performs similarly to its mean-aware extension in the main ablation.

The learned score tracks the broad scale of latent observation error, although the reported summaries do not establish uniformly accurate calibration. The estimator underestimates occlusion and blur and overestimates salt-and-pepper noise. Nevertheless, the gain decreases strongly for unreliable observations while remaining near one for clean observations and blur, whose absolute latent error is small.

B.7 Additional Planning Results

For completeness, Table 6 reports Success@T at $T \in \{10, 20, 30\}$ for the same matched episodes used in the main planning comparison.

B.8 Qualitative Mixed-Shift Rollout

To complement the aggregate planning results, we visualize a representative closed-loop rollout under the mixed mass-$\times 0.2$ + OCC condition in Figure 7. Both methods are evaluated from the same start–goal configuration under the persistent dynamics shift and transient visual corruption. The figure shows the agent's *executed* trajectory at selected MPC steps; no planner-predicted trajectory is inferred or displayed.

The example illustrates how differences in test-time adaptation can accumulate over closed-loop interaction after transient observation corruption. It is included only as a qualitative illustration of the behavior summarized by the aggregate Success@T results in the main paper and Appendix B.7; our quantitative conclusions are based on the full evaluation across episodes and seeds rather than this individual rollout.

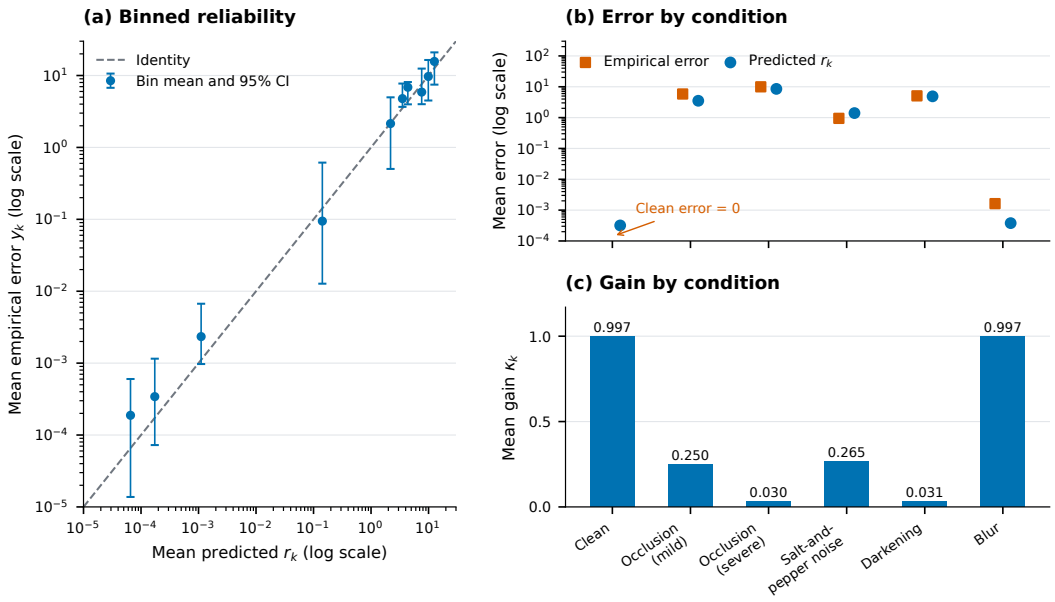

Figure 6: **Observation-uncertainty diagnostics.** (a) Mean empirical latent error y_k against predicted uncertainty r_k in ten bins, with the supplied 95% confidence intervals and an identity reference. (b) Empirical error and predicted uncertainty by condition. (c) Mean scalar filtering gain by condition. Predictions are frozen before evaluation.

Table 6: **Full PointMaze planning profiles.** Each entry reports Success@10/20/30 (%).

Method	OCC	Mass ×0.2	Damping ×20	Mass ×0.2+OCC	Damping ×20+OCC
FrozenJEPA	10.7/49.3/68.7	30.0/56.0/58.7	2.7/18.7/40.7	28.7/53.3/57.3	0.7/10.0/20.7
AdaJEPA	9.3/43.3/61.3	30.7/60.0/63.3	3.3/22.0/43.3	29.3/47.3/50.7	0.7/10.7/22.0
BayesJEPA	10.0/47.3/66.7	30.7/59.3/62.7	2.0/21.3/42.7	30.0/55.3/59.3	1.3/16.7/29.3

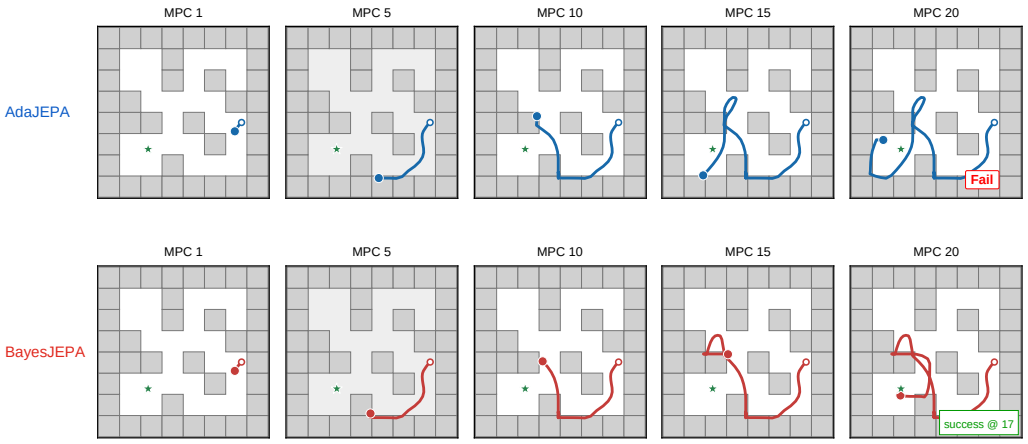

Figure 7: **Representative rollout under a mixed dynamics and observation shift.** Executed trajectories of BayesJEPA (bottom) and AdaJEPA (top) for a representative mass×0.2 + occlusion episode, shown at selected MPC steps. The visualization reports the realized agent trajectory rather than planner-predicted states and serves as a qualitative complement to the aggregate planning results. At MPC 5 step, occlusion is rendered as transparent.