CUG-JEPA: CONTINUAL TEST-TIME ADAPTATION OF LATENT WORLD MODELS UNDER VISUAL CONDITION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models can exploit interactive feedback to improve planning, yet single-episode adaptation does not amount to reliable learning under continual deployment: resetting the model after each episode discards previously acquired corrections, whereas unconstrained continual updates may accumulate negative transfer and catastrophic forgetting. To address this challenge, we propose CUG-JEPA, a continual test-time adaptation framework for sequential shifts in visual conditions. Within the closed loop of model predictive control, CUG-JEPA jointly combines non-monotonic uncertainty weighting, online parameter-importance protection, and conditional local restoration, preserving model state across episodes while balancing adaptation to the current environment against long-term knowledge retention. At deployment time, the model uses real transitions naturally observed after action execution as self-supervision, without requiring additional expert demonstrations. Under the primary continual visual-stream protocol on PushObj, spanning three unseen shapes and five random seeds, CUG-JEPA performs one gradient update per feedback step and improves AdaJEPA's overall success rate from 37.3% to 42.9%, an absolute gain of 5.6 percentage points. Results on disjoint task batches further reveal that the gains depend on the task set, providing practical evidence for moving world models beyond short-term error correction toward controlled self-updating under continual deployment.

1 INTRODUCTION

The long-term objective of latent world models is to capture environmental dynamics in compact internal representations, thereby supporting prediction and planning from high-dimensional observations. Joint-embedding predictive architectures (JEPAs) learn state representations and latent dynamics from reward-free offline trajectories and, when combined with model predictive control (MPC), achieve goal-conditioned control through receding-horizon prediction, action execution, and replanning. Recently, AdaJEPA extended this paradigm to adaptation at deployment time (Wang et al., 2026): it uses real state transitions observed after action execution for self-supervised correction and immediately applies the updated model to the next planning step. Environmental interaction thus becomes not only the process of task execution but also a source of feedback through which the world model learns while acting.

However, the ability to adapt within a single episode does not imply that a world model can learn reliably under continual deployment. AdaJEPA restores the pretrained parameters at the beginning of each episode and therefore cannot retain acquired environmental corrections; simply removing the reset may instead cause continual updates to accumulate negative transfer and catastrophic forgetting. When visual conditions switch repeatedly, the model must not only respond to external environmental changes but also control internal drift induced by its own updates. On the one hand, fitting recent transitions may rewrite the latent-space geometry used for goal comparison and action selection, allowing lower prediction error to coexist with poorer planning and turning short-term adaptation into knowledge loss at the planning level. On the other hand, prediction discrepancies in interactive feedback may arise from either correctable model mismatch or noise that is difficult to exploit; inappropriate updates affect not only model predictions but also subsequent experience through action selection, causing bias to accumulate along the interaction loop. The central chal-

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.

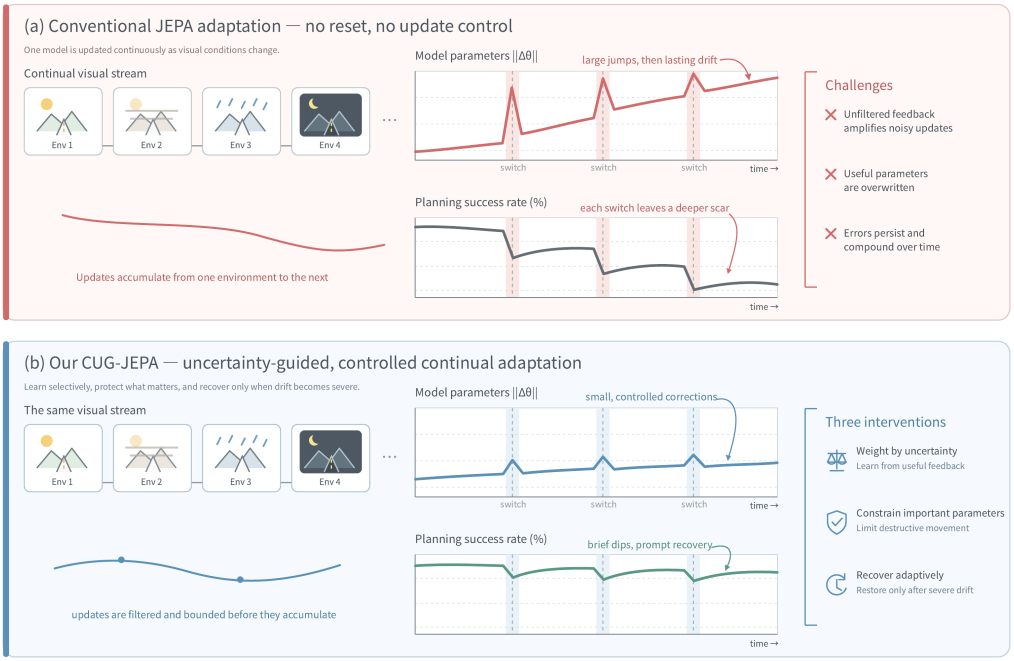

Figure 1: Challenge of continual adaptation under recurring visual conditions. **(a)** Carrying updates across episodes without governance can allow noisy feedback and parameter drift to accumulate, progressively destabilizing latent representations and planning. **(b)** CUG-JEPA controls continual updates through selective feedback weighting, parameter-importance constraints, and conditional local restoration. The curves are conceptual rather than experimental results.

lenge is therefore not merely to keep pace with a changing environment, but to maintain a reliable decision-making basis over time while the world model continually updates itself. Figure 1 conceptually contrasts the cumulative drift that can arise from ungoverned continual updates with the problem addressed by controlled continual adaptation. Inspired by the coordination of adaptation and retention in homeostatic plasticity, we argue for integrating feedback selection, knowledge protection, and interventions against instability into a unified process, so that deployment-time learning serves long-term planning capability rather than treating every decrease in prediction error as progress.

To this end, we propose CUG-JEPA, a continual test-time adaptation framework for latent world models under sequential shifts in visual conditions. Within the MPC loop of planning, execution, adaptation, and replanning, CUG-JEPA retains model parameters and persistent state across episodes, prioritizes moderately uncertain feedback through non-monotonic uncertainty weighting, constrains important parameters from deviating from the pretrained state based on online parameter importance, and triggers local encoder restoration under severe drift, thereby jointly balancing adaptation to the current environment and long-term knowledge retention. The framework performs self-supervised adaptation using only real transitions naturally observed after action execution, with one gradient update per feedback step and no additional expert demonstrations. On the primary continual visual stream in PushObj with three unseen shapes and five random seeds, CUG-JEPA improves AdaJEPA's overall success rate from 37.3% to 42.9%. A deduplicated aggregate evaluation over an expanded task set retains a gain of 3.0 percentage points, although improvements are not consistent across all task batches. These results provide practical evidence for advancing world models from short-term predictive correction toward controlled self-updating under continual deployment.

2 Related Work

JEPA world models and adaptive latent planning. Joint-Embedding Predictive Architectures learn compact representations of high-dimensional observations and action-conditioned latent dynamics through self-supervised prediction (LeCun, 2022; Sobal et al., 2025; Assran et al., 2025). Predicting future representations rather than reconstructing observations provides a latent world model for evaluating candidate action sequences. Goal-conditioned planning combines these predictions with model predictive control (MPC), repeatedly optimizing actions toward an encoded goal (García et al., 1989; Kouvaritakis & Cannon, 2016; Zhou et al., 2025). Parthasarathy et al. (2025) narrow the world-model train–test gap through data synthesis during training, but their model remains fixed during planning and is evaluated only in-distribution. Consequently, changing visual conditions can still misalign observations with learned representations and degrade the predictions on which action selection depends.

AdaJEPA (Wang et al., 2026) addresses this deployment mismatch by incorporating self-supervised model correction into a plan–execute–adapt–replan loop. Real transitions obtained after executing actions supply prediction targets, allowing adaptation before the next MPC optimization without additional expert demonstrations. This closed-loop adaptation interface is the direct foundation of CUG-JEPA. The distinction concerns the lifetime of adapted state: AdaJEPA starts each episode from the same pretrained model, whereas CUG-JEPA carries parameters and continual adaptation state across episodes. We therefore study how to govern accumulated updates under recurring visual shifts, rather than introducing test-time adaptation to latent planning itself. The resulting challenge is to retain useful planning representations while exploiting current interaction feedback.

Continual test-time adaptation and parameter retention. Test-time adaptation updates models using unlabeled deployment inputs, often through self-supervised objectives or entropy minimization (Sun et al., 2020; Wang et al., 2021). Persistent adaptation additionally requires managing forgetting and unreliable feedback. Moving from isolated test inputs to an evolving model makes preservation of the pretrained state a central concern throughout deployment. EATA (Niu et al., 2022) connects efficient adaptation with parameter retention, while CoTTA (Wang et al., 2022) addresses changing test domains; later work further studies stability in dynamic open-world settings and online updates on video streams (Niu et al., 2023; Wang et al., 2025). Relevant strategies include sample selection, parameter regularization, teacher-based stabilization, and restoration of the pretrained state. These approaches largely originate in classification or perception. In closed-loop control, however, an update changes subsequent plans and actions, thereby changing the feedback available for later updates. Adaptation–retention conflicts can thus accumulate through interaction. CUG-JEPA integrates feedback selection, online parameter-importance protection, and conditional local restoration within MPC, rather than introducing these mechanisms individually.

Uncertainty-guided adaptation in planning and control. Online identification has long linked model updates to controller adaptation (Foigel & Richalet, 1979). Online model-based reinforcement learning likewise updates dynamics models from interaction, but usually couples those updates to policy or value learning (Sutton, 1991; Thrun et al., 1990; Hansen et al., 2022; 2024). Uncertainty can inform sample selection, risk assessment, and online updates, but these uses require different interpretations. Predictive risk concerns expected error or adverse consequences; an uncertainty proxy describes predictive variability under model perturbations; update value concerns whether learning from a transition improves subsequent behavior. These quantities need not coincide. CUG-JEPA measures prediction-loss variance for the same transition across repeated predictor-dropout passes, rather than directly estimating control improvement. Its non-monotonic uncertainty weighting emphasizes feedback near the buffer's median uncertainty instead of prioritizing ever-higher variance. Neither high uncertainty nor improved calibration alone establishes useful adaptation. CUG-JEPA differs by placing uncertainty weighting and long-term parameter governance together within a continual latent planning loop.

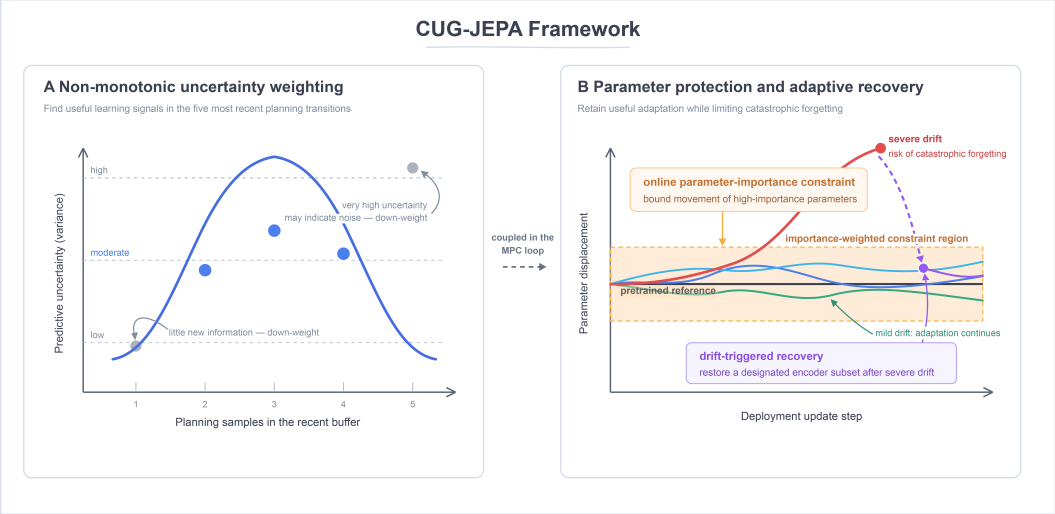

Figure 2: Overview of CUG-JEPA. **(A)** Prediction-loss variance across repeated dropout passes is converted into a non-monotonic weight relative to the current buffer median, emphasizing moderately uncertain feedback. **(B)** An online gradient-squared importance proxy constrains important parameters around the pretrained state, while severe drift indicated by recent-loss variation or latent-distribution KL triggers restoration of a designated encoder subset. The mechanisms are coupled within the continual MPC loop.

3 CUG-JEPA: Continual Test-Time Adaptation of Latent World Models

CUG-JEPA extends AdaJEPA's plan–execute–adapt–replan interface to continual test-time adaptation of a latent world model. Instead of restoring the pretrained state between episodes, it carries the updated model and continual adaptation state through deployment under changing visual conditions. The central problem is to preserve plasticity for current observations without destabilizing representations on which planning depends over time. We address this problem through three coordinated mechanisms: non-monotonic uncertainty weighting selects feedback, online parameter-importance protection constrains accumulated parameter drift, and conditional local restoration intervenes when monitored prediction losses or latent distribution shift indicate severe drift. Figure 2 summarizes these mechanisms and their coupling within the MPC loop.

3.1 Background: JEPA World Models and Closed-Loop Adaptation

Let o_t and a_t denote observations and actions. A latent world model comprises an observation encoder f_{θ_e}, an action encoder g_{θ_a}, and a predictor p_{θ_p}:

$$z_t = f_{\theta_e}(o_t), \qquad u_t = g_{\theta_a}(a_t), \qquad \hat{z}_{t+1} = p_{\theta_p}(z_t, u_t). \tag{1}$$

Here z_t and u_t are observation and action embeddings, and θ_e, θ_a, and θ_p parameterize their respective networks. We use one-step notation, suppressing any short history. Training uses reward-free offline transitions $\mathcal{D}_{\text{off}}$ and future latent targets rather than pixel reconstruction. A JEPA-style prediction objective is

$$\mathcal{L}_{\text{pred}} = \mathbb{E}_{(o_t, a_t, o_{t+1}) \sim \mathcal{D}_{\text{off}}} \ell_t, \qquad \ell_t = \|\hat{z}_{t+1} - \text{sg}(f_{\theta_e}(o_{t+1}))\|_2^2 . \tag{2}$$

The stop-gradient operator sg blocks differentiation through the target representation and is retained during online adaptation (Chen & He, 2021). Segment losses average this objective over valid prediction windows.

For goal observation o_g, let $z_g = f_{\theta_e}(o_g)$. Rollouts start at z_t, recursively applying the predictor to candidate action embeddings with fixed model parameters. Candidate sequences are evaluated using

the following goal-reaching objective:

$$a^*_{t:t+H-1} = \arg\min_{a_{t:t+H-1}} \sum_{h=1}^{H} \alpha_h \left\| \hat{z}_{t+h} - z_g \right\|_2^2 . \tag{3}$$

Here H denotes the planning horizon, α_h nonnegative temporal weights, and $\hat{z}_{t+h}$ the prediction under the candidate sequence. We optimize this squared-distance objective using the cross-entropy method (CEM) (Rubinstein, 1997). Standard MPC executes the first action, observes o_{t+1}, and replans.

AdaJEPA inserts adaptation between execution and replanning. Observed transitions enter a recent buffer, and a small number of gradient updates minimize the same self-supervised prediction loss before planning resumes. The online loss is uniform over buffered feedback, with stop-gradient applied to the next-observation latent target. Each episode originally begins from the same pretrained model. CUG-JEPA extends precisely this episode-reset boundary, retaining parameter updates and continual state across episodes.

3.2 Continual Closed-Loop Plan-and-Adapt

At each episode boundary, CUG-JEPA clears its recent-5 transition buffer while preserving model parameters and continual adaptation state. Visual conditions change with the global interaction step, which continues across episodes. These are observation-level visual condition shifts; physical dynamics remain unchanged. Retained state includes parameter-importance statistics, drift-monitoring statistics, and the restoration cooldown. Episodes therefore do not define independent adaptation problems.

Each iteration encodes the current and goal observations, plans with the current model using CEM, executes the first action chunk, and appends the observed transition to recent-5. It then estimates uncertainty weights, performs one weighted self-supervised update with parameter-importance anchoring, updates drift statistics, and conditionally restores the designated encoder head before replanning. Upon success or after 30 MPC iterations, interaction proceeds to the next episode without resetting parameters to the pretrained state.

Continual interaction state. AdaJEPA's uniform feedback loss remains the underlying adaptation interface, with the same budget of one gradient step per feedback event. Additional dropout passes estimate weights rather than adding gradient updates. Recent-5 contains only the current episode's latest transitions. The weighted implementation additionally disables the baseline's contiguous-segment merging, so differences between AdaJEPA and its uncertainty-weighted variant cannot be attributed solely to the weighting function.

Non-monotonic uncertainty weighting. For each buffered segment i, we perform $K = 10$ forward passes with predictor dropout enabled, obtaining scalar prediction losses $\ell_i^{(k)}$. Their prediction-loss variance, using denominator K, is

$$\bar{\ell}_i = \frac{1}{K} \sum_{k=1}^{K} \ell_i^{(k)}, \qquad v_i = \frac{1}{K} \sum_{k=1}^{K} \left(\ell_i^{(k)} - \bar{\ell}_i \right)^2 . \tag{4}$$

This measures within-segment loss variability, not latent-vector variance or variance across buffered samples. Let $m = \operatorname{median}_i v_i$, using PyTorch's median. For $m \le 0$, all weights equal one; otherwise,

$$w_i = \exp\left[-\frac{1}{2} \left(\frac{\log \max(v_i/m, 10^{-8})}{s} \right)^2 \right], \qquad s = 1. \tag{5}$$

We retain $S = \{ i : w_i \ge 0.05 \}$, keeping a maximum-weight segment if S is empty, and optimize

$$\mathcal{L}_{\mathrm{UG}} = \frac{\sum_{i \in S} w_i \ell_i}{\sum_{i \in S} w_i}. \tag{6}$$

Here ℓ_i is the segment loss used for adaptation. This inverted-U rule favors uncertainty near the current buffer median, reducing the influence of potentially low-information and extremely uncertain

feedback. Weights are relative to the current buffer, not an absolute uncertainty threshold shared across visual conditions. High variance is not identified with noise, nor is intermediate variance assumed to guarantee useful updates.

Online parameter-importance protection. Let θ_t collect adaptable parameters, θ_0 their values in the pretrained state, and $g_t = \nabla_{\theta_t} \mathcal{L}_{\mathrm{UG}}$ the task gradient. We first update the gradient-squared importance estimate

$$F_t = \beta F_{t-1} + (1-\beta) g_t^{\odot 2}, \qquad \beta = 0.9, \tag{7}$$

and then form

$$g_{\mathrm{total}} = g_t + 2\lambda F_t \odot (\theta_t - \theta_0), \qquad \lambda = 10. \tag{8}$$

Here $\odot$ and $\odot 2$ denote elementwise multiplication and squaring, respectively. The resulting gradient drives one Adam step. This Fisher-inspired online importance proxy is not exact Fisher information. Anchoring restricts displacement along important directions while allowing less constrained directions to adapt; it targets retention rather than guaranteeing faster adaptation.

Drift monitoring and conditional restoration. We track the coefficient of variation (CV) of the latest 10 prediction losses and latent distribution shift, measured by diagonal-Gaussian KL divergence from the current exponential-moving-average latent distribution to a source-domain anchor. After adaptation, latent KL ≥ 2000 or loss CV ≥ 1.0 triggers restoration if at least 20 MPC steps have elapsed since the previous restoration. Only the designated encoder head returns to its pretrained state; corresponding importance and monitoring states are reset, while recent-5 remains intact by default. These thresholds specify the current restoration rule, not a protocol-independent stability guarantee.

Planning at the next interaction step uses the resulting model, which also carries into subsequent episodes. Together, these mechanisms govern feedback selection, parameter retention, and conditional intervention during accumulated online updates. Further derivations of the uncertainty profile and protected update, together with explicit definitions of the drift statistics and restoration rule, are provided in Appendix B.1.

4 Experimental Setup

4.1 Tasks and Continual Visual Streams

Our primary evaluation uses PushObj with three unseen object shapes: I, small tee, and square. We evaluate five random seeds (100, 200, 300, 400, and 500), with 30 episodes for each shape–seed–method cell. Each method therefore contributes 450 episodes to the primary comparison. Planning uses CEM with at most 30 replanning iterations per episode. The visual stream cycles through blur, light salt-and-pepper noise, stronger salt-and-pepper noise, and dark observations, switching every 100 global MPC steps for two cycles. Model parameters persist across episodes, whereas the recent transition buffer is cleared at each episode boundary and contains at most five within-episode segments.

We separately evaluate a disjoint paired task batch containing 90 episodes per core method. This batch reuses seed 100 to preserve paired paths across methods and therefore provides evidence about task sampling rather than an additional independent random seed. We also report a deduplicated aggregate over the primary and disjoint batches, comprising 147 unique runs and 4,410 episodes in total.

4.2 Methods and Adaptation Budget

The primary comparison includes AdaJEPA, uncertainty weighting (UG), UG with parameter-importance protection (UG+KP), and the full CUG-JEPA method. Five alternative uncertainty estimators are evaluated under the same primary protocol: latent predictive entropy (LPE), Monte Carlo dropout, augmentation consistency, Mahalanobis distance, and heteroscedastic prediction. All adaptive methods use the same budget of one Adam update after each real feedback event. CUG-JEPA uses $K = 10$ stochastic predictor passes, log-scale parameter $s = 1$, a weight-retention threshold of 0.05, parameter-protection strength $\lambda = 10$, and the restoration rule described in Section 3.

5.1 Experiment 1: End-to-End Performance under Continual Visual Shifts

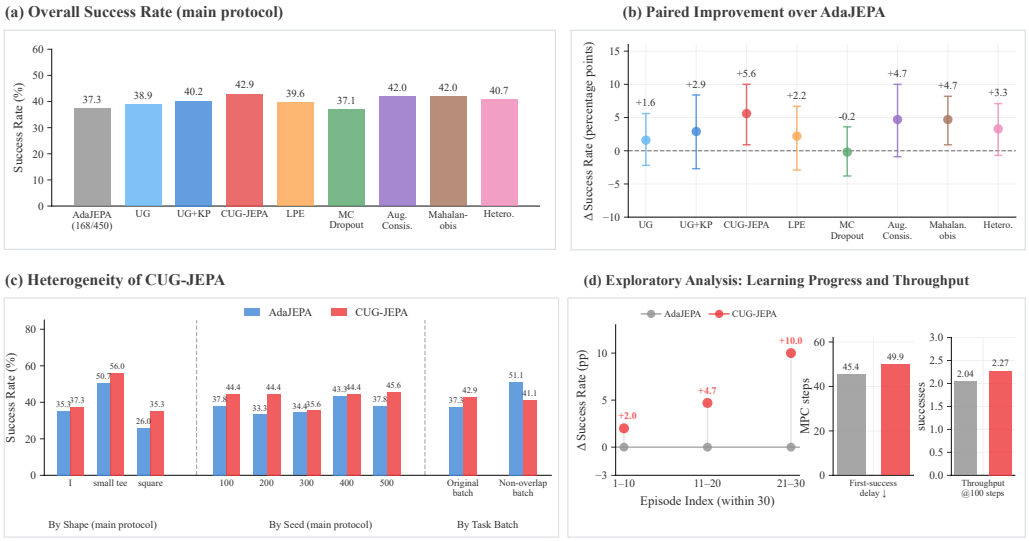

Figure 3: End-to-end performance under continual visual shifts. **(a)** Overall success rate for nine methods under the primary five-seed protocol; counts are shown for AdaJEPA and each method has 450 episodes. **(b)** Paired improvement over AdaJEPA with 95% cluster-bootstrap confidence intervals. **(c)** CUG-JEPA and AdaJEPA across shapes, random seeds, and task batches. **(d)** Exploratory episode-block gains, restricted mean first-success delay, and success throughput at 100 MPC steps. The temporal summaries are descriptive and are not used for confirmatory claims.

Mechanism and boundary-condition experiments use episode-reset protocols where stated. These include the three-seed AdaJEPA Figure 2 reproduction with gradient-descent and CEM planners, clean-I and fixed-noise PushObj evaluations, a PushT protocol that switches among three visual corruptions every five MPC steps, uncertainty-calibration comparisons, return-to-clean restoration tests, and a separate LeWorldModel $A \rightarrow B \rightarrow A'$ diagnostic. These experiments test specific mechanisms and should not be treated as additional samples from the primary continual stream.

4.3 Metrics and Statistical Analysis

The primary endpoint is task success rate. For the main protocol, method differences from AdaJEPA are computed over the 15 paired shape–seed clusters. We report paired cluster-bootstrap 95% confidence intervals, raw sign-flip tests, and Holm-adjusted p-values across the eight method-to-AdaJEPA comparisons. Absolute success counts accompany the main rates. Exploratory episode-block and switch-aligned analyses are reported descriptively and are not used for confirmatory statistical claims. In particular, a confidence interval that crosses zero is not interpreted as evidence of improvement.

5 Experimental Results

Figure 3 summarizes end-to-end performance, paired effects, heterogeneity, and exploratory temporal diagnostics under continual visual shifts.

Primary continual-stream performance. CUG-JEPA obtains the highest point estimate in the primary matrix, with 193 successes in 450 episodes (42.9%), compared with 168 in 450 (37.3%) for AdaJEPA. The paired difference is $+5.6$ percentage points with a 95% cluster-bootstrap interval of $[+0.9, +10.0]$. The raw sign-flip test gives $p = 0.0479$, but the comparison does not remain significant after correction across the eight method comparisons (Holm $p = 0.3350$). Augmentation consistency and Mahalanobis each reach 42.0%, only 0.9 points below CUG-JEPA. Heteroscedastic

5.2 Experiment 2: Mechanism Diagnostics and Boundary Conditions

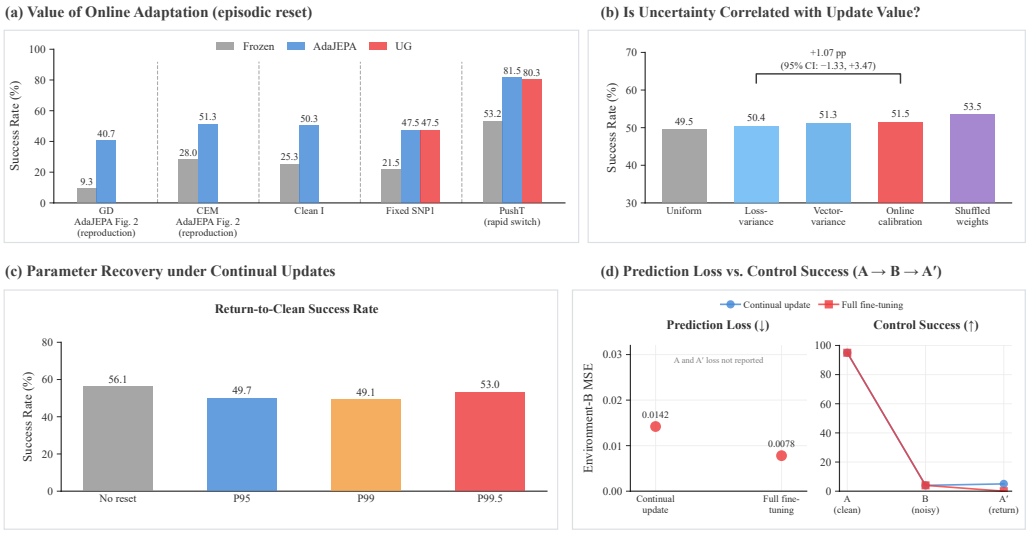

Figure 4: Mechanism diagnostics and boundary conditions. **(a)** Online adaptation versus a frozen model under episode-reset protocols; UG is shown only where it was measured. **(b)** Streaming success for uncertainty weighting and calibration strategies, including the paired online-calibration contrast. **(c)** Verified return-to-clean success rates under restoration thresholds. **(d)** Environment-B prediction losses and the $A \to B \to A'$ control diagnostic, demonstrating that lower prediction loss need not imply better control or retention.

prediction reaches 40.7%, UG+KP 40.2%, LPE 39.6%, UG 38.9%, and Monte Carlo dropout 37.1%. Thus, the evaluated matrix supports a favorable point estimate for CUG-JEPA, but not a claim of corrected statistical superiority over all alternatives.

Heterogeneity and task-set sensitivity. The aggregate gain does not arise uniformly across shapes. On I, CUG-JEPA reaches 37.3% versus 35.3% for AdaJEPA; on small tee, 56.0% versus 50.7%; and on square, 35.3% versus 26.0%. Seed-level differences are nonnegative but vary from $+1.1$ to $+11.1$ points. More importantly, the disjoint task batch reverses the primary ranking: AdaJEPA reaches 51.1% whereas CUG-JEPA reaches 41.1%. After deduplicating and combining the primary and disjoint batches, CUG-JEPA remains the highest point estimate at 42.6%, compared with 39.6% for AdaJEPA, but the difference contracts to $+3.0$ points with a 95% interval of $[-2.0, +7.8]$ and Holm $p = 0.8833$. The main result is therefore sensitive to the sampled task set.

Exploratory temporal diagnostics. Across episode blocks 1–10, 11–20, and 21–30, the descriptive CUG-JEPA gains are $+2.0$, $+4.7$, and $+10.0$ points, respectively. Because block-specific intervals were not estimated, this pattern cannot establish cumulative improvement. Switch-aligned diagnostics likewise do not support faster immediate recovery: the restricted mean first-success delay is 49.9 MPC steps for CUG-JEPA and 45.4 for AdaJEPA. At 100 steps, success throughput is 2.27 versus 2.04, a difference of $+0.22$ with a 95% interval of $[-0.31, +0.76]$. These results suggest a possible later-throughput advantage while ruling out a general claim of faster adaptation after every switch.

Figure 4 reports mechanism diagnostics and boundary conditions. These experiments clarify which effects are attributable to online adaptation, whether uncertainty scores identify update value, and where restoration or prediction-loss optimization fails to preserve control.

Value of online adaptation. Episode-reset controls consistently show that adapting the latent world model is valuable. In the three-seed AdaJEPA reproduction, adaptation raises success from 9.3% to 40.7% with the gradient-descent planner and from 28.0% to 51.3% with CEM. On clean I, AdaJEPA reaches 50.3% versus 25.3% for the frozen model. Under fixed light salt-and-pepper

noise, the corresponding rates are 47.5% and 21.5%. In the rapid-switch PushT protocol, they are 81.5% and 53.2%. These results establish the value of deployment-time adaptation, but they do not establish an additional benefit from uncertainty weighting: UG matches AdaJEPA at 47.5% under fixed noise and reaches 80.3% under rapid switching.

Does uncertainty identify update value? In the calibration study, uniform weighting reaches 49.5% streaming success, loss-variance weighting 50.4%, vector-variance weighting 51.3%, online calibration 51.5%, and shuffled weights 53.5%. The exact paired difference between online calibration and loss-variance weighting is $+1.07$ points with a 95% interval of $[-1.33, +3.47]$, below the pre-specified $+5$ point practical target. Shuffling the correspondence between samples and weights does not reduce the point estimate. The uncertainty signal may therefore contain risk information, but the current weighting rule has not been shown to identify which transition will produce a useful control update.

Parameter recovery under continual updates. In the return-to-clean experiment, carrying the adapted model without restoration achieves 56.1% success after returning to the clean environment. The P95, P99, and P99.5 restoration thresholds reach 49.7%, 49.1%, and 53.0%, respectively. P99.5 reduces the measured forgetting difference relative to less conservative thresholds, but its absolute return success and aggregate score remain below the no-restoration condition. Fixed-threshold local restoration is therefore a boundary mechanism rather than a universally beneficial solution.

Prediction loss versus control success. The separate LeWorldModel diagnostic (Maes et al., 2026) changes both visual appearance and dynamics, so it is not direct evidence for the primary visual-only protocol. Nevertheless, it exposes a central failure mode. Continual adaptation and full fine-tuning achieve environment-B prediction MSE values of 0.0142 and 0.0078, yet both obtain only 4% control success in B. When returned to A', their control success falls from the 95% source-domain level to 5% and 0%, respectively. The lowest prediction loss thus coincides with neither improved control nor preserved planning capability.

5.1 Scope and Limitations

The current evidence is limited to simulated visual-condition changes, primarily with an AdaJEPA base model and PushObj, plus targeted PushT and LeWorldModel diagnostics. It does not establish robustness to dynamics changes, real-camera shifts, real-robot deployment, or other world-model families. The disjoint task batch demonstrates substantial task-sampling sensitivity, and the present method ladder is not a complete factorial ablation with matched effective update strength. The switch-aligned analysis is exploratory because successful episodes can alter exposure timing and goals are not phase matched. Future evaluation should use forced switches from shared checkpoints, independent task batches, complete component ablations, a second control task and world-model base, and controlled compute and latency measurements.

6 Conclusion

This paper presented CUG-JEPA, a continual-adaptation framework for latent world models under sequential shifts in visual conditions. By embedding uncertainty-guided feedback weighting, online parameter-importance protection, and conditional local restoration into the MPC loop, CUG-JEPA combines the use of interactive feedback with constraints on cumulative parameter updates, exploring an alternative to both episode-wise forgetting and unconstrained continual adaptation.

Experiments show that the framework improves overall control success on the primary continual visual stream and retains a positive gain in aggregated results over the expanded task set, although not every task batch benefits. More broadly, world models deployed in changing environments must address not only how to reduce immediate prediction error, but also how to preserve the knowledge on which future actions depend while updating themselves. We argue that controlled continual self-updating is an important bridge between short-term adaptation and long-term autonomous behavior: continual interaction should become a usable source of knowledge rather than an uncontrolled source of model drift, thereby laying the foundation for reliable perception and planning in changing environments.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, were used to assist with language editing, manuscript organization, literature retrieval, bibliographic verification, and software-code editing. They were also used to draft or revise portions of the manuscript and, where applicable, to provide feedback on experimental design and result interpretation. All AI-assisted text, citations, code, analyses, and reported results were manually reviewed and verified by the authors. Generative AI was not used to generate experimental datasets or prove mathematical claims. The authors take full responsibility for the final content of this work.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew J. Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668.

J. K. Foigel and Jacques Richalet. Self-adapting IDCOM. *IFAC Proceedings Volumes*, 12(8):1095–1102, 1979. doi: 10.1016/S1474-6670(17)65531-9.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989. doi: 10.1016/0005-1098(89)90002-2.

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

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

Basil Kouvaritakis and Mark Cannon. *Model Predictive Control: Classical, Robust and Stochastic*. Springer, 2016. doi: 10.1007/978-3-319-24853-0.

Yann LeCun. A path towards autonomous machine intelligence. Technical report, OpenReview, 2022. Version 0.9.2.

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. doi: 10.48550/arXiv.2603.19312.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*, 2023.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025. doi: 10.48550/arXiv.2512.09929.

Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997. doi: 10.1016/S0377-2217(96)00385-2.

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 *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1465.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377.

Sebastian Thrun, Knut Möller, and Alexander Linden. Planning with an adaptive world model. In *Advances in Neural Information Processing Systems*, volume 3, 1990.

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.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022.

Renhao Wang, Yu Sun, Arnuv Tandon, Yossi Gandelsman, Xinlei Chen, Alexei A. Efros, and Xiaolong Wang. Test-time training on video streams. *Journal of Machine Learning Research*, 26(9): 1–29, 2025.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model. *arXiv preprint arXiv:2606.32026*, 2026. doi: 10.48550/arXiv.2606.32026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

A Environments and Data

A.1 Main Continual Visual Stream on PushObj

The primary experiments use AdaJEPA's PushObj environment and the released `pushobj_shape_shift` pretrained checkpoint (Wang et al., 2026). PushObj extends the T-shaped object in PushT to multiple geometries (Chi et al., 2025; Zhou et al., 2025). The checkpoint corresponds to the shape-shift setting trained on the T, L, Z, and + shapes. CUG-JEPA does not retrain this world model; deployment-time adaptation methods are compared from the same pretrained starting point. Evaluation uses three shapes unseen during training: I, small tee, and square. Each shape–method–seed run evaluates 30 tasks from the corresponding `val_<shape>/plan_targets.pkl` target set. Methods with the same shape and seed share task sampling, enabling paired comparisons.

The continual visual stream follows blur→SNP1→SNP5→dark, switching every 100 global MPC steps for two configured cycles. These perturbations modify only the input images, leaving physical state, action dynamics, and goal definitions unchanged. Model parameters, Adam optimizer state, online parameter importance, and drift statistics persist across episodes. The recent-5 transition buffer is cleared at episode boundaries so that discontinuous episodes are not concatenated into one training segment. Successful episodes terminate early, so methods may receive different amounts of exposure to individual visual conditions. Primary inference therefore clusters at the complete shape–seed run rather than treating episodes within a run as independent observations.

A.2 Rapid Within-Episode Switching on PushT

The rapid-switch diagnostic uses the original T-shaped PushT environment and 200 unique target tasks from `val_T/plan_targets.pkl`. Observations switch among blur, SNP1, and SNP5 every five MPC iterations. Seeds 101, 202, and 303 balance the starting phase so that every task experiences each of the three initial phases across seeds. Each method contains 600 episodes; the complete eight-method matrix comprises 24 method–seed runs and 4,800 episodes. Unlike the primary continual protocol, this diagnostic restores the source model, recreates an optimizer with no accumulated momentum, and clears the buffer, method caches, and uncertainty state at the start of every episode. It therefore tests rapid within-episode adaptation rather than cross-episode retention. Because no matched static PushT control was run, the protocol cannot by itself estimate the cost caused specifically by switching.

A.3 Separate Retention Diagnostic and Uncovered Environments

The LeWorldModel (Maes et al., 2026) $A \to B \to A'$ study is a separate retention diagnostic rather than part of the primary PushObj benchmark. Environment B jointly changes visual appearance, friction, and action scaling. Each method runs for 50 episodes in B and is then evaluated for 20 episodes in A'. This study tests whether prediction error, control success, and return-to-source retention agree, and its results are not pooled with success rates from the observation-only PushObj benchmark. The PointMaze-Medium dynamics shifts and Diverse PointMaze layout shifts in the AdaJEPA appendix are not currently included in the formal CUG-JEPA evaluation; accordingly, this paper reports no CUG-JEPA results for those environments.

B Experimental Details

B.1 Derivations and Formal Update Rules

This section expands the operational definitions in Section 3.2. The derivations clarify algebraic properties of the implemented weighting, protection, and restoration rules; they do not constitute a convergence or closed-loop stability guarantee.

Shape of the uncertainty-weighting function. For $m > 0$, define the log-relative uncertainty $r_i = \log(v_i/m)$. Before applying the numerical floor in Equation 5, the weighting rule can be written as

$$r_i = \log \frac{v_i}{m}, \qquad w_i = \exp\left(-\frac{r_i^2}{2s^2}\right). \tag{B1}$$

For $v_i > 0$, differentiating with respect to the uncertainty gives

$$\frac{\partial w_i}{\partial v_i} = -\frac{\log(v_i/m)}{s^2 v_i}\, w_i. \tag{B2}$$

Because $w_i > 0$, the derivative is positive when $v_i < m$, zero at $v_i = m$, and negative when $v_i > m$. Hence w_i reaches its maximum at the buffer median and decreases when uncertainty moves away from that median in either multiplicative direction. The 10^{-8} floor in Equation 5 only prevents an undefined logarithm; this derivative applies outside the clipped region.

Importance-weighted quadratic anchoring. After computing $g_t = \nabla_{\theta_t} \mathcal{L}_{\mathrm{UG}}$ and updating F_t with Equation 7, consider the protected objective

$$\begin{aligned} \mathcal{J}_t(\theta) &= \mathcal{L}_{\mathrm{UG}}(\theta) + \lambda \left\| F_t^{1/2} \odot (\theta - \theta_0) \right\|_2^2 \\ &= \mathcal{L}_{\mathrm{UG}}(\theta) + \lambda \sum_j F_{t,j} (\theta_j - \theta_{0,j})^2. \end{aligned} \tag{B3}$$

Treating the current importance estimate as fixed during this optimizer step yields

$$\nabla_\theta \mathcal{J}_t(\theta)|_{\theta=\theta_t} = g_t + 2\lambda F_t \odot (\theta_t - \theta_0), \tag{B4}$$

which is the gradient in Equation 8. Since F_t is an exponential moving average of squared gradients, its entries are nonnegative, and Equation B3 defines a positive-semidefinite quadratic penalty. Parameters with larger accumulated importance therefore receive stronger resistance to displacement from the pretrained state. This interpretation does not make F_t exact Fisher information.

Drift statistics and restoration trigger. Let $n_t \le H$ denote the number of available losses in the sliding window, with $H = 10$. For $n_t \ge 2$, the implemented loss coefficient of variation is

$$\begin{aligned} \bar{\ell}_t &= \frac{1}{n_t} \sum_{h=t-n_t+1}^{t} \ell_h, \\ \mathrm{CV}_t &= \frac{\sqrt{\frac{1}{n_t} \sum_{h=t-n_t+1}^{t} (\ell_h - \bar{\ell}_t)^2}}{\bar{\ell}_t + 10^{-8}}. \end{aligned} \tag{B5}$$

Let $q_t = \mathcal{N}(\mu_t, \mathrm{diag}(\sigma_t^2))$ be the current latent distribution and $q_0 = \mathcal{N}(\mu_0, \mathrm{diag}(\sigma_0^2))$ the source anchor. The current mean and variance use exponential moving averages with decay $\gamma = 0.9$. The diagonal-Gaussian divergence from the current distribution to the source anchor is

$$D_{\mathrm{KL}}(q_t \| q_0) = \frac{1}{2} \sum_d \left[\frac{\sigma_{t,d}^2 + (\mu_{t,d} - \mu_{0,d})^2}{\sigma_{0,d}^2} - 1 + \log \frac{\sigma_{0,d}^2}{\sigma_{t,d}^2} \right]. \tag{B6}$$

Writing t_{last} for the most recent restoration step, the binary trigger is

$$R_t = \mathbb{I}[(D_{\mathrm{KL}}(q_t \| q_0) \ge 2000 \ \vee \ \mathrm{CV}_t \ge 1.0) \wedge (t - t_{\mathrm{last}} \ge 20)]. \tag{B7}$$

When $R_t = 1$, the designated encoder head is restored to its pretrained value, its associated importance state and the current drift-monitoring state are reset, and t_{last} is updated. The recent transition buffer is retained by default. This rule formalizes a conditional intervention and does not imply a protocol-independent stability guarantee.

B.2 Fixed Implementation and Hyperparameters

CUG-JEPA inherits AdaJEPA's released checkpoint, model architecture, and planning configuration, changing only the lifetime and governance of deployment-time updates. Unless stated otherwise, the primary continual experiments use the frozen settings in Table B1. Pretraining was not rerun in this study, so the released-checkpoint properties and deployment-time parameters are reported separately.

B.3 Experimental and Ablation Matrix

Whereas AdaJEPA primarily studies whether to adapt at test time, CUG-JEPA focuses on whether feedback selection, parameter protection, and conditional intervention can govern accumulated updates once adaptation state persists across episodes. Protocols remain separate: success rates are not pooled across tasks, parameter lifecycles, or sample sizes. Section 5 reports the outcomes; Table B2 records the design scale and role of each evidence block.

AdaJEPA Appendix B also contains systematic ablations of adaptation layers, learning rate and gradient steps, replay-buffer design, and offline training-data scale. CUG-JEPA fixes the released settings as its baseline: it does not rerun the training-scale study or present AdaJEPA's layer-selection ablation as a new CUG-JEPA result. The corresponding new evidence instead consists of the component matrix, alternative-UQ, calibration, restoration, and rapid-switch diagnostics under persistent parameter lifecycles. Consequently, AdaJEPA findings such as robustness to layer choice or larger gains under data scarcity are treated only as baseline context and are not restated as experimental findings of CUG-JEPA.

C Visualization of Planning Trajectories

CUG-JEPA retains developmental planning visualizations under four fixed visual conditions. At each MPC replanning point, the same candidate action sequence is rolled out in two ways: the

Table B1: Fixed implementation and hyperparameters used in the primary continual experiments.

Item	Fixed setting
Released checkpoint	`pushobj_shape_shift/checkpoints/model_latest.pth`
Input and temporal context	224×224 images; 3 history frames; frameskip $= 5$
Planner	CEM; horizon $= 25$; 200 samples; top-30 elites; 10 optimization steps; variance scale $= 1$
MPC execution	Execute 5 actions per step; at most 30 replans per episode; reuse unexecuted actions as a warm start
Online optimization	Adam; one gradient step per feedback; update every MPC iteration
Learning rates	Predictor 5×10^{-4}; encoder 1×10^{-5}
Adapted parameters	Last predictor Transformer block including final LayerNorm, plus the encoder's last submodule
Replay buffer	Recent-5; cleared at each episode boundary
Uncertainty weighting	10 dropout passes; log scale $= 1.0$; weights below 0.05 are filtered
Parameter importance	Fisher-inspired squared-gradient EMA with decay 0.9; anchoring coefficient $\lambda = 10$
Drift monitoring	CV over the latest 10 prediction losses; diagonal-Gaussian KL from latent EMA to a source anchor
Severe-drift thresholds	Latent KL ≥ 2000 or loss CV ≥ 1.0
Conditional restoration	Restore the full encoder head; 20-MPC-step cooldown; retain recent-5 by default

Table B2: Experimental and ablation matrix. Each protocol is analyzed separately.

Protocol	Methods and scale	Primary purpose
Core component matrix	Ada, UG, UG+KP, CUG; 3 shapes $\times$ 5 seeds $\times$ 30 episodes $=$ 60 runs / 1,800 episodes	Compare uniform adaptation, uncertainty weighting, parameter protection, and the full combination
Additional UQ matrix	LPE, MC Dropout, augmentation consistency, Mahalanobis, heteroscedastic; 75 runs / 2,250 episodes; 135 runs / 4,050 episodes including the core matrix	Compare fixed implementations of alternative uncertainty scores
Calibration study	6 methods; 3 shapes $\times$ 5 seeds; 90 runs / 6,300 episodes	Test whether calibration and sample-to-weight correspondence yield control utility
Rapid PushT switching	8 methods $\times$ 3 seeds $\times$ 200 tasks $=$ 24 runs / 4,800 episodes	Isolate rapid within-episode adaptation with a full reset between episodes
Restoration study	No restoration, P90, P95, P99, P99.5; 3 shapes $\times$ 3 seeds	Test local restoration against absolute return-to-source performance
LeWorldModel $A \to B \to A'$	Frozen, episode-reset, continual, full fine-tuning; 50 episodes in B and 20 episodes in A' per method	Test whether prediction loss is a surrogate for control and retention

Simulator row executes the sequence under the true environment dynamics, whereas the Decoder row maps the world model's imagined latent rollout back to pixels; the rightmost column shows the goal observation. Each panel in Figure C1 selects six evenly spaced frames from one source-traceable replanning rollout and retains its original goal column. All four panels use seed 100 and the same CUG-JEPA development visualization configuration. Panel (a) is a successful I-shaped sample under blur, panel (b) a successful small-tee sample under SNP1, panel (c) a successful square sample under SNP5, and panel (d) a failed I-shaped sample under dark. The figure is used to inspect geometric agreement and failure patterns across visual conditions, not to estimate a method effect.

The four-condition visualization is intended as a qualitative illustration rather than a controlled method comparison. The panels correspond to different tasks, so shapes, targets, and outcomes are not fully matched, and the dark panel retains its observed failure. We therefore make no claim that CUG-JEPA qualitatively outperforms AdaJEPA. A strict paired comparison would require frozen task, seed, noise, and initial state across methods.

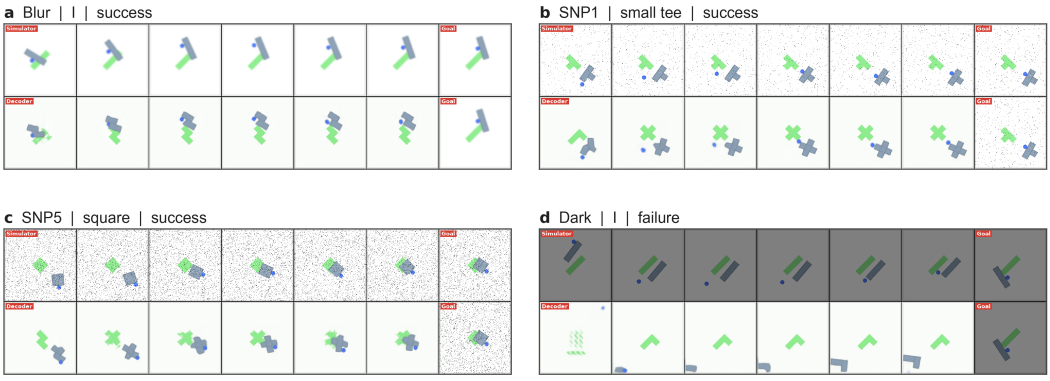

Figure C1: CUG-JEPA planning rollouts under four fixed visual conditions. In each panel, the upper row shows the state sequence produced by the same candidate actions in the Simulator, the lower row shows the decoded latent sequence imagined by the world model, and the rightmost column is the goal; six evenly spaced rollout frames are shown. **(a)** Blur, I shape, success. **(b)** SNP1, small tee, success. **(c)** SNP5, square, success. **(d)** Dark, I shape, failure. All samples use seed 100. Panels come from different tasks and therefore illustrate source-traceable condition-specific behavior rather than a paired comparison or an estimate of aggregate performance.