# FLOW-JEPA: ROBUST LATENT DYNAMICS FOR JEPA WORLD MODELS VIA FLOW MATCHING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Joint-Embedding Predictive Architectures (JEPAs) provide a powerful framework for latent world modeling and planning in a reconstruction-free manner. Although numerous JEPA-based approaches have been proposed to mitigate representation collapse, our experiments on localized, out-of-distribution visual noise reveal that performance degradation remains pronounced and unresolved. We propose Flow-JEPA (F-JEPA), a flow-based latent dynamics model that jointly generates a sequence of future latent states conditioned on the current observation and actions. A Gaussian distribution serves as the flow source, exposing the vector field to perturbed latent trajectories as it learns to transport them toward clean future representations. This formulation retains the reconstruction-free JEPA framework while switching from pointwise transition regression to stochastic trajectory-level prediction. F-JEPA raises mean success from $86\%$ to $92\%$ under clean observations and from $67\%$ to $86\%$ under noisy conditions. Further evaluations over varying perturbation severity and inference settings show that the performance advantage persists across a broad range of conditions. These results suggest that conditional flow matching provides a promising alternative to deterministic autoregressive prediction as a dynamics formulation in JEPA world models.

## 1 INTRODUCTION

World models learn predictive representations of environment dynamics by modeling the future consequences of actions, providing a foundation for planning, policy learning, and synthetic data generation. Early and influential approaches learn compact latent dynamics from high-dimensional observations and use the learned model for imagination-based control or policy optimization (Ha & Schmidhuber, 2018; Hafner et al., 2019a). More recent systems, such as IRIS (Micheli et al., 2022), further demonstrate that sequence models can serve as powerful world models by combining learned visual tokenization with autoregressive dynamics prediction. Despite their success, many such models rely on pixel-level reconstruction or generative prediction objectives. Although these objectives preserve rich visual detail, they can also force the model to spend capacity on task-irrelevant factors such as background or lighting. This motivates reconstruction-free latent world models that predict future states in an abstract representation space rather than reconstructing raw pixels.

Joint-Embedding Predictive Architectures (JEPAs) (LeCun, 2022) provide a fundamental framework for this reconstruction-free setting. Instead of predicting pixels, JEPA-style methods encode observations into latent embeddings and train a predictor to forecast missing or future representations. LeWorldModel (LeWM) (Maes et al., 2026) extends this idea to compact, end-to-end world modeling directly from pixels in reward-free environments. It learns an action-conditioned latent dynamics predictor, while using the Sketched-Isotropic-Gaussian Regularizer (SIGReg) (Balestriero & LeCun, 2025) to prevent representation collapse without reconstruction losses, frozen encoders, exponential moving averages, or auxiliary supervision. Avoiding reconstruction, however, does not by itself guarantee that the resulting dynamics models are insensitive to nuisance variation.

A common formulation of latent world-model dynamics is deterministic autoregressive prediction, in which a one-step state-transition model is recursively applied to generate a future trajectory. This design has two structural properties that make it vulnerable to perturbations:



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.

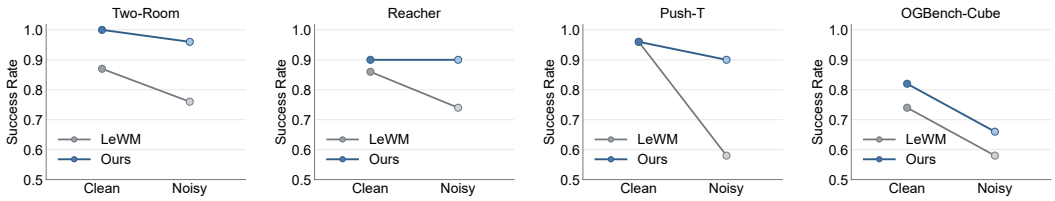



Figure 1: **Latent-dynamics formulation affects planning robustness.** F-JEPA, which instantiates trajectory-level vector-field dynamics, achieves higher average success under clean observations and substantially smaller degradation under noisy observations than the autoregressive LeWM baseline across four environments. The $y$-axis starts at 0.5.

- **Error accumulation.** The latent history, increasingly composed of model predictions, is reused as input to subsequent transitions, allowing perturbation-induced deviations and prediction errors to propagate across the rollout.
- **Pointwise supervision.** The next-latent regression objective supervises the predictor at observed latent targets. While this objective encourages accurate prediction on clean training trajectories, it provides no explicit mechanism for modeling or recovering from local perturbations of the latent state.

Using LeWM as a representative autoregressive JEPA world model, we find that planning can degrade substantially under localized visual perturbations that leave the underlying physical state and environment dynamics unchanged. This failure mode suggests that robustness may depend not only on how observations are encoded, but also on how dynamics are modeled within the learned representation space, raising a broader question: *to what extent does the formulation of latent dynamics determine the robustness of JEPA world models?*

We investigate trajectory-level conditional vector-field prediction as an alternative formulation of latent dynamics. The resulting model, Flow-JEPA (F-JEPA), jointly processes the complete future latent trajectory and uses a conditional flow matching (CFM) objective to learn its evolution conditioned on the current latent state and an action sequence. CFM learns a time-dependent vector field that transports samples from a source distribution toward a conditional target distribution and has been used for continuous generation in images, video, and robot actions (Lipman et al., 2022; Wan et al., 2025; Black et al., 2024). During training, the vector-field predictor is supervised at stochastic interpolation states between source and target trajectories rather than only at clean latent endpoints.

SIGReg and CFM introduce Gaussian structure at different stages of the model. SIGReg regularizes the marginal encoder distribution, whereas CFM uses a Gaussian source to define stochastic transport paths for dynamics learning. By exposing the dynamics model to latent states beyond the observed transitions, such training may improve tolerance to perturbation-induced deviations. We test this hypothesis across four control environments. F-JEPA improves average planning performance under clean conditions and exhibits substantially smaller degradation under perturbations than the autoregressive baseline. Mean success increases from $86\%$ to $92\%$ under clean observations and from $67\%$ to $86\%$ under noisy observations. Additional stress tests show that the observed advantage is not confined to a single perturbation strength or inference configuration.

Our main findings are summarized as follows:

- We identify and characterize a robustness failure mode in a representative autoregressive JEPA world model: task-irrelevant visual perturbations can substantially degrade latent-space planning even when the underlying task dynamics are unchanged.
- We study latent-dynamics robustness by contrasting deterministic autoregressive state-transition prediction and trajectory-level conditional vector-field prediction, and instantiate the latter formulation as F-JEPA.
- Across four control environments, the flow-based dynamics model increases average planning performance and substantially reduces degradation under visual perturbations. Additional analyses over varying perturbation severity and inference settings show that this trend persists across a broad range of evaluation conditions.

## 2 Related Work

**Latent World Models.** Latent world models compress high-dimensional observations into compact state representations and model their temporal evolution for prediction and control. Early approaches typically combine latent dynamics with observation reconstruction and task-specific reward supervision (Ha & Schmidhuber, 2018; Hafner et al., 2019b). The Dreamer family replaces online action search with actor-critic learning over trajectories imagined by the world model (Hafner et al., 2019a; 2020; 2023; 2025). Other generative approaches include IRIS (Micheli et al., 2022), which models discrete visual tokens with an autoregressive Transformer, and DIAMOND (Alonso et al., 2024), which uses diffusion-based observation prediction for Atari control. Recent work has further explored stochastic generative forecasting in fixed pretrained feature spaces (Walker et al., 2025; Porcher et al., 2026). Our focus instead lies in reconstruction-free JEPA world modeling, where we study how the formulation of latent predictive dynamics affects planning robustness.

**Joint-Embedding Predictive Architectures.** Joint-Embedding Predictive Architectures (JEPAs) learn to forecast target embeddings without reconstructing raw observations. I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024; Assran et al., 2025) show that representation prediction can learn strong image and video features without pixel reconstruction. This idea has recently been adopted for world modeling. DINO-WM (Zhou et al., 2024) and PLDM (Sobal et al., 2026) are important examples of JEPA-style world models for control. PLDM learns visual representations end-to-end, whereas DINO-WM models dynamics in a frozen pretrained feature space. Our study builds on LeWorldModel (Maes et al., 2026), which trains a compact end-to-end JEPA world model using a next-embedding prediction objective and SIGReg (Balestriero & LeCun, 2025). F-JEPA adopts the Gaussian-regularized encoder and models latent dynamics with CFM over complete future trajectories, providing a trajectory-level vector-field alternative to autoregressive one-step prediction.

**Flow Matching for Conditional Generation.** Flow matching is a generative modeling framework that learns a time-dependent vector field transporting samples from a source distribution to a target data distribution (Lipman et al., 2022; Liu, 2022; Liu et al., 2022). Compared with direct regression, flow-based objectives provide a flexible way to model continuous conditional distributions and have been applied to domains such as image generation, video generation, and robot action generation (Wan et al., 2025; Black et al., 2024). Recent work has explored replacing uninformed Gaussian initialization with structured source distributions (Jia et al., 2026; Kim et al., 2026), exploiting temporal continuity and strengthening conditional generation. Our work applies CFM to future JEPA latent states, providing a distributional formulation of latent dynamics.

## 3 Rethinking Latent Dynamics in JEPA world models

### 3.1 Stable Representations Do Not Imply Robust Dynamics

Reconstruction-free prediction creates an opportunity to discard visual detail that is unnecessary for modeling environment dynamics (Sun et al., 2026). This opportunity should not, however, be confused with a guarantee of robustness. A JEPA world model can learn a non-collapsed, well-structured representation while its action-conditioned predictor remains sensitive to changes within that representation space.

LeWM (Maes et al., 2026) provides a useful example of this distinction. It jointly trains a visual encoder, which maps an observation $o_t$ to a latent state $z_t = f_\xi(o_t)$, and a dynamics predictor, which forecasts the next latent from the current latent and action. Because next-embedding prediction alone admits a constant collapsed solution, LeWM applies SIGReg (Balestriero & LeCun, 2025) to the encoder embeddings. Let $Z \in \mathbb{R}^{N\times d}$ denote the latent embeddings collected across observations and trajectories. SIGReg samples $M$ unit-norm directions $u^{(m)} \in \mathbb{S}^{d-1}$ and projects the embeddings onto each direction:

$$h^{(m)} = Zu^{(m)}. \tag{1}$$

A univariate Epps-Pulley (Epps & Pulley, 1983) test statistic $\mathcal{T}$ is then applied to each projection,

$$\mathcal{L}_{\mathrm{SIG}}(Z) = \frac{1}{M}\sum_{m=1}^{M} \mathcal{T}\left(h^{(m)}\right). \tag{2}$$

By encouraging the sampled one-dimensional projections to follow a standard Gaussian distribution, SIGReg promotes an approximately isotropic Gaussian embedding distribution and prevents the encoder from mapping all inputs to a constant.

The representation-level role of SIGReg is important, but distinct from constraining the dynamics predictor. In particular, SIGReg regulates the geometry of encoder outputs aggregated across samples; it does not directly specify how an action-conditioned transition should behave when nuisance variation moves an individual observation within that space. If two observations correspond to the same physical state but encode to $z_t$ and $z_t + \delta$, respectively, non-collapse alone does not require the predictor to assign them equivalent future dynamics.

To expose this distinction empirically, we introduce localized Gaussian noise into task-irrelevant background regions while preserving the underlying physical state and environment dynamics. We refer to the resulting observations as *noisy*, in contrast to the original *clean* observations. The exact perturbation formulation is described in Section 5.1.

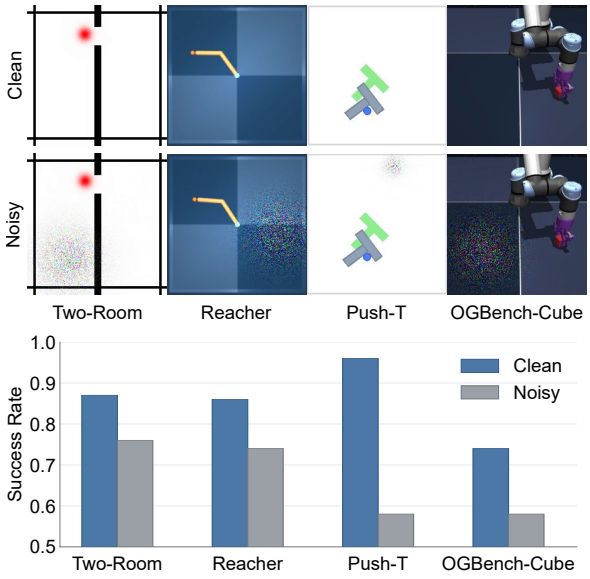



Figure 2: **Task-irrelevant visual perturbations degrade LeWM's latent-space planning.** The $y$-axis starts at 0.5.

Figure 2 illustrates this robustness gap: mean planning success with the autoregressive LeWM baseline decreases from $86\%$ to $67\%$ after the background noise is introduced. This motivates examining the latent-dynamics formulation.

### 3.2 Structural Vulnerabilities of Autoregressive Latent Prediction

Autoregressive predictors construct future trajectories by recursively composing local transitions,

$$\hat{z}_{t+k} = g_\phi(\mathcal{H}_{t+k-1}, a_{t+k-1}), \qquad k = 1, \dots, h, \tag{3}$$

where $\mathcal{H}_{t+k-1}$ denotes the latent history within a fixed-size window. This formulation introduces two structural vulnerabilities to perturbations. First, model-generated latent states progressively enter the history used for subsequent transitions, so deviations introduced at early horizons can propagate through the rollout and affect the terminal state used for planning. Second, the one-step objective

$$\mathcal{L}_{\text{pred}} = \|g_\phi(\mathcal{H}_t, a_t) - z_{t+1}\|_2^2 \tag{4}$$

directly supervises the predictor at observed latent inputs and clean target endpoints, but does not train it on intermediate perturbed states away from those samples (Huang, 2026). If a visual perturbation shifts the encoded observation from $z_t$ to $z_t + \delta$, the transition at the shifted input is not directly specified by the objective. Together, these properties lead to a contrasting formulation that jointly models the future trajectory and trains the dynamics predictor beyond clean latent endpoints.

## 4 Method

### 4.1 Training Objective

We investigate trajectory-level conditional vector-field prediction as an alternative to autoregressive latent dynamics. CFM (Lipman et al., 2022) learns a continuous-time vector field that transports samples from a source distribution $p_0$ toward a target distribution $p_1$. Given a source sample $x_0 \sim \mathcal{N}(0, I)$, a target sample $x_1 \sim p_{\text{data}}$, and flow time $\tau \sim \mathcal{U}[0, 1]$, a commonly used linear path is

$$x_\tau = (1 - \tau)x_0 + \tau x_1, \tag{5}$$

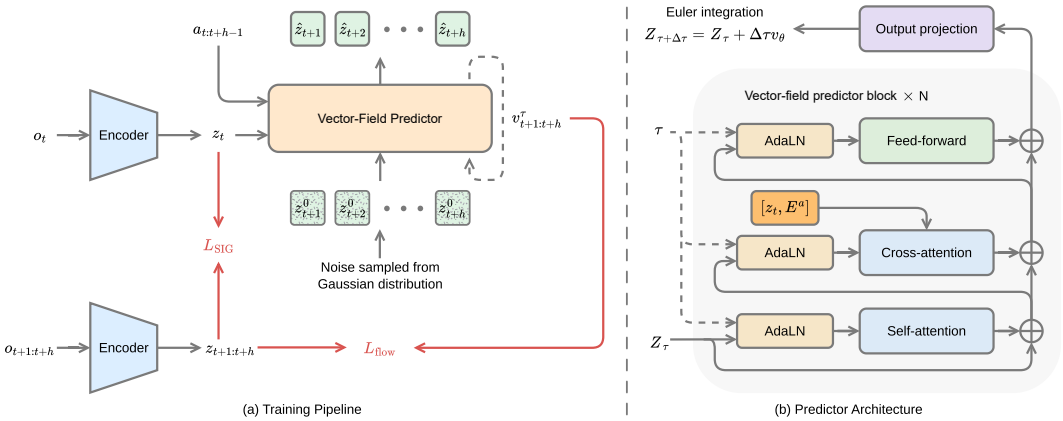



Figure 3: **Overview of F-JEPA. (a)** training pipeline. The conditional flow matching objective trains the predictor to transport Gaussian source tokens toward future latent representations conditioned on the current latent state and action sequence, while SIGReg regularizes the visual encoder to prevent representation collapse. **(b)** predictor architecture. The predictor jointly updates future latent tokens using Transformer blocks with flow-time conditioning and action-dependent context.

whose target velocity is constant along the path,

$$u_\tau(x_\tau \mid x_0, x_1) = x_1 - x_0. \tag{6}$$

The corresponding CFM objective is

$$\mathcal{L}_{\mathrm{CFM}} = \mathbb{E}_{\substack{x_0 \sim \mathcal{N}(0,I),\\ x_1 \sim p_{\mathrm{data}},\\ \tau \sim \mathcal{U}[0,1]}} \left[ \left\| v_\theta(x_\tau, \tau) - (x_1 - x_0) \right\|^2 \right]. \tag{7}$$

Training therefore requires only regression of the vector field at randomly sampled intermediate states and does not require numerical ordinary differential equation (ODE) integration.

We instantiate this formulation for action-conditioned latent world modeling, as illustrated in Figure 3(a). For a prediction horizon of $h$, each training sample contains $h+1$ observations $o_t, o_{t+1}, \ldots, o_{t+h}$ and $h$ temporally aligned action blocks $a_t, a_{t+1}, \ldots, a_{t+h-1}$. Each action block contains 5 consecutive low-level environment actions, matching the frame skip used in the dataset. A shared visual encoder $f_\xi$ maps observations to $d$-dimensional latent states $z_t = f_\xi(o_t) \in \mathbb{R}^d$, while an action encoder $h_\psi$ maps each action block to the same embedding dimension $e_t^a = h_\psi(a_t) \in \mathbb{R}^d$. The future latent trajectory and encoded action sequence are respectively defined as

$$Z = [z_{t+1}, z_{t+2}, \ldots, z_{t+h}] \in \mathbb{R}^{h \times d}, \tag{8}$$

$$E^a = \left[e_t^a, e_{t+1}^a, \ldots, e_{t+h-1}^a\right] \in \mathbb{R}^{h \times d}. \tag{9}$$

For each future position, we independently sample a source latent from a Gaussian distribution,

$$z_{t+i}^0 \sim \mathcal{N}(\mu, \sigma^2 I), \qquad i = 1, \ldots, h, \tag{10}$$

where $\mu \in \mathbb{R}^d$ and $\sigma > 0$ specify the source distribution. The intermediate trajectory is defined as

$$Z_\tau = (1-\tau)Z_0 + \tau Z, \tag{11}$$

where $Z_0 = [z_{t+1}^0, \ldots, z_{t+h}^0]$. Conditioned on the current latent state $z_t$, encoded action sequence $E^a$, and flow time $\tau$, the predictor

$$v_\theta(Z_\tau, \tau, z_t, E^a) \tag{12}$$

estimates the velocity field directing the intermediate trajectory toward the clean target trajectory. Following the standard CFM formulation, the training objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\substack{(z_t, E^a, Z)\sim p_{\mathrm{data}},\ \tau\sim\mathcal{U}[0,1],\\ z^0_{t+i}\overset{\text{i.i.d.}}{\sim}\mathcal{N}(\mu,\sigma^2 I),\ i=1:h}}\left[\left\|v_\theta(Z_\tau, \tau, z_t, E^a) - (Z - Z_0)\right\|^2\right]. \tag{13}$$

The two predictor paradigms differ structurally. Autoregressive predictors construct a trajectory by recursively composing one-step transitions, whereas F-JEPA evolves the complete trajectory jointly through a conditional vector field. In addition, autoregressive regression is supervised at observed latent transitions, while flow matching evaluates the vector field along stochastic source–target interpolation paths.

Finally, we adopt SIGReg to prevent representation collapse. The regularizer is applied to the embeddings of both the current and future observations in each sampled sequence. The complete optimization objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{SIG}}\mathcal{L}_{\mathrm{SIG}}. \tag{14}$$

At inference time, generation starts by drawing a fresh sample from the Gaussian distribution as a source trajectory $Z_0$. Conditioned on $z_t$ and the encoded action sequence $E^a$, the learned vector field defines the ODE

$$\frac{\mathrm{d}Z_\tau}{\mathrm{d}\tau} = v_\theta(Z_\tau, \tau, z_t, E^a). \tag{15}$$

We solve this ODE using explicit Euler integration. Given $N_{\text{flow}}$ flow steps, the integration interval is $\Delta\tau = 1/N_{\text{flow}}$, and the latent trajectory is updated as

$$Z_{\tau+\Delta\tau} = Z_\tau + \Delta\tau\, v_\theta(Z_\tau, \tau, z_t, E^a). \tag{16}$$

Starting from $Z_0$, the final integrated sample $Z_1$ is taken as the predicted future latent trajectory $\hat{Z}$.

### 4.2 Model Architecture

Each observation $o_t$ is encoded by a Vision Transformer (ViT) (Dosovitskiy et al., 2020), and its `[CLS]` representation is projected through a multilayer perceptron (MLP) to obtain the latent state $z_t$. Action blocks are independently mapped to action embeddings by the action encoder. Figure 3(b) illustrates the predictor used by F-JEPA. The dynamics module is a Transformer-based (Vaswani et al., 2017) vector-field predictor. Each future latent token is augmented with a temporal position embedding identifying its prediction horizon. The scalar flow time $\tau$ is first mapped through a sinusoidal embedding and then injected into each Transformer block using adaptive layer normalization (AdaLN) (Peebles & Xie, 2023). The current latent state and encoded action sequence are concatenated as the cross-attention context for future trajectory prediction. Additional implementation details, including the Gaussian source configuration and hyperparameters, are provided in Appendix A.

## 5 Experiments

### 5.1 Experimental Setup

**Planning configurations.** We evaluate two latent-dynamics formulations on four environments: Two-Room (Sobal et al., 2025), Reacher (Tassa et al., 2018), Push-T (Zhou et al., 2024), and OGBench-Cube (Park et al., 2025). See Appendix B.1 for details of the environments and the datasets. We use model-predictive control (MPC) (Richalet et al., 1978; Hansen et al., 2022; 2024) with the cross-entropy method (CEM) (Rubinstein & Kroese, 2004) for planning. At each planning step, CEM optimizes a sequence of 5 action blocks, each containing 5 low-level actions. Each refinement iteration evaluates 300 randomly sampled candidate sequences and updates the action distribution with the best 30. Only the terminal predicted latent state is used to compute the planning cost. The selected five-block sequence is executed before replanning from the new observation. We report success rates over 50 evaluation rollouts. Planning settings follow LeWM except on Push-T, where F-JEPA uses 10 CEM iterations while the reported LeWM baseline uses its original 30-iteration setting. We examine the effect of this difference explicitly in Section 5.3.

**Visual perturbations.** To evaluate robustness independently of the underlying task dynamics, we introduce a spatially localized Gaussian perturbation directly in RGB space. Let $p$ denote a pixel location, $k$ an RGB channel, and $c$ the sampled perturbation center. We draw $\epsilon_{p,k} \sim \mathcal{N}(0, s^2)$ and construct

$$\widetilde{o}_{p,k} = \mathrm{clip}\left(o_{p,k} + \exp\left(-\frac{\|p-c\|_2^2}{2r^2}\right)\epsilon_{p,k}, 0, 255\right), \tag{17}$$

where $s$ controls the noise magnitude and $r$ controls its spatial decay. The center is sampled from the background, away from the agent and task-relevant objects, and remains fixed within each rollout. In the main comparison, we use $s = 100$ and $r = 35$ for Two-Room, Reacher, and OGBench-Cube. For Push-T, we use $r = 10$, since $r = 35$ reduces the LeWM baseline to near-random performance and yields a less informative comparison.

### 5.2 Planning Performance under Clean and Perturbed Observations

**Q1: Does trajectory-level vector-field prediction preserve clean-condition performance?** We first ask whether trajectory-level vector-field prediction compromises planning performance under clean observations relative to autoregressive dynamics. Table 1 reports numerical results, and Figure 1 provides an aggregate comparison with LeWM. F-JEPA matches or exceeds LeWM across all four environments and raises mean success from 86% to 92%. The flow-based dynamics formulation therefore preserves strong clean-condition performance.

Table 1: **Planning success rates (%) under clean visual conditions.** F-JEPA achieves the highest mean success rate (92%) and the best or tied-best performance on three of the four environments, showing that the alternative dynamics formulation preserves strong clean-condition planning performance. Results for PLDM (Sobal et al., 2026), DINO-WM (Zhou et al., 2024), and LeWM are taken from (Maes et al., 2026). The best result in each column is shown in bold.

| Method | Two-Room | Reacher | Push-T | OGBench-Cube | Mean |
|---|---|---|---|---|---|
| PLDM | 97 | 78 | 78 | 65 | 80 |
| DINO-WM | **100** | 79 | 74 | **86** | 85 |
| LeWM | 87 | 86 | **96** | 74 | 86 |
| F-JEPA | **100** | **90** | **96** | 82 | **92** |

**Q2: How do autoregressive and vector-field dynamics differ under visual perturbations?** We next evaluate the central question of the paper: whether the alternative latent-dynamics formulation changes sensitivity to task-irrelevant visual variation. Table 2 reports performance under localized perturbations that leave the task dynamics unchanged. Because this perturbation evaluation is introduced in our study and corresponding results are unavailable for the other baselines, we focus the robustness comparison on F-JEPA and LeWM, which instantiate vector-field and autoregressive latent dynamics, respectively.

Table 2: **Planning success rates (%) under noisy visual conditions.** F-JEPA improves performance across all four environments and increases the mean success rate from 67% to 86%. The final row reports the absolute percentage-point difference from LeWM.

| Method | Two-Room | Reacher | Push-T | OGBench-Cube | Mean |
|---|---|---|---|---|---|
| LeWM | 76 | 74 | 58 | 58 | 67 |
| F-JEPA | **96** | **90** | **90** | **66** | **86** |
| $\Delta$ | +20 | +16 | +32 | +8 | +19 |

F-JEPA improves success in every environment under visual perturbations, raising the mean from 67% to 86%, a difference of 19 percentage points. The mean degradation from clean to noisy observations is 19 points for LeWM and 6 points for F-JEPA. This substantially smaller degradation suggests that the choice between recursive autoregressive prediction and joint vector-field evolution

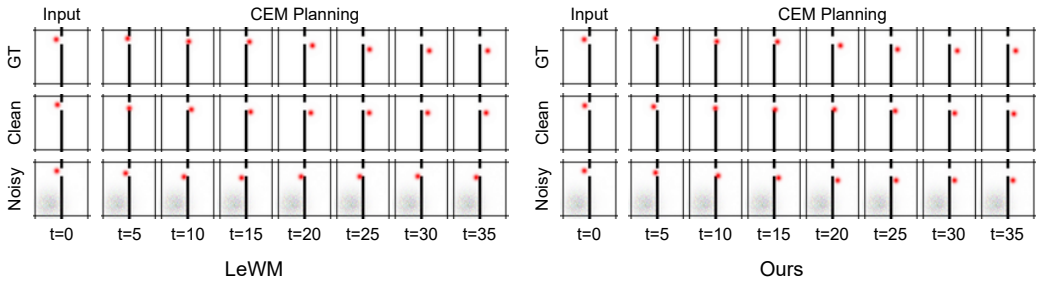



Figure 4: **Qualitative rollouts in Two-Room.** Both LeWM (left) and F-JEPA (right) reach the goal under clean observations, whereas LeWM fails after task-irrelevant background noise is introduced and F-JEPA remains successful, illustrating improved robustness to visual perturbations.

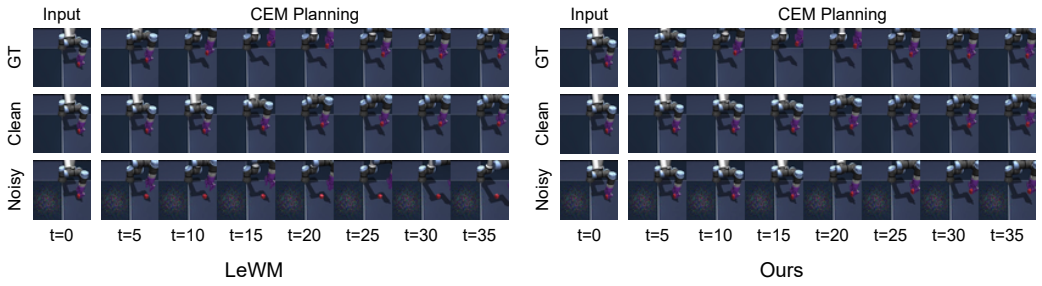



Figure 5: **Qualitative rollouts in OGBench-Cube.** Both LeWM (left) and F-JEPA (right) complete the task under clean observations, whereas LeWM fails after task-irrelevant background noise is introduced and F-JEPA remains successful, illustrating improved robustness to visual perturbations.

can substantially affect planning robustness. These results are consistent with the hypothesis that stochastic trajectory-level training helps the predictor tolerate latent deviations.

The qualitative rollouts in Figures 4 and 5 illustrate the same trend. Both methods succeed under clean observations, while after task-irrelevant visual perturbations are introduced, LeWM fails in both examples whereas F-JEPA remains successful.

### 5.3 Perturbation and Inference Stress Tests

**Q3: Does the robustness advantage persist across perturbation and inference settings?** We examine whether the observed robustness difference is confined to a particular perturbation configuration or inference budget. We vary perturbation radius and magnitude, flow integration steps, and CEM refinement iterations; additional results across environments are provided in Appendix B.2.

**Robustness across perturbation radius.** We first vary the spatial extent of the perturbation on Reacher while holding all other settings fixed. Table 3 shows that F-JEPA consistently outperforms LeWM across tested radii, so the robustness gain is not specific to a single spatial perturbation scale.

Table 3: **Reacher success rates (%) across Gaussian-patch radii.** F-JEPA consistently outperforms LeWM across all tested perturbation radii, indicating that its robustness gain is not specific to a particular spatial noise scale. The noise standard deviation is fixed at $s = 100$, and $\Delta$ denotes the absolute percentage-point improvement over LeWM.

| | **Gaussian-patch radius $r$ (pixels)** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Method** | 10 | 15 | 20 | 25 | 30 | 35 | 40 |
| LeWM | 80 | 84 | 72 | 74 | 68 | 74 | 72 |
| F-JEPA | **88** | **86** | **82** | **88** | **82** | **90** | **84** |
| $\Delta$ | +8 | +2 | +10 | +14 | +14 | +16 | +12 |

Table 4: **Reacher success rates (%) across Gaussian-noise standard deviations.** F-JEPA consistently outperforms LeWM across all tested noise levels and maintains an advantage even at the largest perturbation magnitude. The patch radius is fixed at $r = 35$, and $\Delta$ denotes the absolute percentage-point improvement over LeWM.

| | Noise standard deviation $s$ | | | |
|---|---|---|---|---|
| **Method** | 50 | 100 | 150 | 200 |
| LeWM | 82 | 74 | 66 | 56 |
| F-JEPA | **88** | **90** | **78** | **76** |
| $\Delta$ | +6 | +16 | +12 | +20 |

Table 5: **Push-T success rates (%) across flow integration steps and corresponding inference cost (s).** 8 Euler steps achieve the highest success rates. Times report wall-clock seconds over 50 rollouts; bold denotes the highest success rate within each condition.

| | Clean | | Noisy | |
|---|---|---|---|---|
| **Flow steps** | **Success** | **Time** | **Success** | **Time** |
| 4 | **96** | 31.7 | 82 | 35.0 |
| 8 | **96** | 42.7 | **90** | 47.7 |
| 16 | **96** | 66.9 | 82 | 78.0 |
| 32 | 92 | 111.8 | 84 | 130.2 |

**Robustness across perturbation magnitude.** We next hold the patch radius fixed and vary the noise standard deviation. Table 4 shows that F-JEPA remains stronger across every tested magnitude, including the largest perturbation. Together, the radius and magnitude sweeps show that the observed robustness difference is not confined to a single perturbation configuration.

**Sensitivity to flow integration.** Table 5 studies the accuracy–computation trade-off of ODE integration on Push-T. 8 Euler steps preserve the best clean performance while achieving the highest noisy success rate. Increasing the number of integration steps beyond 8 substantially increases runtime without improving control performance, indicating that finer numerical integration is not the limiting factor in this setting. We therefore use 8 flow steps in the main experiments.

Table 6: **Push-T success rates (%) across CEM iterations.** F-JEPA maintains strong clean performance across all tested planning budgets and achieves its best noisy success rate with only 10 CEM iterations. Under the same 10-iteration budget, F-JEPA substantially outperforms LeWM under noise. Bold denotes the highest success rate for each method and visual condition.

| | Clean | | | Noisy | | |
|---|---|---|---|---|---|---|
| **Method** | 10 | 20 | 30 | 10 | 20 | 30 |
| LeWM | 94 | 92 | **96** | **60** | 56 | 58 |
| F-JEPA | **96** | **96** | **96** | **90** | 88 | 84 |

**Sensitivity to planning budget.** Finally, we examine whether the Push-T improvement can be attributed to differences in CEM planning compute. The main F-JEPA result uses 10 CEM iterations, whereas the reported LeWM baseline retains its original 30-iteration setting. Table 6 evaluates both methods under matched CEM budgets. F-JEPA maintains 96% clean success across the full sweep and achieves its highest noisy success with only 10 iterations. Additional robustness and inference analyses are provided in Appendix B.2.

## 6 Conclusion

We studied how latent-dynamics formulation affects planning robustness in reconstruction-free JEPA world models, focusing on the contrast between autoregressive state-transition prediction and trajectory-level conditional vector-field prediction. Across four control environments, the vector-field formulation substantially reduces degradation under task-irrelevant visual perturbations compared with autoregressive prediction while maintaining strong performance under clean observations. This advantage persists across perturbation strengths and inference settings, highlighting flow-based latent dynamics as a promising alternative to deterministic autoregressive prediction for robust JEPA world modeling.

**Limitations and future work.** Our comparison does not isolate the individual contributions of joint trajectory prediction and stochastic flow training, or their interaction. Controlled intermediate formulations remain important future work. Our evaluation is also limited to relatively short-horizon tasks and primarily localized Gaussian perturbations, motivating studies of longer horizons and broader distribution shifts. Finally, flow inference requires multiple vector-field evaluations, making more efficient flow-based latent prediction another important direction.

### AI USE STATEMENT

In this work, we used generative AI tools to assist with formulating mathematical claims, providing feedback on experiments, implementing methods, and interpreting results. We did not use generative AI tools to develop theoretical models or conceptual frameworks or to propose or refine hypotheses. The following uses were not applicable to this work: generating synthetic datasets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, cleaning or reformatting datasets, and supporting qualitative or thematic data analysis.

Additionally, we used generative AI tools to edit the research paper for improved readability, identify relevant literature, and propose a title for the paper. We reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors, and LLM-generated mathematical claims were examined and refined to ensure that they were correct and aligned with the actual method. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

For reproducibility, Section 4 provides the complete training objective and model architecture. Appendix A documents implementation details, training hyperparameters, Gaussian source configurations, and hardware used for all experiments. The planning and evaluation protocols are specified in Section 5.1, and test environments and datasets are detailed in Appendix B.1. We also analyze key inference-time parameters, including flow integration steps and CEM refinement iterations, to make the reported performance and computational settings directly reproducible.

## REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 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.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Thomas W Epps and Lawrence B Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Yongchao Huang. Vjepa: Variational joint embedding predictive architectures as probabilistic world models. *arXiv preprint arXiv:2601.14354*, 2026.

Jindou Jia, Gen Li, Xiangyu Chen, Tuo An, Yuxuan Hu, Jingliang Li, Xinying Guo, and Jianfei Yang. Action-to-action flow matching. *arXiv preprint arXiv:2602.07322*, 2026.

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*, 2026.

Yann LeCun. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. *Open Review*, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Francois Porcher, Nicolas Carion, Karteek Alahari, and Shizhe Chen. Flow matching in feature space for stochastic world modeling. *arXiv preprint arXiv:2606.29059*, 2026.

J Richalet, A Rault, J Testud, and J Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978.

Reuven Y Rubinstein and Dirk P Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer New York, 2004.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *Advances in Neural Information Processing Systems*, 38:43905–43941, 2026.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Stress-testing offline reward-free reinforcement learning: A case for planning with latent dynamics models. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jacob C Walker, Pedro Vélez, Luisa Polania Cabrera, Guangyao Zhou, Sayna Ebrahimi, Rishabh Kabra, Carl Doersch, Maks Ovsjanikov, João Carreira, and Shiry Ginosar. Frozen forecasting: A unified evaluation. *arXiv preprint arXiv:2507.13942*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.