EpicWorldModel: Exploration-driven Planning with Latent World Models

Bowen Feng[1†] **Julian Ost**[1†] **May Mei**[1] **Anirudha Majumdar**[1] **Felix Heide**[1,2]
[1]Princeton University [2]Torc Robotics

Abstract

Latent world models based on Joint-Embedding Predictive Architecture (JEPA) are deterministic by design. While successful in fully observable scenarios, this paradigm breaks down when past observations and actions lead to multiple plausible future possibilities, e.g., due to occlusion. We introduce *EpicWorldModel*, a framework to train stochastic JEPAs for environments and tasks with inherent uncertainty under partially observability. We jointly train the *EpicWorldModel* predictor with its latent representation space to directly predict multiple potential future states using a flow-matching objective, when the goal-relevant scene content is absent from the conditioning history. We show that flow predictive variance, motivated by its relation to an upper bound on predictive entropy, serves as a useful exploration guidance for planning. By incorporating this uncertainty signal into Cross-Entropy Method (CEM)-based planning, our approach balances goal-reaching with exploration of uncertain regions where occluded goals are most likely to be located. We demonstrate the effectiveness of EpicWorldModel through a series of latent planning experiments with the best or on-par performance across tasks, showing up to 22% empirical improvement in success rate over LeWorldModel. Project page: https://epicworldmodel.com

1 Introduction

Learning world models for latent-space planning has seen increased applications in sequential decision-making, enabling agents to learn compact latent dynamics models and plan directly in the learned representation space [Maes et al., 2026, Zhou et al., 2025, Bardes et al., 2023a, Assran et al., 2025]. Unlike model-free end-to-end visuomotor policy learning, latent-space planning with world models does not directly learn a policy that maps observations to actions. Instead, it learns to compress high-dimensional sensory observations, such as visual inputs, into a latent representation and uses a learned dynamics model to perform model-based planning within this shared latent space.

Joint-Embedding Predictive Architecture (JEPA) models have emerged as a promising family of models for this task. These methods learn predictive representations by mapping observations into an embedding space and predicting future or masked target embeddings from context embeddings, rather than reconstructing raw pixels [LeCun et al., 2022]. This principle has been instantiated in image representation learning through I-JEPA [Assran et al., 2023], video representation learning through V-JEPA [Bardes et al., 2023a, Assran et al., 2025], and motion-aware predictive representation learning through MC-JEPA [Bardes et al., 2023b]. Shown to be promising for representation learning, early world models suffered from collapse [LeCun et al., 2022] when trained without complex training objectives, regularization through exponential moving averages and stop-gradients, the correct hyper-parameters [Sobal et al., 2025] or pre-trained encoders [Zhou et al., 2025]. More recently, LeWorldModel [Maes et al., 2026] stabilizes JEPA world model training with a simple yet effective regularization of the representation space to an isotropic Gaussian distribution.

[†]Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, a key limitation of existing JEPA-based world models is that their prediction objective is often a deterministic single-target regression in the latent space. This formulation can be effective when the future state is largely determined by the historical observation context. The contrary is the case when the conditioned observation history does not contain sufficient information to uniquely determine the future. For example, consider the scenario where a robot is opening a drawer. From the robot's perspective, any reasonable object could be inside the drawer. Further, the ground-truth future state could easily change over time, if the content of the drawer changed. A world model predicting only a single future could lead to planning failure, if the predicted content of the drawer does not match the real world. In such cases, multiple plausible futures may be consistent with the same observation history and the action chunk, but only a subset of them may lead toward the desired goal. This raises a central question: *How can we learn a world model to predict useful future representations when the future is inherently ambiguous?* More broadly, we investigate latent-space planning that relies on predictive uncertainty to explore multiple plausible futures and select action sequences that effectively reach *partially observed or occluded goals.*

To this end, we propose EpicWorldModel: Exploration-based Planning with Latent World Model, a framework that trains a stochastic JEPA model to imagine multiple plausible futures in a jointly learned representation space, and plans under uncertainty, balancing goal-reaching and exploration. At its core, EpicWorldModel learns to predict future observations through sampling from the conditional distribution of possible futures given a set of input actions. In practice, we model this stochastic prediction as a conditional flow-matching process [Pooladian et al., 2023] by predicting the average velocity [Geng et al., 2025a,b] between a random distribution and multiple future representations. Past observations and actions serve as additional conditional signals. Encoder and predictor models are jointly learned [LeCun et al., 2022, Maes et al., 2026] on pairs of past and future frames.

Our key insight is that when the model's uncertainty on the future frames is high, the flow-matching predictor will exhibit high variance in generated future states. Conversely, for future frames that have been observed in recent history, the world model will produce near-deterministic predictions with low variance. We motivate this intuition through an information-theoretic framework, relating flow predictive variance to an upper bound on predictive entropy and serves as an exploration guidance. By incorporating this uncertainty signal into Cross-Entropy Method (CEM) planning, our approach balances goal-reaching with exploration of uncertain regions where occluded or unseen future states are most likely located. We support our hypothesis and approach through both theoretical analysis connecting flow variance to information gain and empirical evaluation on planning tasks. Experimental results show that EpicWorldModel achieves the best or on-par performance among all baselines, outperforming existing latent world models by up to 22% across diverse partially observed tasks. Notably, EpicWorldModel also reaches state-of-the-art performance on visually restricted maze navigation tasks that were previously challenging for latent world models.

2 Related Works

World Models. Recent advances in world modeling has extended state-space dynamics learning to high-dimensional representations such as RGB images [Ha and Schmidhuber, 2018a,b]. Although action-conditioned video generation models have emerged as a popular paradigm for modeling photorealistic evolution of the world [Guo et al., 2025, Quevedo et al., 2025, Zhu et al., 2024, Kim et al., 2026], their pixel reconstruction objective tend to be computationally costly, deeming them impractical for real-time inference and control. In contrast, latent world models aim to learn useful representations of the world as opposed to individual pixels, in particular with joint-embedding predictive architectures (JEPAs) [LeCun et al., 2022]. Video JEPA models [Bardes et al., 2024, Assran et al., 2025] are trained on internet-scale video data to capture temporal structure for future prediction. D-JEPA [Chen et al., 2024a] incorporates diffusion and flow matching objectives into JEPA, extending applicability to image, video, and audio generation. Conditioning on robot actions, models such as DINO-WM [Zhou et al., 2025] and LeWorldModel [Maes et al., 2026] leverage learned representations or enforce structure in the latent space, enabling direct application to robot planning and control.

Flow Matching. Early work on flow matching draws inspiration from Continuous normalizing flows (CNFs) [Chen et al., 2018] originally introduced as a way to model ordinary differential equations (ODEs) from data pairs using black-box solvers. CNFs modeled the underlying vector field v_t of the ODE. As a descendant continuation conditional flow matching (CFM) [Lipman et al., 2022]

methods learned such CNFs for optimal transport between two distributions by only using linear solvers and samples from both distributions This simplicity enabled brought use across data domains. The image generation task has been most successfully solved by denoising diffusion probabilistic models [Song et al., 2021] that stochastic differential equations (SDEs) before. Flow matching offered an appealing alternative achieved by reformulating the denoising process as an ODE. The continuous denoising path of CFMs in contrast to the inherent randomness in the path of SDEs resulted in less denoising steps, later further improved with the rectified flow objective [Liu et al., 2022]. Not only predominant in scaled image generation [Esser et al., 2024] other data domains such as 3D shapes [Xiang et al., 2025], molecules [Du et al., 2024], or robotic planning [Black et al., 2024] have seen successful implementations of generative models using flow matching. Most relevant to out work are video generation models [Wan et al., 2025, HaCohen et al., 2024] that have been continuously extended towards unbounded generation horizon [Huang et al., 2025, Chen et al., 2024b] and action conditioning [Guo et al., 2025] rendering them useful for down-stream world modeling [Team et al., 2026]. Unlike video-generation world models that predict pixels, EpicWorldModel 's stochastic predictor follows the JEPA paradigm by learning an expressive representation space and using a flow ODE to predict future representations.

Uncertainty-Aware Planning. Early work in uncertainty-aware planning grounded intrinsic motivation in prediction error within learned feature spaces [Pathak et al., 2017] and modeled aleatoric and epistemic uncertainty through probabilistic ensemble dynamics for trajectory-sampled planning [Chua et al., 2018]. Plan2Explore [Sekar et al., 2020] demonstrates that ensemble disagreement in latent dynamics models provides an effective intrinsic signal for directed exploration. Subsequent work extends this paradigm to structured world models [Sancaktar et al., 2022], achieving zero-shot transfer to multi-object manipulation. More recently, DreamerV3-XP [Bierling et al., 2025] augmented the DreamerV3 model [Hafner et al., 2023] with ensemble-based reward disagreement as an intrinsic exploration incentive. EpicWorldModel draws insight from the exploration literature, and uses a stochastic JEPA world model for both goal-reaching and curiosity signals.

3 EpicWorldModel

We propose a stochastic latent world model for goal-conditioned planning under partial observability. Instead of regressing to a single deterministic future embedding, our model learns a conditional distribution over future latent representations. This allows planning over multiple plausible latent futures when critical scene is occluded or absent from the current observation history.

3.1 Problem Formulation

We consider goal-conditioned planning in a partially observable Markov decision process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, p, q, r, \gamma)$, where $\mathcal{S}$ is the hidden state space, $\mathcal{O}$ is the observation space, and $\mathcal{A}$ is the action space. At time t, the environment is in an unobserved state $s_t \in \mathcal{S}$ and an observation $o_t \in \mathcal{O}$ is made according to $o_t \sim q(\cdot \mid s_t)$. After the agent executes an action $a_t \in \mathcal{A}$, the environment evolves according to the transition rule $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. Since s_t is not directly observed, the agent must plan from a finite observation-action history.

3.2 Learning the Stochastic Latent Predictive World Model

We build upon latent world modeling frameworks [Maes et al., 2026, Sobal et al., 2025, 2022, Zhou et al., 2025]– more precisely JEPA-based models Maes et al. [2026] – which learn an encoder $\mathcal{E}_\theta$ that maps observations into a compact latent space and a predictor $\mathcal{P}_\varphi$ that models action-induced temporal evolution in this latent space. In its standard form, the predictor is trained to regress a single future embedding, which corresponds to a deterministic approximation of the future latent state.

However, we argue that under partial observability, the future may not be uniquely determined by the current observation history and action, especially when uncertainty is present in the problem. As an example, consider a robot tasked to navigate to the fridge in a kitchen that is initially not within its field of view (Figure 1). In vanilla JEPA, the world model will learn to "memorize" the location of the fridge as they were in the training dataset, e.g., to the robot's right. During inference, given a robot action of turning right, the world model will hallucinate a fridge in the future latent state, even if the fridge is actually to the robot's left [Yin et al., 2025]. This is especially challenging when the

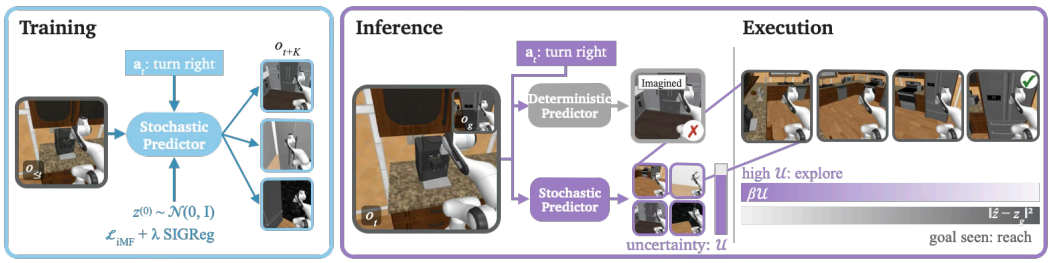


Figure 1: **A robot is asked to navigate to a fridge initially outside its field of view.** A deterministic world model memorizes the location of the fridge from training data and hallucinates it at inference, even when the true location differs. A stochastic model instead maintains a distribution over plausible futures, enabling uncertainty-aware planning. Dark gray framed images indicate observation inputs; colored frames indicate predicted futures. We only show four possibilities at t and the results of a single choice at $t+1$. In practice observations and live in the jointly learned representation space.

goal is absent from the conditioning frames. To model such prediction uncertainty, we replace the deterministic predictor with a conditional generative dynamics model. Given a latent history and a candidate future action chunk, we learn a latent world model that predicts a distribution p_φ over future latent states conditioned on both the latent history and the candidate action chunk as

$$\text{Encoder } \mathbf{z}_t = \mathcal{E}_\theta(o_t), \tag{1}$$

$$\text{Action Embedder } \mathbf{c}_t = \mathcal{C}_\psi(\mathbf{a}_{t:t+K-1}) \tag{2}$$

$$\text{Predictor } \mathbf{z}_{t+1:t+K} \sim \mathcal{P}_\varphi(\mathbf{z}_{t+1:t+K} \mid \mathbf{z}_{t-H+1:t}, \mathbf{c}_t) \tag{3}$$

where H is the history length and K is the planning horizon.

Conditional Flow Matching in Gaussian Latent Space. To learn a $\mathcal{P}_\varphi$ that predicts a distribution over future latent states, we adopt the conditional flow matching (CFM) paradigm [Lipman et al., 2022]. Flow matching learns a time-dependent velocity v_φ of a vector field that transports samples from a source distribution, e.g., a Gaussian distribution $\mathcal{N}(0, I_d)$, to a target distribution. In our case, the target distribution consists of latent future observations $\mathbf{z}_{t+1:t+K}$ given the latent history $\mathbf{z}_{t-H+1:t}$ and action chunk $\mathbf{c}_t$.

Let $\mathbf{z}^{(1)} = \mathbf{z}_{t+1:t+K}$ be the target future latent block and $\mathbf{z}^{(0)} \sim \mathcal{N}(0, I)$ be a Gaussian source sample of the same shape. We can now define a straight-line interpolation Liu et al. [2022] for flow step $\tau \in [0, 1]$, as well as the corresponding conditional velocity target as

$$\mathbf{z}^{(\tau)} = (1-\tau)\mathbf{z}^{(0)} + \tau\mathbf{z}^{(1)}, \qquad v_{\text{tgt},\tau}(\mathbf{z}^{(\tau)} \mid \mathbf{z}^{(1)}) = \mathbf{z}^{(1)} - \mathbf{z}^{(0)}. \tag{4}$$

Training v_φ to match this per-sample target leads to an unbiased estimator, allowing for scalable training without marginalizing over all source-target pairs Lipman et al. [2022]. This leads to the flow matching objective

$$\mathcal{L}_{\text{FM}} = \mathop{\mathbb{E}}_{\tau, \mathbf{z}^{(0)}, (\mathbf{x}_t, \mathbf{c}_t, \mathbf{z}_{t+1:t+K})} \left\| v_\varphi(\mathbf{z}^{(\tau)}, \tau, \mathbf{x}_t, \mathbf{c}_t) - \left(v_{\text{tgt},\tau}(\mathbf{z}^{(\tau)} \mid \mathbf{z}^{(1)}) \right) \right\|_2^2. \tag{5}$$

Average Velocity Prediction for 1-Step Inference. Flow matching comes with an inherently higher complexity than deterministic prediction at inference time caused by a high number of function evaluations (NFE) in each step of the Euler sampling process $\mathbf{z}^{(\tau+\Delta\tau)} = \mathbf{z}^{(\tau)} + \Delta\tau \cdot v_\varphi$.

More recent models [Lu et al., 2026, Geng et al., 2025b,a] characterize the underlying flow field by its average velocity

$$u\left(\mathbf{z}^{(\tau)}, r, \tau\right) \triangleq \frac{1}{(s-r)} \int_r^t v(z^{(\omega)}) d\omega \tag{6}$$

between two timesteps τ and r with $\tau \leq t$ instead of the instantaneous velocity $v(\cdot, \tau)$ at a single step s. After applying the derivative w.r.t. s to Eq (6) and following the rewriting in improved MeanFlow (iMF) Geng et al. [2025b] we get rid of the integral and derive the instantaneous velocity v in Equation (4) from the average velocity as

$$v(z^{(\tau)}, \cdot) = u(z^{(\tau)}, \cdot) + (\tau - r)\frac{d}{ds} u(z^{(\tau)}, \cdot). \tag{7}$$

This yields the learning objective

$$\mathcal{L}_{iMF} = \mathbb{E}\,\|V_\varphi - v_{\text{tgt}}\|_2^2\,, \tag{8}$$

$$V_\varphi = u_\varphi(\mathbf{z}^{(\tau)}, \mathbf{x}_t, \mathbf{c}_t, r, \tau) + (\tau - r)\cdot \text{JVP}_{\text{sg}}\left(u_\varphi; v_\varphi\right), \text{ with } v_\varphi(\mathbf{z}^{(\tau)}, \tau) \equiv u_\varphi(\mathbf{z}^{(\tau)}, \tau, \tau) \tag{9}$$

and JVP_{sg} is the Jacobian-Vector product approximation of $\frac{d}{d\tau}u(\mathbf{z}^{(\tau)}) = \partial_{\mathbf{z}}u(\mathbf{z}^{(\tau)})v(\mathbf{z}^{(\tau)}) + \partial_\tau u(\mathbf{z}^{(\tau)})$ with a stop-gradient sg applied. While the added complexity during training leads to higher complexity at training time, the inference time can be collapsed to a single function evaluation (1-NFE) by choosing $\tau = 0$ and $r = 1$.

Isotropic Gaussian Latent Regularization. We additionally regularize the latent embeddings to follow an isotropic Gaussian distribution using the Sketched Isotropic Gaussian Regularizer (SIGReg) [Balestriero and LeCun, 2025] in line with Maes et al. [2026]. SIGReg leverages the Cramér-Wold theorem: a multivariate distribution matches a target if all its one-dimensional projections match that target. Concretely, latent embeddings $\mathbf{Z}$ are projected onto M random unit-norm directions $u^{(m)} \in \mathcal{S}^{d-1}$, and the Epps-Pulley normality test statistic $T(\cdot)$ is applied to each projection $h^{(m)} = \mathbf{Z}u^{(m)}$

$$\text{SIGReg}(\mathbf{Z}) \coloneqq \frac{1}{M}\sum_{m=1}^{M} T(h^{(m)}). \tag{10}$$

Because the Epps-Pulley statistic tests against $\mathcal{N}(0,1)$ on each projection, convergence of SIGReg implies $p(z_t) \approx \mathcal{N}(0, I_d)$ at each timestep. The complete EpicWorldModel training objective is given by

$$\mathcal{L}_{\text{EpicWorldModel}} \coloneqq \mathcal{L}_{\text{iMF}} + \lambda\,\text{SIGReg}(\mathbf{Z}), \tag{11}$$

jointly supervising stochastic prediction of futures and a well defined latent representation.

Theoretical Properties. The Gaussian latent geometry induced by SIGReg confers several useful properties on the flow matching objective. We state these as formal propositions and provide proofs in Appendix A.

Remark 1 (Source-Target Compatibility)**.** *SIGReg encourages the latent space marginal distribution $\mathbb{P}_{\mathbf{Z}_t} \to \mathcal{N}(0, I_d)$ at each timestep t. With flow matching, the source distribution $p_0 = \mathcal{N}(0, I_d)$ matches the latent space marginal distribution, leading to integrations always beginning in regions well-supported by the data distribution. This can lead to shorter and better conditioned transport paths in flow matching.*

In CFM, the same interpolated point $\mathbf{z}^{(\tau)}$ can be reached from many different $(\mathbf{z}^{(0)}, \mathbf{z}^{(1)})$ pairs. Given only $\mathbf{z}^{(\tau)}$, the network is asked to predict a target $v_{\text{tgt},\tau}$ that is not unique, leading to irreducible gradient variance Pooladian et al. [2023], Tong et al. [2023]. However, since the latent space is encouraged to be $\mathcal{N}(0, I_d)$ (where d is the dimensionality), we can show that the gradient noise is bounded and data-independent.

Proposition 1 (Data-Independent Gradient Noise)**.** *Given an isotropic Gaussian latent space with $\mathbb{P}_{\mathbf{Z}_t} = \mathcal{N}(0, I_d)$, for every $\tau \in [0,1]$, the excess variance of the conditional velocity target is given by*

$$\text{Var}\left(v_{\text{tgt},\tau} \mid \mathbf{z}^{(\tau)}\right) = \frac{1}{(1-\tau)^2 + \tau^2} I_d. \tag{12}$$

This quantity depends only on the latent dimensionality d and the flow step τ (isotropic, $\propto I_d$). Therefore, gradient variance is bounded and independent of data, and does not grow as the latent representation becomes more expressive or the conditioning becomes more complex.

We see smooth convergence as a result of Proposition 1 in the validation loss across individual loss components in Equations (6) and (11). A detailed analysis can be found in the Appendix.

3.3 Uncertainty-Aware Latent Space Planning

Since EpicWorldModel outputs stochastic future predictions, we can approximate a group of sampled outputs as an ensemble, used to quantify the uncertainty in predicted future states. Specifically, given history $\mathbf{x}_t$ and action chunk embedding $\mathbf{c}_t$, we sample N noise vectors $\mathbf{z}^{(0)}[i] \sim \mathcal{N}(0, I_d)$

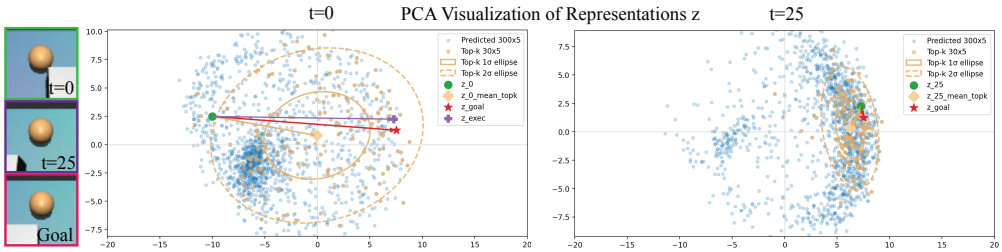


Figure 2: **Latent uncertainty contraction during planning on a single rollout.** Each panel shows the top-k latent predictions in a shared PCA space, together with the current latent ($\mathbf{z}_0$ or $\mathbf{z}_{25}$), the fixed goal latent ($\mathbf{z}_{\text{goal}}$), and the top-k covariance ellipse. The red arrow measures $\|z_t - z_{goal}\|$

and run forward pass with the predictor $\mathcal{P}_\varphi$ for each noise sample to acquire an ensemble of N members, $\{\hat{\mathbf{z}}^{(1)}[i]\}_{i=1}^{N}$. Then, we use the disagreement between the ensemble members as a signal of uncertainty of the future state, computed as

$$\mathcal{U} = \frac{1}{d}\sum_{j=1}^{d} \widehat{\text{Var}}_i(\hat{\mathbf{z}}_j^{(1)}[i]), \tag{13}$$

where j is the latent feature dimension. We motivate this choice of this disagreement signal via the following proposition.

Proposition 2 (Uncertainty as Conditional Entropy Upper Bound)**.** *Let $\Sigma_{\mathbf{c}_t}$ denote the true predictive covariance for an action chunk embedding $\mathbf{c}_t$. For a well-trained $\mathcal{P}_\varphi$, $\mathcal{U}$ is an unbiased estimator of the mean conditional variance*

$$\mathbb{E}[\mathcal{U}] \;=\; \frac{\text{tr}(\Sigma_{\mathbf{c}_t})}{d}. \tag{14}$$

By the maximum entropy theorem, the predictive entropy is upper-bounded by a function of $\mathcal{U}$ as

$$H(\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t) \;\le\; \frac{d}{2}\log(2\pi e \cdot \mathcal{U})\,. \tag{15}$$

Furthermore, $\mathcal{U}$ is bounded within $[0, 1]$, providing an interpretable absolute scale. $\mathcal{U} \approx 0$ indicates a confident, concentrated prediction, while $\mathcal{U} \approx 1$ indicates predictions as diffuse as the marginal prior. The bound in Eq. 15 is tight only for $\Sigma_{c_t} \propto I_d$ (App. A, Eq. 31); for anisotropic covariances it can be loose, so we use $\mathcal{U}$ as a practical exploration bonus rather than an entropy estimate.

Connection to Expected Information Gain. Under partial observability, the agent must explore to reduce its uncertainty about unobserved future states. A principled exploration objective is to maximize the Expected Information Gain (EIG) [Houlsby et al., 2011, Gal et al., 2017], i.e., the amount of information gained from taking an action chunk $\mathbf{a}_{t:t+K-1}$ and observing the outcome $\mathbf{z}$. We motivate our uncertainty measure $\mathcal{U}$ via EIG through the following remark.

Remark 2 (Connection to Expected Information Gain)**.** *In exploration, we seek to gain more information about the future latent state $\mathbf{z}$. Given the flow matching source distribution, we have a prior over $\mathbf{z}^{(0)}$ given by $\mathcal{N}(0, I_d)$. We can interpret sampled pure noises as different "seeds" of the world. In the Bayesian active learning literature [Houlsby et al., 2011, Gal et al., 2017], EIG measures mutual information between the seed and potential future latent states*

$$\text{EIG}(\mathbf{c}_t) = I(\mathbf{z}^{(0)}; \mathbf{z}^{(1)} | \mathbf{x}_t, \mathbf{c}_t) = H[\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t] - H[\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t, \mathbf{z}^{(0)}]. \tag{16}$$

Since the flow ODE is deterministic given $\mathbf{z}^{(0)}$, the per-seed entropy $H[\hat{\mathbf{z}}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t, \mathbf{z}^{(0)}] = 0$. Therefore,

$$\text{EIG}(\mathbf{c}_t) \;=\; H\Big[\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t\Big] \;\lesssim\; \frac{d}{2}\log(2\pi e \cdot \mathcal{U}), \tag{17}$$

where the inequality is from Proposition 2. Therefore, $\mathcal{U}$ is a proxy motivated by EIG, and maximizing $\mathcal{U}$ approximately maximizes an upper bound on EIG.

Uncertainty-guided Planning Objective. At inference time, we optimize over action chunk embeddings using Model Predictive Control (MPC) with the Cross-Entropy Method (CEM)

$$\mathbf{c}_t^\star \;=\; \arg\min_{\mathbf{c}_t} \sum_{i}^{N} \Big[\|\hat{\mathbf{z}}_{t+K}[i] - \mathbf{z}_g\|^2\Big] \;-\; \beta\,\mathcal{U}(\mathbf{c}_t), \tag{18}$$

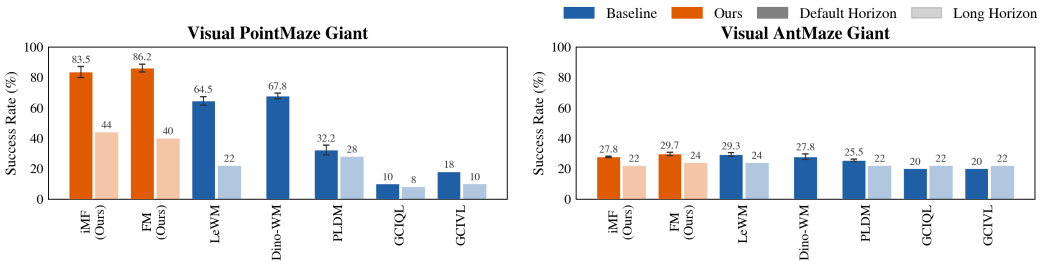


Figure 3: **Success rate comparison across benchmark environments.** FM and IMF represent EpicWorldModel with the flow matching objectives introduced in Equation (5) (FM) and Equation (6) (iMF). EpicWorldModel achieves the best performance in both the default- and long-horizon settings.

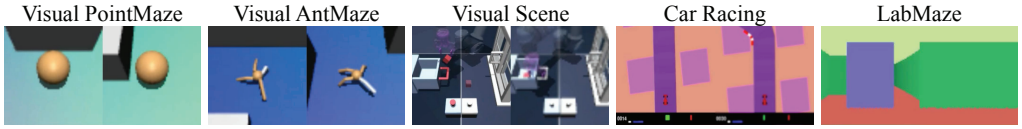


Figure 4: **Evaluation Tasks.** In each of the five tasks the most left image represents the initial observation o_0 and the right a goal observation. All tasks are POMDP evaluations in existing benchmarks. The first three tasks are available in OGbench Park et al. [2025], Car Racing is available in Gymnasium Towers et al. [2024], and LabMaze in DMLab Beattie et al. [2016].

where $\mathbf{z}_g = \mathcal{E}_\theta(o_g)$ is the goal embedding and $\beta \geq 0$ balances exploration and exploitation (goal-reaching).

4 Experiments

We evaluate EpicWorldModel across Visual PointMaze Giant, Visual AntMaze Giant, and Visual Scene, which are representative OGBench tasks with limited fields of view. We also evaluate on Car Racing and LabMaze, which similarly involve partial observability and restricted visual observations. We visualize the five evaluation environments in Figure 4. We further evaluate EpicWorldModel in a more complex and complex 3D robotics benchmark by providing experiments on RoboCasa NavigateKitchen tasks. Across NavigationKirchen we additionally control difficulty and occlusion level to further analyze benefits of EpicWorldModel's planning capabilities under uncertainty. We compare against goal-conditioned offline RL methods that learn value functions for goal reaching (GCIVL [Park et al., 2025], GCIQL [Kostrikov et al., 2021]) and against latent world-model planners, namely PLDM [Sobal et al., 2022], DINO-WM [Zhou et al., 2025] on frozen DINOv2 features, and the JEPA-based LeWorldModel [Maes et al., 2026]. Detailed implementations and inference protocal can be found in Appendix B.

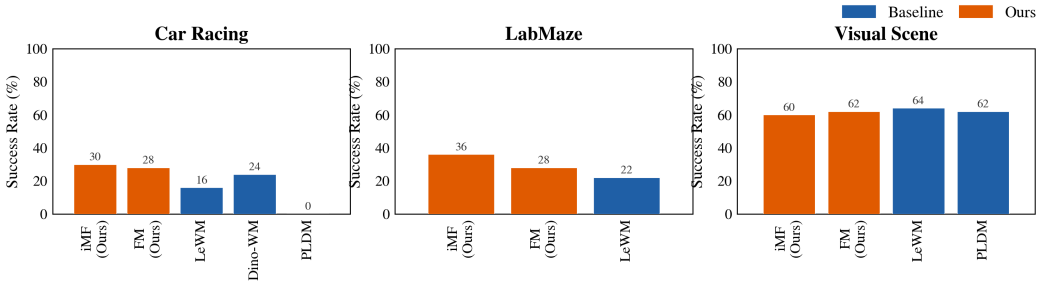
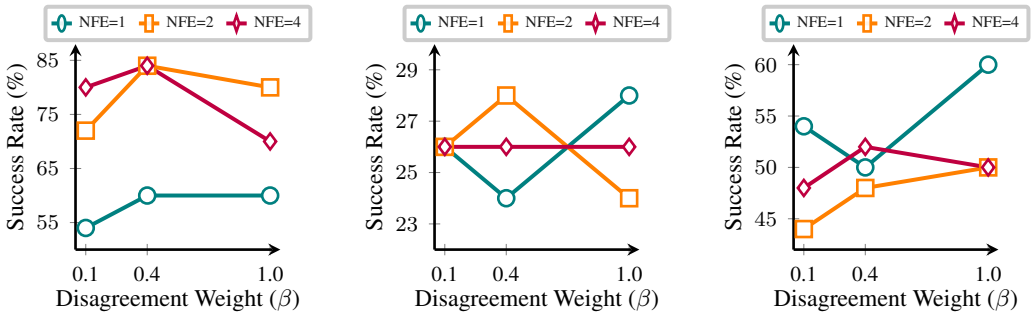


Figure 5: **Success rate comparison across benchmark environments.** EpicWorldModel achieves competitive planning success rate across diverse navigation task under partially observable scenes and is on-par for complex manipulation in the Visual Scene.



(a) PointMaze Giant (b) AntMaze Giant (c) Visual Scene

Figure 6: **Ablation study on disagreement weight across environments. (a)** Visual PointMaze Giant shows clear sensitivity to hyperparameters with optimal performance at $\beta = 0.4$ and Number Flow Evolutions(NFE)=2. **(b)** AntMaze Giant demonstrates more stable performance across settings, with modest gains from moderate exploration. **(c)** Visual Scene shows NFE values (NFE=1) achieve better performance.

4.1 Planning Performance of EpicWorldModel

In Figure 3, we compare our EpicWorldModel with prior goal-conditioned and world-model baselines on two large-scale partially observable navigation tasks. On Visual PointMaze Giant, EpicWorldModel achieves the best performance across all methods, especially we have $\approx 22\%$ more success rate than LeWorldModel. IMF reaches a success rate of $84\% \pm 3.6$, while FM further improves to $86\% \pm 2.6$ vs. LeWM 64.5 ± 2.9 and DINO-WM 67.8 ± 1.8. These results substantially outperform prior model-based baselines such as LeWM, Dino-WM and PLDM, as well as goal-conditioned policy baselines such as GCIQL and GCIVL. This result suggests that explore-driven latent planing is more effective in visually constrained maze environments.

On Visual AntMaze Giant, the task becomes more challenging due to higher-dimensional action spaces. In this setting, FM achieves 30% success and iMF achieves 28%, achieving on-par performance with LeWM, DINO-WM and PLDM. Across both environments, the long-horizon setting, which doubles the target planning horizon, consistently reduces performance.

We further evaluate EpicWorldModel on Car Racing, LabMaze, and Visual Scene, which also involve partial observability and limited visual context. Planning success rate results are summarized in Figure 5. Consistent with the navigation results, EpicWorldModel performs strongly on Car Racing, where iMF and FM achieve 30% and 28% success, respectively, almost doubled the success rate of LeWM (16.0%) and above DINO-WM (24.0%). PLDM fails out-of-regime. On LabMaze, iMF again achieves the best performance. On Visual Scene, our methods remain competitive, though the smaller gap suggests that flow-based latent prediction is most effective in partially observable tasks with relatively low-dimensional action spaces.

The gain is not explained by inference compute: giving LeWM 6000 CEM candidates (20× default) raises PointMaze success only to 74.0% (App. C.2), still ≈ 10 points below iMF; on AntMaze extra samples do not help (24.0–26.5%).

4.2 Planning Performance Under Controlled Uncertainty Levels in RoboCasa

We further evaluate EpicWorldModel on the RoboCasa NavigateKitchen task under controlled levels of uncertainty across general task difficulty and goal observability. We vary the robot's initial distance to the goal, which is moderately correlated with initial goal visibility ($\rho = -0.53$). Across this sweep, the goal is often partially visible or occluded at the start. Figure 7 shows representative EpicWorldModel rollouts in which the fridge is initially outside the robot's field of view or occluded by kitchen fixtures, and the robot must explore before the goal becomes visible. We compare the deterministic LeWM baseline against our stochastic predictor with exploration disabled ($\beta = 0$) and enabled ($\beta = 0.1$, the default weight, not tuned for this task) and LeWM with 3× more CEM candidates (LeWM 3×) to control for sampling budget. We evaluate 15 episodes at each distance, for 60 episodes per method. For RoboCasa all models (LeWM and EpicWorldModel variants) share a frozen DINOv3-ViT-S/16 encoder (App. B)

Table 1: **Controlled difficulty analysis on RoboCasa NavigateKitchen.** Success rate (%) across increasing start distances. Best results are shown in **bold**, and the second-best average is underlined. (iMF NFE=2; FM NFE=5; LeWM 3× n = 8–12)

Start Dist.	**LeWM**	**LeWM** $3\times$	**iMF** ($\beta = 0$)	**iMF** ($\beta = .1$)	**FM** ($\beta = .1$)
0.50 m	60.0	93.3	**100.0**	**100.0**	**100.0**
0.92 m	46.7	75.0	80.0	80.0	**86.7**
1.26 m	53.3	40.0	46.7	**80.0**	**80.0**
1.46 m	20.0	25.0	60.0	60.0	**93.3**
Average	45.0	58.3	71.7	<u>80.0</u>	**90.0**

Table 2: **Occlusion/visibility analysis on RoboCasa NavigateKitchen.** Successes / episodes (pooled %) stratified by initial goal visibility. Lower visibility indicates stronger partial observability. Best results are shown in **bold**. (iMF NFE=2; FM NFE=5; LeWM 3× n = 8–12)

Initial Visibility	**LeWM**	**iMF** ($\beta = 0$)	**iMF** ($\beta = .1$)	n
< 0.5	0/6	1/6	**5/6**	6
0.5–0.8	2/7	5/7	**6/7**	7
0.8–0.95	1/4	**2/4**	1/4	4
≥ 0.95	10/22	**21/22**	19/22	22
Pooled	13/39 (33.3)	29/39 (74.4)	**31/39 (79.5)**	39

Controlled Difficulty. Three trends emerge from experiments documented in Table 1. First, under matched planning compute (the same CEM search budget), the stochastic predictor without exploration already substantially outperforms deterministic LeWM, achieving 71.7% versus 45.0% success rate when pooled across distances. Second, this gain cannot be explained solely by a larger sampling budget: increasing LeWM to $3\times$ more candidates improves short-range performance but provides little benefit at larger distances. Third, the exploration bonus becomes particularly useful when the goal cost is less informative. At 1.26 m, enabling exploration increases iMF success from 46.7% to 80.0%, while leaving easier short-range settings unchanged. The FM variant performs best at the largest distance, reaching 93.3% success compared with 20.0% for LeWM.

Controlled Occlusions/Visibility. Distance, however, is only an indirect proxy for partial observability. We therefore additionally measure the fraction of the goal visible from the robot's initial observation. Specifically, we sample approximately 200 points on the goal fixture geometry and compute the fraction that lies within the camera frustum and is not occluded by scene geometry. The resulting score ranges from 0 (fully hidden or outside the field of view) to 1 (fully visible and unobstructed). This geometric measure requires no learned model or tuned threshold and is additionally verified using segmentation masks and manual inspection.

The occlusion-controlled RoboCasa experiments in Table 2 provide preliminary evidence that uncertainty-driven exploration yields information about the hidden goal location. When the goal is initially occluded (< 0.5), exploration substantially improves success ($1/6 \rightarrow 5/6$). This benefit decreases as visibility increases and can even reduce goal-reaching performance when the goal is fully observable (≥ 0.95; $21/22 \rightarrow 19/22$). These results suggest that the variance bonus is useful for goal discovery under partial observability but unnecessary once the goal is visible.

4.3 Ablation Study on the choice of β and the Number of Flow Evaluations (NFE)

We conduct an ablation study to understand the sensitivity of our approach to key hyperparameters: the disagreement weight β, which controls the exploration-exploitation trade-off, and the Number of Flow Evaluations (NFE) during planning, which determines the computational budget for inference.

We evaluate three representative environments from our benchmark suite: Visual PointMaze Giant, Visual AntMaze Giant, and Visual Scene. For each environment, we conduct an ablation over the disagreement weight $\beta \in \{0.1, 0.4, 1.0\}$ and the number of flow evaluations $\mathrm{NFE} \in \{1, 2, 4\}$. Figure 6 a–c summarize the sensitivity of performance to these two hyperparameters across environments.

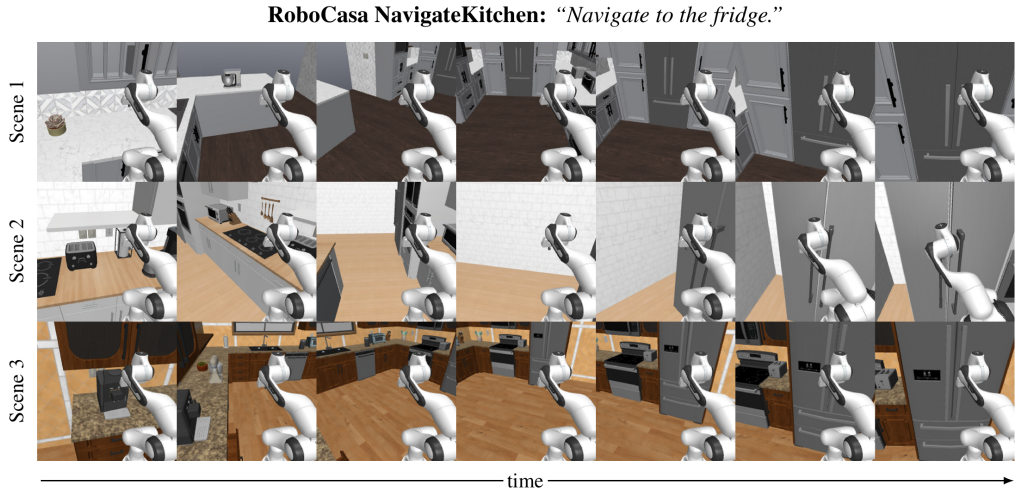


Figure 7: **Qualitative rollouts of EpicWorldModel on RoboCasa NavigateKitchen.** Each row shows an episode in a different kitchen layout from the validation scene split. The initial visibility is < 0.5 for all episodes, only displaying the hardest tasks. All episodes are conditioned on the same instruction, and frames are ordered left to right in time. At the start of each episode the fridge is outside the field of view or occluded by cabinets and counters. Guided by the uncertainty cost $\mathcal{U}$, the robot first turns and moves toward unobserved regions of the kitchen, then commits to the goal once the fridge becomes visible. All shown results are from an iMF model ($\beta = 0.1$, NFE $= 2$).

The results indicate that hyperparameter selection should be adapted to the structure and dimensionality of each task. Across most environments, $\beta = 0.4$ and $\text{NFE} = 2$ achieve a favorable trade-off between uncertainty-driven exploration and computational cost. In contrast, the Visual Scene results suggest that, in high-dimensional action spaces, single-step prediction combined with a larger exploration weight may provide a more effective planning strategy. Automatically adapting β is left to future work.

4.4 Analysis on Exploration Behavior in Latent Space Planning

Figure 2 provides a qualitative view of the top-k and the full cross entropy samples' latent predictions for a single test case. The full cross-entropy sample predictions are shown in blue and the top-k selections in yellow. We project the $D = 192$ latent embeddings $\mathbf{z}_t$ to the PCA 2-Dimensional space for analysis and visualization. At the first planning point ($t = 0$), the elite latent cloud is broad, reflecting substantial uncertainty over future outcomes when the agent is still far from the goal. After executing the first plan, the executed rollout reaches a latent state close to the goal within the first 25-step plan. By the second planning point ($t = 25$), the elite cloud has contracted markedly around the current and goal-adjacent region, with the top-k ellipse area shrinking from 60.99 to 11.86. This example is consistent with the hypothesis that predictive uncertainty is larger early in planning and decreases once the agent approaches the goal. Furthermore, the observed mean of the top-k predictions points in the direction toward the goal, suggesting that the predictive variance of the stochastic predictor is consistent with goal-directed progress rather than random noise.

5 Conclusion

We present EpicWorldModel, an uncertainty-aware planning framework leveraging stochastic JEPAs based on flow matching. Through theoretical motivation and experimental validation, we show that EpicWorldModel is a stable JEPA capable of achieving high planning success rates, especially in environments under partial observability. Although our results demonstrate the promise of exploration-driven planning with latent world models, several limitations remain. In particular, goal-image conditioning can be restrictive, as many real-world tasks are more naturally specified through language and goal images may provide privileged information unavailable at deployment. Extending the framework to align visual observations with language-based or abstract goal specifications is therefore an important direction for future work.

Acknowledgments and Disclosure of Funding

Felix Heide is a co-founder of Algolux (now Torc Robotics), Head of AI at Torc Robotics, and a co-founder of Cephia AI. The other authors declare no competing interests.

References

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.

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, pages 79115–79135. PMLR, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *arXiv preprint arXiv:2404.08471*, 2023a.

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.

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

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*, pages 15619–15629, 2023.

Adrien Bardes, Jean Ponce, and Yann LeCun. Mc-jepa: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698*, 2023b.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Stress-testing offline reward-free reinforcement learning: A case for planning with latent dynamics models. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

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

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018b.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation, 2025. URL `https://arxiv.org/abs/2506.00613`.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: Learning interactive real-robot action simulators. *arXiv preprint arXiv:2406.14540*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

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

Dengsheng Chen, Jie Hu, Xiaoming Wei, and Enhua Wu. Denoising with a joint-embedding predictive architecture. *arXiv preprint arXiv:2410.03755*, 2024a.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *9th International Conference on Learning Representations, ICLR 2021*, October 2021.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21469–21480, 2025.

Yuanqi Du, Arian R Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, 6(6):589–604, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

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.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024b.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious exploration via structured world models yields zero-shot object manipulation. In *Advances in Neural Information Processing Systems*, 2022.

Lukas Bierling, Davide Pasero, Jan-Henrik Bertrand, and Kiki Van Gerwen. DreamerV3-XP: Optimizing exploration through uncertainty estimation. *arXiv preprint arXiv:2510.21418*, 2025.

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

Vlad Sobal, Jyothir SV, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features. *arXiv preprint arXiv:2211.10831*, 2022.

Tenny Yin, Zhiting Mei, Tao Sun, Lihan Zha, Emily Zhou, Jeremy Bao, Miyu Yamane, Ola Shorinwa, and Anirudha Majumdar. Womap: World models for embodied open-vocabulary object localization. *In 9th Annual Conference on Robot Learning (CoRL)*, 2025.

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. *arXiv preprint arXiv:1612.03801*, 2016.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

A Proofs

Proposition 1 (Data-Independent Gradient Noise)**.** *Given an isotropic Gaussian latent space with $\mathbb{P}_{\mathbf{Z}_t} = \mathcal{N}(0, I_d)$, for every $\tau \in [0, 1]$, the excess variance of the conditional velocity target is given by*

$$\mathrm{Var}\left(v_{\mathrm{tgt},\tau} \mid \mathbf{z}^{(\tau)}\right) = \frac{1}{(1-\tau)^2 + \tau^2} I_d. \tag{12}$$

This quantity depends only on the latent dimensionality d and the flow step τ (isotropic, $\propto I_d$). Therefore, gradient variance is bounded and independent of data, and does not grow as the latent representation becomes more expressive or the conditioning becomes more complex.

Proof. By the law of total variance applied to the conditional velocity target $v_{\text{tgt}} = \mathbf{z}^{(1)} - \mathbf{z}^{(0)}$:

$$\mathrm{Var}(v_{\text{tgt}}) = \mathbb{E}_{\mathbf{z}^{(\tau)}}[\mathrm{Var}(v_{\text{tgt}} \mid \mathbf{z}^{(\tau)})] + \mathrm{Var}(\mathbb{E}[v_{\text{tgt}} \mid \mathbf{z}^{(\tau)}]). \tag{19}$$

On the left hand side, by independence of $\mathbf{z}^{(0)}$ and $\mathbf{z}^{(1)}$, we have:

$$\mathrm{Var}(v_{\text{tgt}}) = \mathrm{Var}(\mathbf{z}^{(1)} - \mathbf{z}^{(0)}) = \sigma^2 I_d + I_d = (\sigma^2 + 1) I_d, \tag{20}$$

where we denote the variance of the SIGReg-induced latent space by σ^2. On the right hand side, $\mathbb{E}[v_{\text{tgt}} \mid \mathbf{z}^{(\tau)}] = v_{\text{tgt}}(\mathbf{z}^{(\tau)})$ is the marginal vector field Lipman et al. [2022]. Since $\mathbf{z}^{(\tau)} = (1-\tau)\mathbf{z}^{(0)} + s\mathbf{z}^{(1)}$, the joint covariance of $(\mathbf{z}^{(1)} - \mathbf{z}^{(0)},\, \mathbf{z}^{(\tau)})$ is given by:

$$\mathrm{Cov}(\mathbf{z}^{(1)} - \mathbf{z}^{(0)},\, \mathbf{z}^{(\tau)}) = \tau \,\mathrm{Var}(\mathbf{z}^{(1)}) - (1-\tau)\,\mathrm{Var}(\mathbf{z}^{(0)}) = (\tau\sigma^2 - (1-\tau)) I_d, \tag{21}$$

$$\mathrm{Var}(\mathbf{z}^{(\tau)}) = (1-\tau)^2 I_d + \tau^2\sigma^2 I_d = [(1-\tau)^2 + \tau^2\sigma^2] I_d. \tag{22}$$

The marginal vector field is the linear conditional expectation:

$$v_{\text{tgt}}(\mathbf{z}^{(\tau)}) = \frac{\tau\sigma^2 - (1-\tau)}{(1-\tau)^2 + \tau^2\sigma^2}\, \mathbf{z}^{(\tau)}, \tag{23}$$

with variance:

$$\mathrm{Var}(v_{\text{tgt}}(\mathbf{z}^{(\tau)})) = \frac{(\tau\sigma^2 - (1-\tau))^2}{(1-\tau)^2 + \tau^2\sigma^2} I_d. \tag{24}$$

Subtracting:

$$\mathbb{E}_{\mathbf{z}^{(\tau)}}[\mathrm{Var}(v_{\text{tgt}} \mid \mathbf{z}^{(\tau)})] = \mathrm{Var}(v_{\text{tgt}}) - \mathrm{Var}(\mathbb{E}[v_{\text{tgt}} \mid \mathbf{z}^{(\tau)}]) \tag{25}$$

$$= (\sigma^2 + 1) I_d - \frac{(\tau\sigma^2 - (1-\tau))^2}{(1-\tau)^2 + \tau^2\sigma^2} I_d \tag{26}$$

$$= \frac{\sigma^2}{(1-\tau)^2 + \tau^2\sigma^2} I_d. \tag{27}$$

Since $\mathbf{z}^{(0)} \sim \mathcal{N}(0, I_d)$, $\mathbf{z}^{(1)} \sim \mathcal{N}(0, \sigma^2 I_d)$, and $\mathbf{z}^{(\tau)} = (1-\tau)\mathbf{z}^{(0)} + s\mathbf{z}^{(1)}$, the triple $(\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \mathbf{z}^{(\tau)})$ is jointly Gaussian. As a result, conditional variances are homoscedastic. Therefore, for each flow time s:

$$\mathrm{Var}(v_{\text{tgt}} \mid \mathbf{z}^{(\tau)}) = \frac{\sigma^2}{(1-\tau)^2 + \tau^2\sigma^2} I_d. \tag{28}$$

Under SIGReg, $\sigma^2 \approx 1$, yielding the desired result. □

Proposition 2 (Uncertainty as Conditional Entropy Upper Bound)**.** *Let $\Sigma_{\mathbf{c}_t}$ denote the true predictive covariance for an action chunk embedding $\mathbf{c}_t$. For a well-trained $\mathcal{P}_\varphi$, $\mathcal{U}$ is an unbiased estimator of the mean conditional variance*

$$\mathbb{E}[\mathcal{U}] \;=\; \frac{\mathrm{tr}(\Sigma_{\mathbf{c}_t})}{d}. \tag{14}$$

By the maximum entropy theorem, the predictive entropy is upper-bounded by a function of $\mathcal{U}$ as

$$H(\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t) \;\le\; \frac{d}{2} \log(2\pi e \cdot \mathcal{U})\,. \tag{15}$$

Furthermore, $\mathcal{U}$ is bounded within $[0, 1]$, providing an interpretable absolute scale. $\mathcal{U} \approx 0$ indicates a confident, concentrated prediction, while $\mathcal{U} \approx 1$ indicates predictions as diffuse as the marginal prior. The bound in Eq. 15 is tight only for $\Sigma_{c_t} \propto I_d$ (App. A, Eq. 31); for anisotropic covariances it can be loose, so we use $\mathcal{U}$ as a practical exploration bonus rather than an entropy estimate.

Proof. **(i) Unbiasedness.** For a well-trained $\mathcal{P}_\varphi$, the ensemble $\{\hat{\mathbf{z}}^{(1)}[i]\}_{i=1}^N$ are i.i.d. draws from $p(\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t)$. The sample variance $\widehat{\mathrm{Var}}_i([\hat{\mathbf{z}}^{(1)}[i]]_j)$ is an unbiased estimator of the j-th marginal variance $[\Sigma_{\mathbf{c}_t}]_{jj}$ for each j. Averaging over dimensions:

$$\mathbb{E}[\mathcal{U}] = \frac{1}{d}\sum_{j=1}^{d}[\Sigma_{\mathbf{c}_t}]_{jj} = \frac{\mathrm{tr}(\Sigma_{\mathbf{c}_t})}{d}. \tag{29}$$

(ii) Entropy upper bound. By the maximum entropy theorem, given the same covariance matrix, the Gaussian distribution maximizes differential entropy:

$$H(\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t) \;\le\; \tfrac{1}{2}\log\big((2\pi e)^d\,|\Sigma_{\mathbf{c}_t}|\big). \tag{30}$$

This bound involves the determinant $|\Sigma_{\mathbf{c}_t}|$, which is not directly observable from the ensemble. We relate it to $\mathcal{U}$ via AM-GM applied to the eigenvalues $\{\lambda_i\}_{i=1}^d$ of $\Sigma_{\mathbf{c}_t}$:

$$\frac{1}{d}\log|\Sigma_{\mathbf{c}_t}| = \frac{1}{d}\sum_{j=1}^{d}\log\lambda_j \;\le\; \log\left(\frac{1}{d}\sum_{i=1}^{d}\lambda_i\right) = \log\frac{\mathrm{tr}(\Sigma_{\mathbf{c}_t})}{d} = \log\mathcal{U}, \tag{31}$$

where we applied Jensen's inequality to the concave $\log$, with equality if and only if $\Sigma_{\mathbf{c}_t} \propto I_d$. Therefore, $|\Sigma_{\mathbf{c}_t}| \le \mathcal{U}^d$, and we obtain:

$$H(\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t) \;\le\; \tfrac{1}{2}\log\big((2\pi e\,\mathcal{U})^d\big) = \tfrac{d}{2}\log(2\pi e\cdot\mathcal{U}). \tag{32}$$

(iii) Bounded scale. By the law of total variance:

$$\mathrm{Cov}(\mathbf{z}^{(1)}) = \mathbb{E}_{\mathbf{x}_t,\mathbf{c}_t}[\Sigma_{\mathbf{c}_t}] + \mathrm{Cov}(\mathbb{E}[\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t]). \tag{33}$$

Given the isotropic Gaussian latent space, $\mathbb{E}[\mathrm{tr}(\Sigma_{\mathbf{c}_t})] \le \mathrm{tr}(\mathrm{Cov}(\mathbf{z}^{(1)})) \approx d$. Therefore, $\mathbb{E}[\mathcal{U}] \lesssim 1$, with equality when $\mathrm{Cov}(\mathbb{E}[\mathbf{z}^{(1)} \mid \mathbf{x}_t, \mathbf{c}_t]) \approx 0$, i.e. when conditioning is uninformative. □

B Baselines and Implementation Details

B.1 Baselines

Goal-Conditioned Reinforcement Learning. We evaluate against standard goal-conditioned RL approaches: GCIVL [Park et al., 2025] (Goal-Conditioned Implicit Value Learning) and GCIQL [Kostrikov et al., 2021] (Goal-Conditioned Implicit Q-Learning), which learn value functions for goal-reaching tasks. **World Model based Planning.** We compare with recent world model approaches including PLDM [Sobal et al., 2022] (Planning with Latent Dynamics Models), a latent dynamics approach for control; DINO-WM Zhou et al. [2025], a deterministic latent world model on frozen DINOv2 features and LeWorldModel [Maes et al., 2026], a JEPA-style world model that learns predictive representations for planning in latent space.

B.2 Implementation Details

Token Sequence Construction for Causal Attention. The velocity field v_ϕ is implemented as a transformer operating over a length-$(H+K)$ token sequence constructed by concatenating history and noised future embeddings: $x_{1:(H+K)} = [z^{(1)}_{\text{hist}}, \ldots, z^{(H)}_{\text{hist}}, z^{(1)}_\tau, \ldots, z^{(K)}_\tau]$, augmented with learnable positional embeddings. We employ causal attention masking, ensuring each future token at position $H+i$, where $i \in \{1, 2, ..., K\}$ can only attend to the H history tokens and earlier future tokens, preventing information leakage from "unproduced" future states.

Conditioning Architecture. For history tokens, conditioning combines action embeddings with temporal information: $c_{ctx} = c_t + t_{tok}$, where c_t is the action embedding defined in the earlier section and t_{tok} is the temporal embedding from TimeMLP(τ). Future tokens receive only temporal conditioning: $c_{fut} = t_{tok}$. The full conditioning sequence is constructed as $c_{1:(H+K)} = [c_{\text{ctx}}, c_{\text{fut}}]$ and passed to the transformer alongside the token sequence.

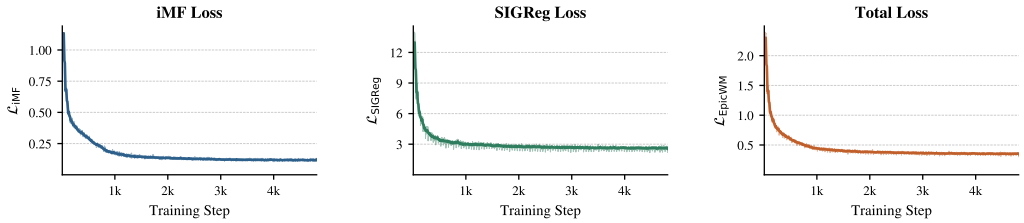


Figure 8: **Loss curves of EpicWorldModel on the visual PointMaze environment.** The bold colored lines represents the smoothed loss and the faint colored lines the actual per step loss. Overall we observe minimal variance and smooth convergence of the encoder and stochastic predictor.

Inference. For each action chunk, we sample 10 noise samples. For each noise sample, we require 2-NFE in case of average velocity or 10-NFE in case of FM. All models are trained with three seeds (0, 1, 2) and each is evaluated with three evaluation seeds; we report mean and variance. During evaluation we evaluate 50 test scenarios for all methods and seeds in parallel on a single A6000 GPU with 48GB of GPU-RAM showing the efficiency of latent world models. The typical goal offset distance from the start planning place is 25 steps.

β and NFE are selected by grid search on a held-out set of scenes disjoint from training and test scenes. N, K and NFE are fixed across all main experiments.

Training Details. All models were trained on A6000 GPUs. Especially with the iFM objective achieved early convergence after less than 5 GPU hours for Visual PointMaze data with simple action spaces and up to 20 GPU hours for Visual AntMaze with higher dimensional actions. All other environments land in-between. Training data across all OGBench Park et al. [2025] tasks and LabMaze Beattie et al. [2016] tasks was produced from heuristic experts trajectories. RoboCasa experiments were trained on 500 human demonstrations of NavigationKitchen.

Latent Space Architecture. EpicWorldModel operates in a learned latent space rather than directly on high-dimensional observations. Input observations $o_t \in \mathbb{R}^{3\times224\times224}$ are processed by $\mathcal{E}_\theta$, which has a Vision Transformer (ViT-Tiny) backbone with patch size 14 and embedding dimension $D_e = 192$, and trained from scratch without pretraining. The CLS token is projected through a two-layer MLP with BatchNorm to produce latent embeddings $z \in \mathbb{R}^D$, where $D = 192$. Goal states z_{goal} are encoded using the same encoder $\mathcal{E}_\theta$ to ensure goal distance computes within the shared latent space.

Action Encoding. Actions are embedded via a Conv1d-MLP architecture operating on frame-skipped action vectors $a \in \mathbb{R}^{F\cdot d_a}$ with frameskip $F = 5$. For example, PointMaze environments with $d_a = 2$ produce 10-dimensional action inputs, which are embedded to generate one D-dimensional action token per predicted future step.

Output Slicing. The transformer processes both history and future tokens but only returns predictions for the future portion: the output is sliced to out$[:, K:]$, corresponding to the predicted velocity field for K future time steps.

Planner. LeWM Maes et al. [2026], iMF and FM share a CEM planner over 25-step action sequences (5 chunks $\times$ 5 actions). Each planning call runs 30 iterations with 300 candidates, and a selection of top-30 sequences, initialised with std equal to the action scale. The final best mean is executed open-loop before replanning. We do 2 planning calls per 50-step episode and 3 planning calls for the long-horizon experiments. LeWM scores each candidate with one deterministic rollout using the final-step latent MSE to the latent goal z_g. EpicWorldModel draws N rollouts (FM: 10 NFE; iMF: 2 NFE) and scores mean goal $MSE - \beta U$, both z-score normalised across candidates.

Implementation and training details for RoboCasa task. All models (LeWM and EpicWorld-Model) are trained on 500 human demonstrations of NavigateKitchen (72,786 frames; held-out "target" scene split), with a frozen DINOv3-ViT-S/16 encoder (mean-pooled patch tokens, 384-d; zero encoder updates), followed by a trained projector and latent-dynamics predictor,using the JEPA-style latent-prediction recipe (predicting future projected latents conditioned on actions; SIGReg regularization; no pixel reconstruction). Inputs consist of a 3-frame history with a frameskip of 5 (single egocentric camera, 224×224); actions are 3-DoF base velocity commands (the arm is masked for this navigation task). Optimization uses AdamW ($\text{lr} = 5 \times 10^{-5}$, $\text{wd} = 10^{-3}$), with a batch size of

448, bf16 precision, and 100 epochs on a single GPU ($\sim$2.5–4 h per model). The three variants—deterministic AR, one-step iMeanFlow (iMF), and 10-step flow matching (FM)—differ only in the predictor family; the encoder, projector architecture, data, and all hyperparameters are identical. All evaluations use each model's final (epoch-100) checkpoint; training losses converged well before epoch 100. Figure 8 shows the validation loss across individual loss components in Equations (6) and (11) indicating smooth convergence as a result of Proposition 1 of the main manuscript.

C Additional Results

We also report additional experiments that complement the main results including:

- An ablation disabling the exploration term $\mathcal{U}$
- a compute-matched comparison (in contrast to a step-matched) against the deterministic baseline
- a sweep over sampled futures
- and a population-level check of the latent-space analysis in Sec. 4.4.

Unless noted otherwise, all experiments follow the CEM/MPC planning protocol of App. B as used in the main paper.

C.1 Effect of the Exploration Term ($\beta = 0$)

To isolate the contribution of the exploration term from the contribution of the predictor itself, we extend β/NFE sweep of Fig. 6(a) with a $\beta = 0$ condition on Visual PointMaze Giant. Disabling the exploration term lowers success by 7 points for iMF (NFE = 4, $\beta = 0.4$) and by 6 points for FM at NFE = 10 ($\beta = 0.4$). The stochastic predictor alone therefore explains part, but not all of the gain over the deterministic LeWM Maes et al. [2026]. This can only be attributed to the exploration term in the planning cost.

Table 3: Effect of disabling the exploration term ($\beta = 0$) on Visual PointMaze Giant, as a function of the number of function evaluations (NFE) at planning time. Left: iMeanFlow (iMF); right: flow matching (FM). Success rate (%), single training/evaluation run per cell. Best value per column in **bold**.

	iMF					**FM**	
$\beta \backslash$ NFE	1	2	4	8	10	4	10
0.0	51.0	69.5	76.5	81.5	81.5	78.0	79.5
0.4	**51.5**	**74.5**	**83.5**	**86.0**	**86.0**	**81.0**	**85.5**

C.2 Compute-Matched LeWorldModel

Our method evaluates N stochastic rollouts per candidate at planning time, while LeWM, as a deterministic model, does only compute a single rollout per action candidate. Therefore a fraction of EpicWorldModel could be in principle come from the additional inference compute rather than the exploration cost term and stochastic predictor. We test this by giving the deterministic LeWM baseline additional CEM action candidates to approximate the per-candidate cost of our $N \times \text{NFE} = 20$ (NFE=4, N=5) evaluations up to $20 \times 300 = 6000$ candidates.

Table 4 shows that scaling LeWM's action candidate budget alone recovers only part of the gap. On PointMaze the success rises from 54.0% to 74.0% but plateaus with any additional compute. The remaining gap to EpicWorldModel iMF with $\approx$10 points remains at lower and matched compute. Similarly to the main findings AntMaze is not bounded by planning compute or methodology and rather is bounded by task and action-space complexity. We refer to the discussions in Sec. 4.3 for the discussion on the AntMaze success rate.

C.3 Number of Sampled Futures

We further ablate how the number of sampled future states N (distinct from the planning horizon K) affects planning performance. In Tab. 5 we sweep $N \in \{2, 5, 10\}$ and at $\beta \in \{0, 0.4\}$ on the Visual

Table 4: Compute-matched comparison: LeWM success rate (%) against the number of CEM action candidates (default: 300), on Visual PointMaze Giant and Visual AntMaze Giant. Bottom rows: EpicWorldModel (iMF, $\beta = 0.4$) at NFE = 2 and NFE = 4 for reference (from Tab. 3). Best value per column in **bold**.

CEM candidates	PointMaze	AntMaze
300 (default)†	64.0	26.0
600	68.0	**26.5**
900	72.0	24.0
1500	74.0	**26.5**
6000	74.0	25.0
EpicWM (iMF, β=0.4, NFE=2)	74.5	—
EpicWM (iMF, β=0.4, NFE=4)	**83.5**	—

PointMaze task for FM (NFE=10) and iMF (NFE=2) stochastic predictors. Across all four settings the effect of N is small, showing the robustness of the stochastic predictor. Similar to the optimal selection of β we see an automatic N selection beyond a single default across tasks as future work.

Table 5: Effect of the number of sampled futures N on Visual PointMaze Giant success rate (%), for both predictor families (FM and iMF) at $\beta \in \{0, 0.4\}$ (NFE fixed at each method's main-text default; App. B). * marks N=5, the default value used in the main results. "Range" is $\max - \min$ over the row. Best value per row in **bold**.

Method	β	$N = 2$	$N = 5^*$	$N = 10$	Range
iMF	0.0	**78.0**	76.5	77.5	1.5
iMF	0.4	80.0	**83.5**	78.0	5.5
FM	0.0	79.0	**82.0**	81.0	3.0
FM	0.4	**81.5**	81.0	81.0	0.5

C.4 Latent-Space Analysis over All Evaluation Scenarios

Section 4.6 shows an qualitative example of the exploration signal on a single rollout. We show that the top-k predictions in there latent space contract as the agent approaches the goal (Fig. 7; 1σ ellipse area shrinking from 60.99 to 11.86). In addition to that, we provide a quantitative analysis over 50 evaluation scenarios. Across all episodes the distance to the goal latent $\|z_t - z_{\text{goal}}\|$ decreases from 17.4 ± 5.1 at t=0 to 10.4 ± 5.8 at t=25, corresponding to a median progress of 45.5% (58.6% on the 80% of episodes that succeed). Among all 1.500 sampled future predictions across these rollouts, the closest sample to the ground-truth future latent lies at a median distance of 3.5. This is roughly five times smaller than the initial distance to the goal and further delivers a median cosine alignment of 0.88 between the predicted and ground-truth future directions. These results are consistent with the sampled futures tracking the ground-truth trajectory rather than producing arbitrary or uninformative spread, consistent with the exploration signal being useful for goal approach, as illustrated qualitatively in Fig. 7.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Introduction and abstract clearly explain the contributions of our work. They also provide the relevant context and introduce the justification for a stochastic latent world model.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Experimental results are discussed and show a quantitative difference in specific tasks that is not reaching the performance of RL methods yet. Additionally the paper clearly states it focus on POMDP settings but shows limited improvements in other settings. This is discussed as part of the result discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer:

Justification: All assumptions and most proofs are provided in the main paper. Additional details of each proof are delivered in the supplementary.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper builds all its experimental results on top of existing benchmarks and baselines. The implementation of the core method is fully described in the main paper and implementation details are given.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer:[No]

   Justification: At the time of the submission the code was only described and printed in the supplementary, but not submitted. The authors promise to release the code at a later stage.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Implementation details and hyperparameter are provided, either ass part of the implementation section or as part of the ablations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are provided for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about training and inference GPU cost.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not violate any data license by building fully on open simualtors and datasets. No human subject was involved in the experiments.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper proposes a new concept for latent world models under uncertainty and feasibility of such models in toy experiments. Real-world impact is not the goal of this paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and code used is properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM use is decleared on OpenReview. This paper did not use LLm for anything but the state use to support the integration of baselines, integration of datasets/simulation environments.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.