Policy-Guided World Model Planning for Language-Conditioned Visual Navigation

Anonymous CVPR submission

Paper ID *****

Abstract

Navigating to a visually specified goal given natural language instructions remains a fundamental challenge in embodied AI. Existing approaches either rely on reactive policies that struggle with long-horizon planning, or employ world models that suffer from poor action initialization in high-dimensional spaces. We present PiJEPA, a two-stage framework that combines the strengths of learned navigation policies with latent world model planning for instruction-conditioned visual navigation. In the first stage, we finetune an Octo-based generalist policy, augmented with a frozen pretrained vision encoder (DINOv2 or V-JEPA-2), on the CAST navigation dataset to produce an informed action distribution conditioned on the current observation and language instruction. In the second stage, we use this policy-derived distribution to warm-start Model Predictive Path Integral (MPPI) planning over a separately trained JEPA world model, which predicts future latent states in the embedding space of the same frozen encoder. By initializing the MPPI sampling distribution from the policy prior rather than from an uninformed Gaussian, our planner converges faster to high-quality action sequences that reach the goal. We systematically study the effect of the vision encoder backbone, comparing DINOv2 and V-JEPA-2, across both the policy and world model components. Experiments on real-world navigation tasks demonstrate that PiJEPA significantly outperforms both standalone policy execution and uninformed world model planning, achieving improved goal-reaching accuracy and instruction-following fidelity.

1. Introduction

Building autonomous agents that can navigate to a goal specified by an image and a natural language instruction is a long-standing challenge in robotics and computer vision [1–3]. Given a current egocentric observation o_t, a goal image o_g, and an instruction ℓ (*e.g.* "move towards the stairs"), the agent must produce a sequence of actions that reliably reaches the goal while respecting the semantics of the instruction.

Recent vision-language-action (VLA) models [3–6] have made remarkable strides toward this objective. Generalist policies such as Octo [4] and NoMaD [7] can be finetuned on domain-specific navigation data to achieve impressive short-horizon control, while the CAST framework [6] further strengthens instruction following through counterfactual language-action augmentation. Despite their effectiveness, these reactive policies predict actions in a single forward pass and thus lack the capacity to reason about the long-term consequences of their decisions.

A complementary line of work addresses this limitation by learning world models that predict future states conditioned on candidate actions, enabling explicit planning toward the goal [8–12]. Among these, Joint-Embedding Predictive Architecture World Models (JEPA-WMs) [12–14] are especially attractive: by learning dynamics in the latent space of a frozen pretrained encoder, they support efficient rollouts without pixel-level reconstruction. At inference time, a sampling-based optimizer such as MPPI [15, 16] or CEM [17] searches over action sequences by unrolling the learned predictor and minimizing the embedding-space distance to the encoded goal. However, this search must typically begin from an uninformed prior over a high-dimensional action space, resulting in slow convergence and susceptibility to local minima [13].

We propose PiJEPA, a framework that bridges these two paradigms by using a learned policy to *warm-start* world model planning. Our key insight is that a finetuned VLA policy, while insufficient for long-horizon reasoning on its own, provides a highly informative *action prior*: it concentrates probability mass in the region of action space most likely to make progress toward the goal. Initializing the MPPI sampling distribution from this prior allows the world model planner to devote its search budget to refining already-promising trajectories rather than exploring the full action space from scratch.

Concretely, our pipeline operates as follows (see Fig-

ure 1):

1. **Policy prior.** Given the current observation o_t and instruction ℓ, a finetuned Octo model produces N_π action chunk samples via its diffusion head. We transform these into the world model's local-frame action space and compute their empirical mean μ_π and standard deviation σ_π.
2. **MPPI planning.** A JEPA world model, trained with the same frozen encoder, predicts future latent states. We run MPPI initialized at (μ_π, σ_π) and optimize against the embedding-space distance to the encoded goal o_g.
3. **Execution.** The first action of the optimized sequence is executed, and the process repeats at the next replanning step.

Our contributions are:

1. We propose PiJEPA, a unified framework for language-conditioned visual navigation that combines a finetuned Octo policy with MPPI-based planning over a JEPA world model, using the policy output to warm-start the planner.
2. We train both components with modern pretrained vision encoders (DINOv2 and V-JEPA-2), providing a systematic comparison of image-based vs. video-based representations for policy learning and world modeling in navigation.
3. We demonstrate on real-world navigation tasks that policy-guided planning significantly outperforms both standalone policy execution and uninformed MPPI planning, improving goal-reaching accuracy and instruction-following fidelity.

2. Related Work

Our work draws on two active research threads—language-conditioned policies and latent world models—and unifies them through policy-guided planning. We review each in turn.

2.1. Language-Conditioned Navigation Policies

The integration of natural language with embodied perception has given rise to Vision-Language-Action (VLA) models that map instructions and visual observations directly to motor commands. Generalist architectures such as Octo [4] learn broadly from diverse robot data, while recent scalable designs including Dita [18], LAPA [19], and CLAP [20] push the frontier of reactive instruction-following through latent action representations. In the navigation domain specifically, UrbanNav [21] extends language-guided policies to complex urban environments via web-scale trajectory learning. Although these systems achieve strong short-horizon performance, they share a common limitation: actions are selected greedily without explicit reasoning about future states, which degrades performance on tasks that require multi-step spatial planning. PiJEPA retains the rapid, instruction-grounded action proposals of a reactive VLA policy but uses them as a warm start for a separate planning stage, thereby combining language understanding with long-horizon deliberation.

2.2. Latent World Models for Planning

World models offer a principled route to long-horizon reasoning by enabling agents to simulate the consequences of candidate actions before committing to them. Early approaches such as Navigation World Models (NWM) [11] synthesize pixel-level futures, but predicting dynamics in a learned latent space drastically reduces computational cost. Joint-Embedding Predictive Architecture (JEPA) world models built on frozen foundation encoders like DINOv2 have proven particularly effective for physical planning tasks [12, 13]. This family of models has been further advanced through continuous latent action formulations [22], principled regularization strategies such as LeJEPA [23], and joint VLA-latent pretraining [24]. Complementary efforts have also targeted the efficiency of the planning loop itself; for instance, One-Step World Model [25] reduces the computational latency of multi-step rollouts, and benchmarks like Target-Bench [26] evaluate world-model-based path planning toward text-specified targets. Despite these advances, sampling-based planners such as MPPI and CEM remain sensitive to their initialization: starting from an uninformed Gaussian prior over high-dimensional action spaces leads to slow convergence and frequent entrapment in local minima [13]. PiJEPA directly addresses this inference-time bottleneck. Rather than conditioning the world model itself on language—which would conflate representation learning with instruction grounding—we keep the latent dynamics model language-agnostic and instead anchor the MPPI optimization with an instruction-aware action prior drawn from the VLA policy. This decoupled design lets each component focus on what it does best: the policy provides efficient, semantically informed proposals, while the world model evaluates and refines them over extended horizons.

3. Method

We present PiJEPA, a two-stage approach for instruction-conditioned visual goal navigation. An overview is shown in Figure 1. Our pipeline consists of three learned components: a frozen pretrained vision encoder, an Octo-based policy that generates an informed action prior, and a JEPA world model over which we perform MPPI planning warm-started by the policy prior.

3.1. Problem Formulation

At each timestep t, the agent observes an egocentric RGB image o_t, is given a goal image o_g and a natural language

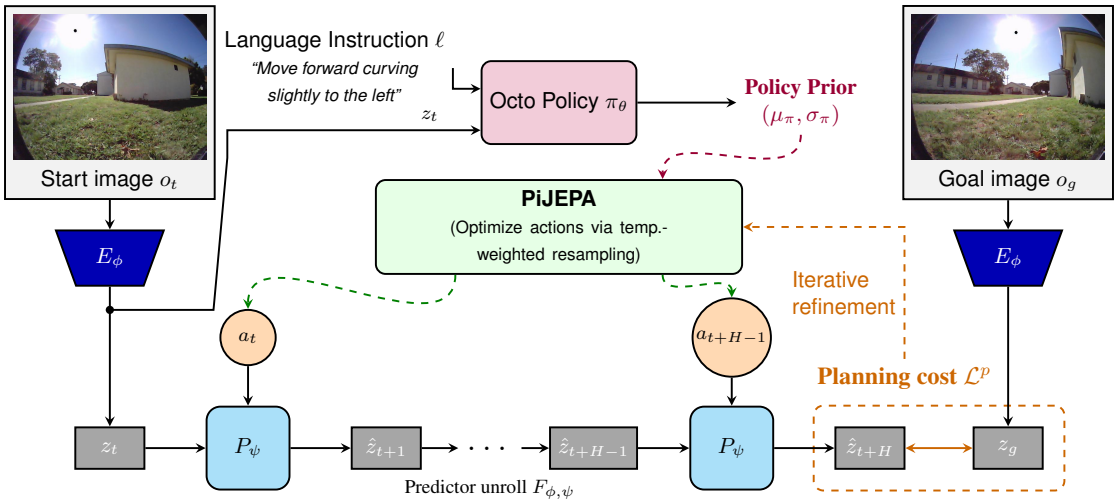


Figure 1. **Overview of PiJEPA. Top:** The Octo policy, finetuned with a frozen vision encoder (E_ϕ), takes the current latent observation z_t and instruction ℓ as input and produces action chunk samples via its diffusion head. These are transformed from the global frame to the world model's local body frame. **Middle:** The policy's statistics (μ_π, σ_π) warm-start MPPI, which iteratively optimizes the action distribution over J iterations. **Bottom:** The JEPA world model predictor (P_ψ), trained with the same frozen encoder, autoregressively predicts future latent states. The MPPI candidates are scored by unrolling the world model and evaluating the latent-space distance to the encoded goal z_g (Algorithm 1).

instruction ℓ, and must produce a sequence of navigation actions $a_{1:H} \in \mathbb{R}^{H\times A}$ over a planning horizon H that guides the robot towards the goal while respecting the instruction semantics.

3.2. Policy Prior from Octo

The first stage of PiJEPA extracts an action prior from a finetuned Octo policy [4]. Octo is a transformer-based generalist policy with a diffusion action head that models the distribution $\pi_\theta(a|o_t, \ell)$ over action chunks.

Both the policy and the world model operate on top of a shared frozen pretrained vision encoder E_ϕ that maps observations to latent tokens $z_t = E_\phi(o_t) \in \mathbb{R}^{n\times d}$, where n is the number of spatial tokens and d the embedding dimension. The encoder weights ϕ remain frozen throughout, ensuring both components share a consistent representation space.

3.3. JEPA World Model

The second component is a Joint-Embedding Predictive World Model (JEPA-WM) [12, 13] that predicts future latent states given the current state and actions. The world model pairs the same frozen encoder E_ϕ with a learnable predictor P_ψ and action encoder A_ψ. Given a context of encoded observations and actions, the predictor forecasts the next latent state:

$$\hat{z}_{t+1} = P_\psi(z_{t-w:t},\ A_\psi(a_{t-w:t}))\,. \tag{1}$$

The predictor uses a causal attention mask so it can predict from all context lengths up to a maximum window w. Actions are injected into the predictor via a conditioning mechanism at every layer to prevent vanishing action signals through depth [13].

The predictor is trained to minimize the MSE between predicted and target embeddings. Following [13], we use a multi-step rollout loss to improve long-horizon accuracy:

$$\mathcal{L}_{\text{total}} = \sum_{k=1}^{K_{\text{roll}}} \frac{1}{B} \sum_{b=1}^{B} \left\| F_{\phi,\psi}(z_t^b, a_{t:t+k-1}^b) - z_{t+k}^b \right\|_2^2, \tag{2}$$

where $F_{\phi,\psi}$ denotes the autoregressive unrolling of the predictor (feeding each predicted $\hat{z}_{t+j}$ as context for the next step), trained with truncated backpropagation through time.

3.4. Policy-Guided MPPI Planning

At test time, we combine the Octo policy prior with the JEPA world model through MPPI-based planning [15, 16]. We first draw N_π action chunk samples from the policy diffusion head. Because the policy and world model may operate in different action coordinate frames, each sample is transformed by a mapping T into the world model frame (details in Section 4.1). From the transformed samples $\tilde{a}^{(i)} = T(a^{(i)})$, we compute the initial MPPI mean and standard deviation:

$$\mu^0 = \frac{1}{N_\pi} \sum_{i=1}^{N_\pi} \tilde{a}^{(i)}, \quad \sigma^0 = \text{clamp}\Big(\text{std}(\{\tilde{a}^{(i)}\}),\ \sigma_{\min},\ \sigma_{\max}\Big)\,, \tag{3}$$

which defines a policy-informed initialization over action sequences.

Planning objective. Given the encoded current observation z_t and goal encoding z_g, we evaluate a candidate action sequence $a_{1:H}$ by the terminal latent distance

$$\mathcal{L}^p(z_t, a_{1:H}, z_g) = \frac{1}{n}\sum_{i=1}^{n}\left\|\hat{z}_{t+H}^{(i)} - z_g^{(i)}\right\|_2^2, \tag{4}$$

where $\hat{z}_{t+H} = F_{\phi,\psi}(z_t, a_{1:H})$ is obtained by autoregressively rolling out the world model for H steps.

MPPI refinement. Starting from (μ^0, σ^0), MPPI iteratively samples candidate action sequences, evaluates them using Eq. 4, and updates the sampling distribution using temperature-weighted elite trajectories. At iteration j, elite weights are computed as

$$w_k = \frac{\exp\big(\lambda\,(c_{\min} - c_k)\big)}{\sum_{k'=1}^{K}\exp\big(\lambda\,(c_{\min} - c_{k'})\big)}, \tag{5}$$

and the Gaussian parameters are updated by

$$\mu^{j+1} = \sum_k w_k\, a^{(k)}, \qquad (\sigma^{j+1})^2 = \sum_k w_k\big(a^{(k)} - \mu^{j+1}\big)^2, \tag{6}$$

with σ^{j+1} clamped to $[\sigma_{\min}, \sigma_{\max}]$. After J iterations, one elite trajectory is sampled proportionally to w_k and executed. The full procedure is summarized in Algorithm 1.

Conditioning. The language instruction ℓ enters only through the Octo policy, which shapes the initial planning distribution. The world model is language-agnostic and models only visual-action dynamics. The goal image o_g is encoded into z_g and enters through the planning cost in Eq. 4.

Algorithm 1 PiJEPA: Policy-Guided World Model MPPI Planning

Require: Obs. o_t, goal o_g, instruction ℓ, policy π_θ, world model P_ψ, encoder E_ϕ
Require: Horizon H, samples N, elites K, iterations J, temperature λ, policy samples N_π

Stage 1: Policy prior
$z_t \leftarrow E_\phi(o_t), \quad z_g \leftarrow E_\phi(o_g)$
for $i = 1, \dots, N_\pi$ **do**
  $a^{(i)} \sim \pi_\theta(\cdot \mid o_t, \ell)$ ▷ diffusion sampling
  $\tilde{a}^{(i)} \leftarrow T(a^{(i)})$ ▷ coordinate transform
end for
$\mu^0 \leftarrow \text{mean}(\{\tilde{a}^{(i)}\})$
$\sigma^0 \leftarrow \text{clamp}(\text{std}(\{\tilde{a}^{(i)}\}), \sigma_{\min}, \sigma_{\max})$

Stage 2: MPPI planning
for $j = 0, \dots, J-1$ **do**
  for $i = 1, \dots, N$ **do**
    $a_{1:H}^{(i)} \leftarrow \text{clamp}\big(\mu^j + \sigma^j \odot \epsilon^{(i)},\ -1,\ 1\big)$
    $\epsilon^{(i)} \sim \mathcal{N}(0, I)$
    $\hat{z} \leftarrow z_t$
    for $h = 1, \dots, H$ **do**
      $\hat{z} \leftarrow P_\psi(\hat{z},\ a_h^{(i)})$ ▷ autoregressive rollout
    end for
    $c^{(i)} \leftarrow \|\hat{z} - z_g\|_2^2 / n$ ▷ Eq. 4
  end for
  $\mathcal{E} \leftarrow$ top-K by lowest $c^{(i)}$
  $w_k \leftarrow \exp(\lambda\,(c_{\min} - c_k))\ /\ \sum_{k'}\exp(\lambda\,(c_{\min} - c_{k'}))$
  $\mu^{j+1} \leftarrow \sum_k w_k\, a^{(k)}$
  $\sigma^{j+1} \leftarrow \text{clamp}\Big(\sqrt{\sum_k w_k (a^{(k)} - \mu^{j+1})^2},\ \sigma_{\min},\ \sigma_{\max}\Big)$
end for
Sample $k^* \in \mathcal{E}$ with probability w_{k^*}
return $a_{1:H}^{(k^*)}$

4. Experiments

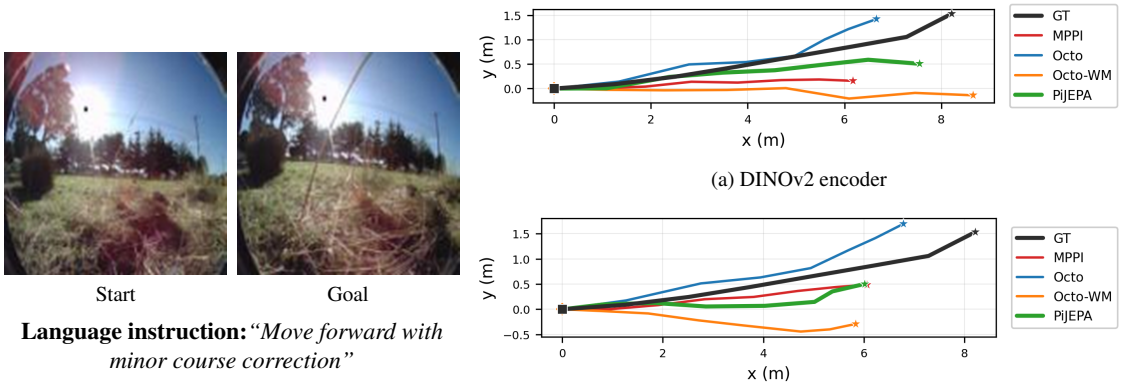
4.1. Experimental Setup

Dataset. We train and evaluate on the CAST dataset [6], a large-scale visual navigation dataset augmented with counterfactual instruction-action pairs. CAST addresses the problem of posterior collapse—where the policy ignores the language instruction because the observation alone suffices to predict the action—by using a VLM to generate alternative instructions and an atomic policy to produce corresponding counterfactual action labels. Each action is a 4-dimensional vector $a = (\Delta x, \Delta y, \sin\Delta\phi, \cos\Delta\phi)$ specifying the robot's displacement and heading change; the $(\sin\Delta\phi, \cos\Delta\phi)$ encoding avoids angle discontinuities at $\pm\pi$. Actions are normalized to $[-1, 1]$ using bounds normalization based on the 1st and 99th percentile statistics.

Vision encoders. We study two frozen pretrained encoder families: *DINOv2 ViT-S* [27], a self-supervised image encoder with strong spatial and object segmentation features; and *V-JEPA-2 ViT-L* [28], a self-supervised video encoder that captures temporal dynamics. For V-JEPA-2, we follow the frame-duplication strategy of [13], encoding each frame as a duplicated 2-frame video. We apply layer normalization to the encoder output to stabilize prediction targets and the planning cost landscape.

Policy training. We finetune the Octo-Small model [4] on CAST, replacing its original visual encoder with the frozen pretrained encoder E_ϕ followed by a learnable projection $W_{\text{proj}} \in \mathbb{R}^{d \times d_{\text{octo}}}$. Language instructions are encoded with a pretrained T5 model [29] (16 tokens). The diffusion action head is a 3-layer MLP with residual connections, trained with the DDPM objective [30] using a cosine noise schedule. Training uses the CAST-augmented data including both original and counterfactual trajectory segments.

Coordinate transform. The Octo policy outputs global-frame displacements, while the world model expects local body-frame actions. We bridge this gap by accumulating



(a) DINOv2 encoder

(b) V-JEPA-2 encoder

Figure 2. **Qualitative trajectory comparison.** Given a start observation, goal image, and language instruction (left), we compare trajectories produced by each method under two encoder backbones (right). The black curve (GT) shows the ground-truth path; colored curves show MPPI (red), Octo policy (blue), Octo-WM scoring (orange), and PiJEPA (green). Stars mark final positions. PiJEPA most closely tracks the ground truth in both settings.

headings from the $(\sin \Delta\phi, \cos \Delta\phi)$ components and rotating the displacement into the robot's local frame at each timestep:

$$\phi_t = \sum_{\tau=0}^{t-1} \operatorname{atan2}(\sin \Delta\phi_\tau, \cos \Delta\phi_\tau),$$
$$\begin{pmatrix} \Delta x_t^\ell \\ \Delta y_t^\ell \end{pmatrix} = \begin{pmatrix} \cos \phi_t & \sin \phi_t \\ -\sin \phi_t & \cos \phi_t \end{pmatrix} \begin{pmatrix} \Delta x_t \\ \Delta y_t \end{pmatrix}. \quad (7)$$

The rotated actions are then normalized to $[-1, 1]$ using the dataset bounds statistics.

World model training. The JEPA world model predictor is a ViT with frame-causal attention and Adaptive Layer Normalization (AdaLN) action conditioning [13, 31], which modulates scale and shift at every transformer block. It is trained on CAST trajectory segments with the multi-step rollout loss (Eq. 2, $K_{\text{roll}} = 2$) using truncated backpropagation through time. The encoder weights remain frozen; only the predictor and action encoder parameters are updated.

Encoder variants. We train four model configurations to systematically study encoder choice: (i) DINOv2-Policy + DINOv2-WM, the most natural consistent-space pairing; (ii) V-JEPA-2-Policy + V-JEPA-2-WM, leveraging temporal features in both components; (iii) DINOv2-Policy + V-JEPA-2-WM; and (iv) V-JEPA-2-Policy + DINOv2-WM. The cross-encoder configurations test whether the action prior from one representation space transfers to planning in another. Prior work [13] found DINOv2 excels at fine-grained spatial reasoning while V-JEPA-2 captures richer temporal structure; evaluating all four combinations reveals which properties matter for the policy prior versus latent dynamics.

MPPI hyperparameters. We use $J = 2$ MPPI iterations, $N = 32$ candidate samples, $K = 4$ elites, inverse temperature $\lambda = 0.8$, and $N_\pi = 4$ Octo diffusion samples for the policy prior. The standard deviation is clamped to $[\sigma_{\min}, \sigma_{\max}] = [0.01, 0.05]$.

4.2. Baselines and Evaluation

To isolate the contribution of each component, we compare four planning strategies:

- **Default MPPI**: Standard MPPI initialized with $\mu^0 = \mathbf{0}$, $\sigma^0 = \sigma_{\max} \mathbf{1}$, serving as a pure world model planning baseline.
- **PiJEPA** (ours): Warm-started MPPI with $\mu^0 = \mu_\pi$, $\sigma^0 = \sigma_\pi$ from the policy prior (Algorithm 1).
- **Octo-WM**: Draws N_π samples from the policy, evaluates each via a full world model rollout, and selects the sample with the lowest cost $\mathcal{L}^p$—using the world model for scoring without trajectory optimization.
- **Octo**: Octo predicted actions with no world model involvement, serving as a pure reactive policy baseline.

We report Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) for both XY position (in meters) and heading (in degrees), computed between predicted and ground-truth trajectories on the CAST validation set.

4.3. Results

Tables 1 and 2 report the full set of trajectory metrics for the DINOv2 and V-JEPA-2 encoder configurations, respectively. We discuss the main findings below.

PiJEPA achieves the best positional accuracy across both encoders. PiJEPA leads on ATE XY metrics in both configurations: 1.78 m RMSE and 3.12 m Final with DINOv2, and 1.65 m RMSE and 1.32 m Mean with V-JEPA-2—the lowest across all settings. Octo-WM narrowly edges PiJEPA on V-JEPA-2 Final position (2.87 m vs. 2.88 m), but

Table 1. **DINOv2 ViT-S** on CAST validation episodes. Best in **bold**. All ↓.

	ATE XY (m)			ATE Hdg (°)			RPE XY (m)		RPE Hdg (°)	
	RMSE	Mean	Final	RMSE	Mean	Final	RMSE	Mean	RMSE	Mean
Octo	1.98	1.65	3.19	20.08	16.76	28.55	0.61	0.57	8.29	6.94
MPPI	1.85	1.48	3.23	**18.79**	**15.68**	**27.86**	0.55	0.51	**5.70**	**5.11**
Octo-WM	1.80	1.43	3.15	22.16	18.60	29.16	0.56	0.51	7.51	6.45
PiJEPA	**1.78**	**1.42**	**3.12**	19.89	16.52	28.63	**0.56**	**0.51**	7.05	6.07

Table 2. **V-JEPA-2 ViT-L** on CAST validation episodes. Best in **bold**. All ↓.

	ATE XY (m)			ATE Hdg (°)			RPE XY (m)		RPE Hdg (°)	
	RMSE	Mean	Final	RMSE	Mean	Final	RMSE	Mean	RMSE	Mean
Octo	1.72	1.36	3.02	18.72	15.64	27.72	0.53	0.48	6.086	5.34
MPPI	1.87	1.50	3.29	19.15	15.98	28.50	0.55	0.52	**5.75**	**5.15**
Octo-WM	1.67	1.35	**2.87**	20.63	17.27	28.06	0.54	0.49	6.93	5.99
PiJEPA	**1.65**	**1.32**	2.88	**19.13**	**15.95**	**27.93**	**0.53**	**0.49**	6.46	5.62

PiJEPA dominates on the remaining trajectory-level metrics, confirming that warm-starting MPPI with a policy prior enables the planner to refine already-promising trajectories rather than searching from scratch.

Uninformed MPPI struggles with position but excels at heading. MPPI exhibits a striking divergence across metric types: it is the weakest method on ATE XY under V-JEPA-2 (RMSE 1.87 m, Final 3.29 m), confirming that uninformed initialization wastes the sampling budget, yet it achieves the best heading metrics across both encoders (e.g., DINOv2: ATE Hdg RMSE 18.79°, RPE Hdg Mean 5.11°). We attribute this to the embedding-space objective implicitly regularizing heading alignment even when translational accuracy is poor.

World model scoring captures much of the planning benefit. Octo-WM provides large positional gains over the raw Octo policy—with V-JEPA-2, ATE XY Final drops from 3.02 m to 2.87 m—suggesting that even without iterative optimization, using the world model to filter poor policy proposals is a powerful strategy. The relatively small gap between Octo-WM and PiJEPA indicates that filtering accounts for much of the benefit, though the additional MPPI refinement still yields the best overall trajectory accuracy.

The reactive policy provides strong local control. Despite lacking look-ahead, Octo remains competitive on step-level metrics: with V-JEPA-2 it achieves the best RPE XY Mean (0.48 m) and strongest ATE Heading (RMSE 18.72°, Final 27.72°). Planning thus contributes its main benefit at the trajectory level, where cumulative errors compound.

V-JEPA-2 yields stronger overall performance. V-JEPA-2 with PiJEPA achieves the best overall positional accuracy, and even its Octo baseline (ATE XY RMSE 1.72 m) surpasses the best DINOv2 method (1.78 m). However, uninformed MPPI performs worst under V-JEPA-2 (Final 3.29 m vs. 3.23 m for DINOv2), suggesting its richer latent dynamics create a harder optimization landscape for uninformed search. Once anchored by the policy prior, this space becomes an asset: the gap between PiJEPA and uninformed MPPI on ATE XY Final grows from 0.11 m (DINOv2) to 0.41 m (V-JEPA-2).

Planning latency. The Octo policy requires approximately 2.13 s to generate its action proposals via diffusion sampling, while the MPPI planning stage adds only 0.35 s, bringing the total PiJEPA inference time to roughly 2.48 s for an 8-step trajectory. The lightweight MPPI overhead demonstrates that the planning stage is practical to layer on top of the policy, with the vast majority of latency attributable to the diffusion-based action sampling rather than the world model rollouts.

Qualitative results and failure analysis. Figure 2 shows that PiJEPA (green) most closely tracks the ground-truth path under both encoders, while MPPI (red) deviates substantially and Octo (blue) drifts over longer horizons. Figure 3 presents a failure case with the ambiguous instruction *"Follow the building,"* where the Octo policy veers toward the wrong building. The world model itself gets stuck in its rollouts, and the predicted latent states remain nearly identical across the planning horizon regardless of the sampled actions, leaving the planner unable to make progress. PiJEPA partially mitigates this through the policy

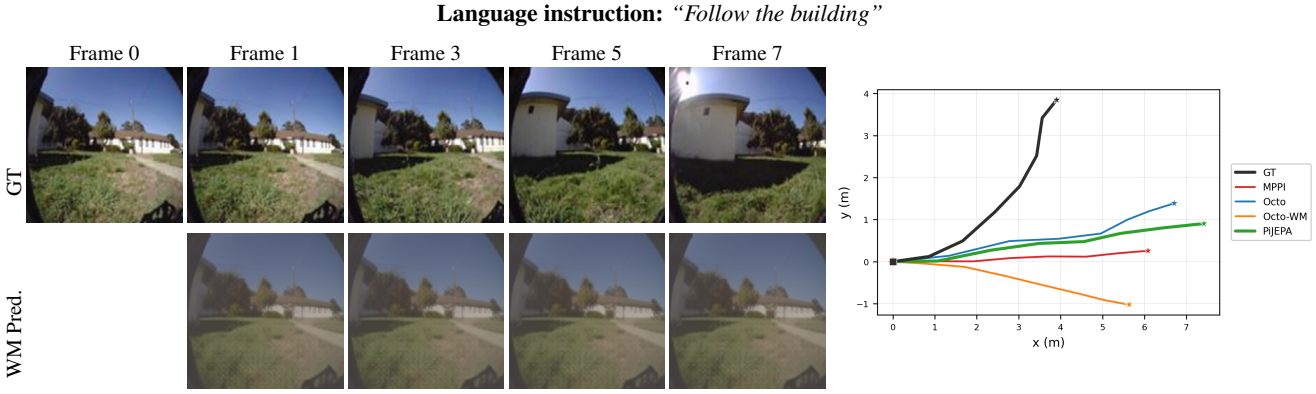


Figure 3. **Failure case analysis.** The language instruction *"Follow the building"* is inherently ambiguous, as multiple buildings are visible in the scene. The Octo policy (blue) misinterprets the referent and veers toward a different building, illustrating how vague instructions can mislead reactive policies that lack long-horizon reasoning. Meanwhile, the world model fails to make meaningful progress because it becomes stuck in its rollouts, predicting nearly identical latent states, which causes the planner to stagnate. The WM Pred. row confirms this directly. The predicted observations remain largely unchanged across the planning horizon. PiJEPA (green) partially mitigates both issues by grounding the planner with a policy-derived prior, though it still undershoots the ground-truth trajectory.

prior, which bypasses the stalled rollouts by anchoring the search in a productive region of action space, though it still undershoots the ground truth, indicating that ambiguous instructions and world model stagnation remain challenging.

5. Conclusion

We have presented PiJEPA, a framework for instruction-conditioned visual navigation that warm-starts MPPI planning over a JEPA world model using a finetuned VLA policy prior. PiJEPA achieves the best positional accuracy across both DINOv2 and V-JEPA-2 encoders, outperforming reactive policies, uninformed planning, and world model scoring baselines. Our analysis reveals a natural division of labor: the reactive policy excels at local control and heading prediction, uninformed MPPI achieves strong heading alignment through its embedding-space objective, and world model scoring captures much of the look-ahead benefit by filtering poor proposals—PiJEPA combines these strengths, with the additional MPPI refinement yielding the best trajectory-level coherence. The MPPI planning latency overhead is negligible relative to the accuracy gains.

However, our failure analysis reveals that the world model can become stuck in its rollouts, producing static latent predictions that render the planner ineffective. Addressing this stagnation through improved dynamics architectures or diversity-promoting rollout mechanisms is a key direction for future work, alongside intermediate waypoint costs, richer action spaces, and closed-loop replanning.

References

[1] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2017. 1

[2] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "ViNT: A foundation model for visual navigation," in *Conference on Robot Learning*, 2023. 1

[4] O. M. Team, D. Ghosh, H. R. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, P. R. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," *ArXiv*, vol. abs/2405.12213, 2024. 1, 2, 3, 4

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, 2023.

[6] C. Glossop, W. Chen, A. Bhorkar, D. Shah, and S. Levine, "CAST: Counterfactual labels improve instruction following in vision-language-action models," *arXiv preprint arXiv:2508.13446*, 2025. 1, 4

[7] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal masked diffusion policies for navigation and exploration," *arXiv preprint arXiv:2310.07896*, 2023. 1

[8] Y. LeCun, "A path towards autonomous machine intelligence," *OpenReview preprint*, 2022. 1

[9] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," in *International Conference on Machine Learning*, 2024.

[10] N. Hansen, H. Su, and X. Wang, "TD-MPC2: Scalable, robust world models for continuous control," *arXiv preprint arXiv:2310.16828*, 2024.

[11] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, "Navigation world models," *arXiv preprint arXiv:2412.03572*, 2024. 2

[12] G. Zhou, H. Pan, Y. LeCun, and L. Pinto, "Dino-wm: World models on pre-trained visual features enable zero-shot planning," 2024. 1, 2, 3

[13] B. Terver, T.-Y. Yang, J. Ponce, A. Bardes, and Y. LeCun, "What drives success in physical planning with joint-embedding predictive world models?," 2025. 1, 2, 3, 4, 5

[14] M. Assran *et al.*, "V-JEPA-2-AC: Vision joint-embedding predictive architecture with action conditioning," *arXiv preprint*, 2025. 1

[15] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE International Conference on Robotics and Automation*, 2016. 1, 3

[16] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic MPC for model-based reinforcement learning," *IEEE International Conference on Robotics and Automation*, 2017. 1, 3

[17] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and Computing in Applied Probability*, vol. 1, pp. 127–190, 1999. 1

[18] Z. Hou, T. Zhang, Y. Xiong, H. Duan, H. Pu, R. Tong, C. Zhao, X. Zhu, Y. Qiao, J. Dai, and Y. Chen, "Dita: Scaling diffusion transformer for generalist vision-language-action policy," *ArXiv*, vol. abs/2503.19757, 2025. 2

[19] S. Ye, J. Jang, B. Jeon, S. J. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Lidén, K. Lee, J. Gao, L. S. Zettlemoyer, D. Fox, and M. Seo, "Latent action pretraining from videos," *ArXiv*, vol. abs/2410.11758, 2024. 2

[20] C. Zhang, J. Wang, Z. Gao, Y. Su, T. Dai, C. Zhou, J. Lu, and Y. Tang, "Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos," *ArXiv*, vol. abs/2601.04061, 2026. 2

[21] Y. Mei, Y. Yang, L. Guo, Q. Wang, M. Yu, X. He, W. Wu, and J. Liu, "Urbannav: Learning language-guided urban navigation from web-scale human trajectories," *ArXiv*, vol. abs/2512.09607, 2025. 2

[22] Q. Garrido, T. Nagarajan, B. Terver, N. Ballas, Y. LeCun, and M. Rabbat, "Learning latent action world models in the wild," *ArXiv*, vol. abs/2601.05230, 2026. 2

[23] R. Balestriero and Y. LeCun, "Lejepa: Provable and scalable self-supervised learning without the heuristics," *ArXiv*, vol. abs/2511.08544, 2025. 2

[24] J. Sun, W. Zhang, Z. Qi, S. Ren, Z. Liu, H. Zhu, G. Sun, X. Jin, and Z. Chen, "Vla-jepa: Enhancing vision-language-action model with latent world model," 2026. 2

[25] W. Shen, Z. Meng, J. Ma, M. Zhou, and D. Xiang, "An efficient and multi-modal navigation system with one-step world model," *ArXiv*, vol. abs/2601.12277, 2026. 2

[26] D. Wang, H. Ye, Z. Liang, Z. Sun, Z. Lu, Y. Zhang, Y. Zhao, Y. Gao, M. Seegert, F. Schäfer, H. Qin, W. Li, L. Palmieri, F. Jahncke, M. Piccinini, and J. Betz, "Target-bench: Can world models achieve mapless path planning with semantic targets?," *ArXiv*, vol. abs/2511.17792, 2025. 2

[27] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024. 4

[28] M. Assran *et al.*, "V-JEPA-2: Self-supervised video models enable understanding, generation and planning," *arXiv preprint*, 2025. 4

[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. 4

[30] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020. 4

[31] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *International Conference on Computer Vision*, 2023. 5