SKIPWORLD: LATENT SKIP PLANNING WITH DISTILLED WORLD MODELS FOR EFFICIENT VISUAL NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

World-model-based visual navigation enables action planning by predicting future states under candidate trajectories. However, long-horizon rollout remains challenging: models are often trained with ground-truth visual histories but deployed in closed-loop prediction, and evaluating many candidates with step-by-step rollout is computationally expensive. We propose **SkipWorld**, a latent skip planning framework for efficient visual navigation. SkipWorld introduces three components: (i) **Latent AutoRoll**, which trains a world model to recursively roll out future states in latent space, reducing the mismatch between training and closed-loop inference; (ii) **SkipDistill**, which distills the latent autoregressive model into a skip world model that directly predicts long-stride future latent states; and (iii) **Skip-then-Refine Planning**, a dual-model coarse-to-fine planner that first filters candidate trajectories with the skip model and then reranks selected candidates with the stronger autoregressive model. This design preserves accurate trajectory evaluation while reducing redundant rollout computation. Experiments on RECON, HuRoN, SCAND, and TartanDrive show that, compared with NWM, SkipWorld reduces the average ATE by 4.2% and RPE by 2.7%, while decreasing planning time by 62.4% with a 2.7× speedup.

1 INTRODUCTION

Visual navigation requires an agent to anticipate how its actions may change future observations and whether those changes will lead it closer to the goal. Rather than directly mapping observations to actions, world-model-based methods learn to predict future states and use these predictions for planning Bar et al. (2025); Huang et al. (2026a). Given the current observation, goal, and candidate action trajectories, the model evaluates their possible outcomes and selects the sequence that best supports the navigation objective Bar et al. (2025); Chahe & Zhou (2026); Shah et al. (2023b). This explicit look-ahead is especially useful for long-horizon navigation, where reactive policies may struggle with delayed consequences Shah et al. (2023b); Garg et al. (2025); Shah et al. (2023a); Sridhar et al. (2024).

Despite this promise, efficient planning with visual world models remains difficult. A central challenge is the mismatch between training and deployment. During training, many predictive models condition on ground-truth observation histories, where each future prediction receives clean visual context from the dataset. During planning, however, the model must operate in a closed loop: once the first future state is predicted, subsequent predictions depend on previously predicted states rather than ground-truth observations Bar et al. (2025). This discrepancy can cause errors to accumulate over long horizons, degrading the reliability of predicted trajectories and weakening downstream action selection Huang et al. (2026b).

A second challenge is computational cost. Planning requires evaluating many candidate action trajectories, and each candidate may require multiple autoregressive prediction steps. When the model predicts or decodes high-dimensional images at every step, the cost grows quickly with the planning horizon, the number of samples, and the number of repeated evaluations. As a result, accurate world-model planning can become too slow for practical visual navigation, where decisions must be made repeatedly and efficiently Hansen et al. (2024); Chahe & Zhou (2026).

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

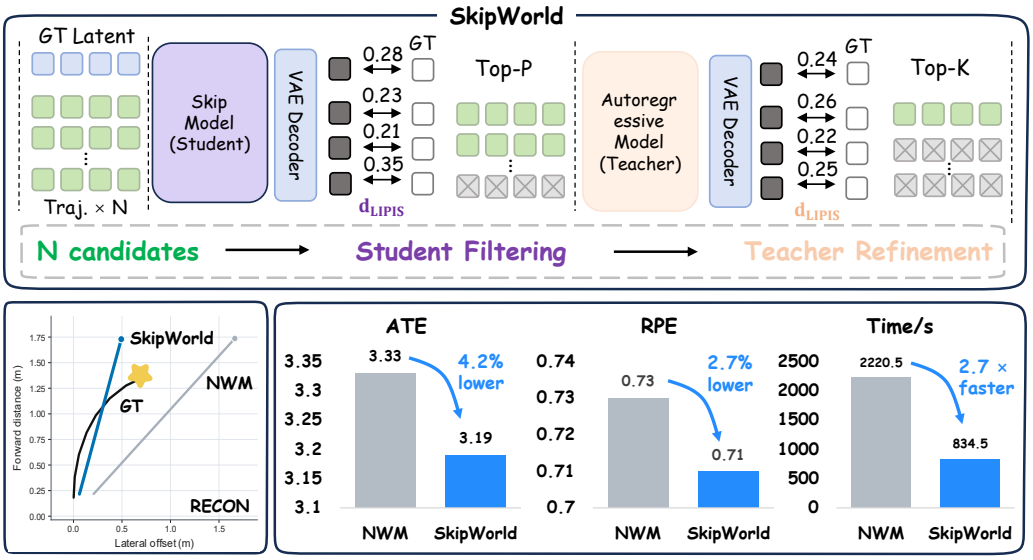

Figure 1: **Overview and performance of SkipWorld. Top:** SkipWorld performs coarse-to-fine planning by first using a skip student to filter N candidate trajectories into a compact Top-P subset, followed by latent autoregressive teacher refinement. **Bottom-left:** Example trajectory prediction on RECON, where SkipWorld more closely follows the ground-truth trajectory than NWM. **Bottom-right:** Compared with NWM, SkipWorld reduces average ATE by 4.2% and RPE by 2.7%, while achieving a 2.7× planning speedup.

One natural way to reduce this cost is to move prediction from pixel space to latent space. Latent representations provide a compact state abstraction that can preserve task-relevant visual and geometric information while avoiding unnecessary image reconstruction during planning Zhou et al. (2024); Chahe & Zhou (2026). However, simply encoding observations into a latent space is not sufficient. If the model is still trained with clean ground-truth histories but evaluated using its own predictions, the closed-loop mismatch remains. Likewise, if every candidate is still rolled out one step at a time, planning can remain expensive even when each step is cheaper.

To address these issues, we propose **SkipWorld** (Figure 1), a latent skip planning framework for efficient visual navigation. SkipWorld consists of three components. First, **Latent AutoRoll** trains a world model with closed-loop latent rollout, reducing the mismatch between training and inference Hafner et al. (2025); Zhou et al. (2024); Yu et al. (2025). Second, **SkipDistill** distills this autoregressive model into a skip world model that directly predicts long-stride future latent states, enabling efficient trajectory evaluation. Third, **Skip-then-Refine Planning** combines the two models in a coarse-to-fine planner: the skip model rapidly filters a large candidate set, while the autoregressive model refines only promising trajectories Schrittwieser et al. (2021); Hansen et al. (2024). Together, these components improve the efficiency-accuracy trade-off of long-horizon visual world-model planning.

Our contributions are summarized as follows:

- We introduce **SkipWorld**, a unified latent planning framework that integrates closed-loop latent prediction, skip distillation, and coarse-to-fine candidate evaluation.
- We develop **Latent AutoRoll** and **SkipDistill** to model long-horizon dynamics efficiently, combining autoregressive latent rollout with direct long-stride prediction.
- We propose **Skip-then-Refine Planning**, which allocates expensive autoregressive rollout only to promising candidate trajectories.
- **SkipWorld** reduces average ATE by 4.2%, RPE by 2.7%, and planning time by 62.4% compared with NWM.

2 Related Work

Visual navigation. Learning-based visual navigation aims to infer goal-directed behaviors directly from egocentric observations. Large-scale training on heterogeneous robot trajectories has substantially improved cross-environment and cross-embodiment generalization Shah et al. (2023a;b), while diffusion-based policies further enable multimodal navigation behaviors for both goal-directed navigation and exploration Sridhar et al. (2024). Despite their strong generalization, such policy-based approaches typically map observations directly to actions without explicitly reasoning about the visual consequences of candidate trajectories Bar et al. (2025); Zhou et al. (2024). This limitation has motivated predictive approaches that evaluate potential actions through their anticipated future observations Hafner et al. (2025); Hao et al. (2026).

World Models for Planning. World models enable decision making by predicting future states conditioned on candidate actions, and have been extensively studied for planning and control from visual observations Hafner et al. (2019); Hansen et al. (2024); Hafner et al. (2025). In visual navigation, **Navigation World Models (NWM)** Bar et al. (2025) formulate goal-conditioned navigation as visual foresight, predicting future observations under candidate action sequences and ranking them according to their proximity to the goal. Subsequent work has explored richer predictive representations, memory mechanisms, and alternative action parameterizations Dong et al. (2025); Mei et al. (2026); Stapf et al. (2026). **RAE-NWM** Zhang et al. (2026) performs navigation world modeling in a dense representation space, while **Learning Latent Action World Models in the Wild** Garrido et al. (2026) learns latent action dynamics from large-scale unlabeled video and transfers them to navigation through lightweight action alignment. **Beyond Language Modeling** Tong et al. (2026) further studies how multimodal pretraining data affect world-model-based navigation planning. Despite these advances, candidate trajectories are still commonly evaluated through multi-step sequential prediction, making planning increasingly expensive as the number of candidates and rollout horizon grow.

Efficient Candidate Evaluation. Efficient model-based planning has traditionally reduced computation through compact latent dynamics, efficient trajectory optimization, or learned predictive representations Hafner et al. (2019); Hansen et al. (2024). In navigation, **NWM** Bar et al. (2025) evaluates a large population of candidate action sequences by repeatedly rolling the world model forward, which provides detailed visual foresight but incurs substantial redundant computation. Recent approaches improve efficiency by reducing the computational cost of the underlying representation or rollout model Zhang et al. (2026); Stapf et al. (2026). In particular, **CompACT** Kim et al. (2026) compresses observations into a small number of discrete tokens, substantially accelerating world-model inference while retaining step-by-step candidate rollout. These approaches primarily reduce the cost of individual prediction steps, whereas the number of sequential predictions required for evaluating each candidate largely remains unchanged.

3 Method

3.1 Problem Formulation

As illustrated in Fig. 2, SkipWorld performs world-model-based visual navigation by predicting future observations under candidate action trajectories and selecting actions according to their visual discrepancy to the goal.

Let the robot observe a context window of RGB frames and a goal image. We first encode each observation into latent tokens using a pretrained VAE encoder Blattmann et al. (2023), i.e., $z_t = E(o_t)$. The latent context is represented as $C_t = [z_{t-c+1}, \dots, z_t]$, where c denotes the context size. We use t to denote the current time step and $h \in \{1, \dots, H\}$ to index future rollout steps.

The world model predicts the next latent state conditioned on the latent context, the corresponding action, and the relative temporal interval, i.e., $\hat{z}_{t+1} = F_\theta(C_t, a_{t+1}, \tau)$, where F_θ denotes the latent diffusion world model and τ is the relative temporal condition. In our implementation, the denoising network is a conditional diffusion transformer Bar et al. (2025).

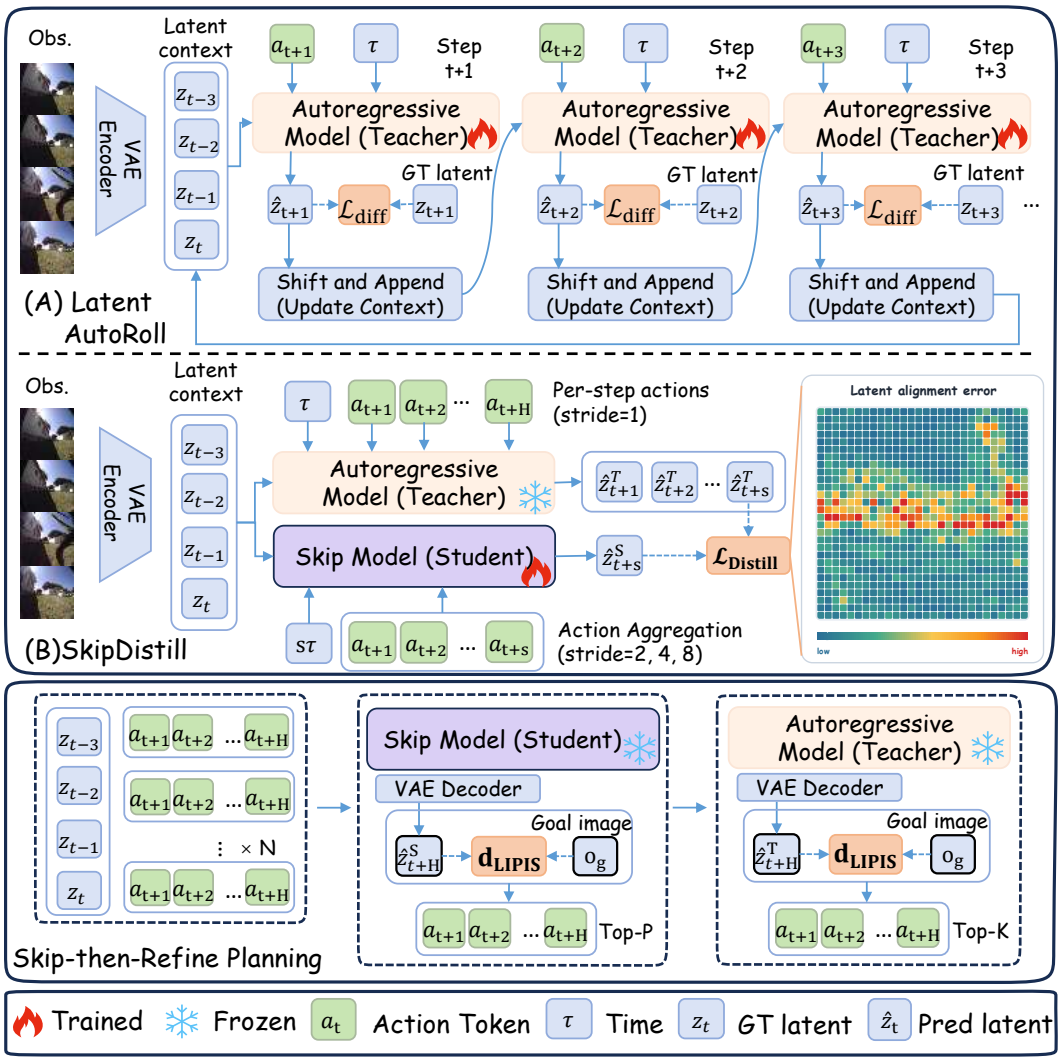

Figure 2: Overview of SkipWorld. First, **Latent AutoRoll** trains an autoregressive world model that rolls out directly in latent space. Second, **SkipDistill** trains a skip student to predict longer-horizon latent transitions through teacher endpoint distillation, where the latent alignment error (Section A.5) visualizes the spatial discrepancy between the student skip endpoint and the autoregressive teacher endpoint. Third, **Skip-then-Refine Planning** uses the skip student to cheaply filter candidate trajectories and then uses the autoregressive teacher to accurately rerank only the selected candidates.

During planning, each candidate action trajectory $a_{t+1:t+H} = [a_{t+1}, \ldots, a_{t+H}]$ is evaluated by rolling out the world model into the future. The final predicted latent $\hat{z}_{t+H}$ is decoded into image space using the VAE decoder $D(\cdot)$ and compared with the goal image o_g using the perceptual distance d_{LPIPS} Zhang et al. (2018). This distance is used as the planning cost for candidate ranking.

3.2 Latent AutoRoll

The first component, Latent AutoRoll, trains the world model under the same closed-loop latent rollout pattern used during inference. Standard world-model training typically conditions future predictions on ground-truth histories, whereas planning recursively feeds the model's own predictions back into the context. This discrepancy can cause prediction errors to accumulate over long horizons.

Latent AutoRoll explicitly exposes the model to its own predicted latents during training. Starting from the current latent context C_t, the model recursively predicts future latent states as

$$\hat{z}^T_{t+h} = F_\theta\left(C_{t+h-1}, a_{t+h}, \tau\right), \qquad h = 1, \ldots, H. \tag{1}$$

After each prediction, the context is updated as $C_{t+h} = \mathrm{shift}(C_{t+h-1}, \hat{z}^T_{t+h})$, where $\mathrm{shift}(\cdot)$ removes the oldest latent from the context window and appends the predicted clean latent as the newest context state. Therefore, predictions at later rollout steps are conditioned on previous model-generated latents rather than ground-truth future states, matching the closed-loop rollout used during planning.

For a training sequence with future target latents $\{z_{t+1}, \ldots, z_{t+H}\}$, the Latent AutoRoll objective is

$$\mathcal{L}_{\mathrm{AR}} = \frac{1}{H}\sum_{h=1}^{H} \mathcal{L}_{\mathrm{diff}}\left(\theta; z_{t+h}, C_{t+h-1}, a_{t+h}, \tau\right), \tag{2}$$

where $\mathcal{L}_{\mathrm{diff}}$ denotes the diffusion denoising objective and z_{t+h} is the ground-truth target latent at rollout step h. By recursively training on predicted latent contexts, Latent AutoRoll enables the world model to recover from its own prediction errors and reduces the mismatch between training and planning-time rollout.

3.3 SkipDistill

Although Latent AutoRoll improves consistency between training and closed-loop inference, dense autoregressive rollout remains expensive. If every candidate trajectory requires H sequential one-step model calls, planning cost grows rapidly with both the rollout horizon and the number of sampled candidates.

SkipDistill addresses this issue by distilling the dense autoregressive teacher into a skip student that directly predicts a future latent endpoint over a larger temporal interval. For a sampled skip stride $s \in \{2, 4, 8\}$, the actions over the skipped interval are accumulated as $\bar{a}_{t,s} = \sum_{i=1}^{s} a_{t+i}$. The student predicts the endpoint latent from the current context, the accumulated action, and the corresponding temporal displacement:

$$\hat{z}^S_{t+s} = F_\phi\left(C_t, \bar{a}_{t,s}, s\tau\right), \tag{3}$$

where F_ϕ denotes the skip student, $\hat{z}^S_{t+s}$ is the student-predicted endpoint latent, $\bar{a}_{t,s}$ represents the accumulated action over the skipped interval, and $s\tau$ is the scaled relative temporal condition.

The student is supervised by the endpoint produced by the frozen Latent AutoRoll teacher. Starting from the same context C_t, the teacher performs s dense one-step predictions,

$$\hat{z}^T_{t+j} = F_\theta\left(C^T_{t+j-1}, a_{t+j}, \tau\right), \qquad j = 1, \ldots, s, \tag{4}$$

with the context recursively updated as $C^T_{t+j} = \mathrm{shift}(C^T_{t+j-1}, \hat{z}^T_{t+j})$. This dense rollout produces the teacher endpoint $\hat{z}^T_{t+s}$, which serves as the distillation target for the student.

The endpoint distillation objective is

$$\mathcal{L}_{\mathrm{distill}} = \mathrm{SmoothL1}\left(\hat{z}^S_{t+s}, \mathrm{sg}\left(\hat{z}^T_{t+s}\right)\right), \tag{5}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient and $\mathrm{SmoothL1}$ Girshick (2015) is the robust regression loss used in implementation. By matching the endpoint of the dense teacher rollout, the skip student learns to approximate multiple autoregressive steps with a single prediction, substantially reducing sequential inference cost during candidate screening.

Algorithm 1 Skip-then-Refine Planning

Require: observation context o_c, goal image o_g, skip student F_ϕ, autoregressive teacher F_θ, number of candidates N, retained candidates P, top candidates K, horizon H, skip stride s, stochastic repeats R

```
Encode observation context into latent context C_t
Sample N candidate action trajectories A_N = {a^1, ..., a^N}
for each candidate a^i ∈ A_N do
    Accumulate actions over each s-step interval
    Roll out the skip student F_φ with stride s
    Decode the final predicted latent
    Compute J_skip(a^i) using LPIPS
end for
Retain the best P candidates A_P according to J_skip
for each candidate a^i ∈ A_P do
    for r = 1, ..., R do
        Roll out the teacher F_θ with stride 1
        Decode the final predicted latent
        Compute the LPIPS cost
    end for
    Average the R costs to obtain J_AR(a^i)
end for
Select the top-K candidates according to J_AR
return the mean action of the top-K candidates
```

3.4 SKIP-THEN-REFINE PLANNING

At inference time, SkipWorld combines the efficient skip student and the stronger Latent AutoRoll teacher in a coarse-to-fine planning strategy. The planner first samples N candidate action trajectories over a horizon of H steps.

In the coarse stage, the skip student evaluates all candidates using stride s. For each candidate a^i, the student recursively performs skip predictions until the planning horizon is reached. The final predicted latent $\hat{z}^{S,i}_{t+H}$ is decoded and compared with the goal image using

$$J_{\text{skip}}(a^i) = d_{\text{LPIPS}}\left(D(\hat{z}^{S,i}_{t+H}), o_g\right). \tag{6}$$

The P candidates with the lowest coarse costs are retained for dense refinement. This coarse filtering stage avoids applying the expensive autoregressive teacher to candidates that are unlikely to reach the goal.

The retained trajectories are then evaluated by the Latent AutoRoll teacher using dense stride-1 rollout. Since diffusion-based world models are stochastic, each retained candidate is evaluated with R independent rollouts. The refined planning score is

$$J_{\text{AR}}(a^i) = \frac{1}{R}\sum_{r=1}^{R} d_{\text{LPIPS}}\left(D(\hat{z}^{T,i,r}_{t+H}), o_g\right), \tag{7}$$

where $\hat{z}^{T,i,r}_{t+H}$ denotes the final latent predicted for candidate a^i in the r-th dense teacher rollout.

The retained trajectories are reranked according to J_{AR}. The top-K candidates with the lowest refined costs are selected, and their actions are averaged to obtain the final action. In this way, the skip student performs efficient coarse screening over the full candidate set, while the stronger autoregressive teacher is reserved only for the most promising trajectories.

Table 1: Main comparison on four visual navigation benchmarks. Lower is better. The best result is shown in bold and the second best is underlined. NWM (original) is from its original paper. NWM (pretrained) denotes our four-dataset retrained baseline. Due to privacy restrictions, only a low-resolution HuRoN subset is available. For fair comparison, We thus fine-tune it from the 100K-step checkpoint for another 50K steps with identical settings.

Model	RECON		HuRoN		TartanDrive		SCAND		Avg.	
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
GNM	1.87	0.73	<u>3.71</u>	1.00	6.65	1.62	2.12	0.61	3.59	0.99
NoMaD	1.95	0.53	3.73	0.96	6.32	1.31	2.24	0.49	3.56	0.82
NWM + NoMaD ($\times$16)	1.88	0.51	3.73	<u>0.95</u>	6.26	1.30	2.18	0.48	3.51	0.82
NWM + NoMaD ($\times$32)	1.79	0.49	3.68	<u>0.95</u>	6.25	1.29	2.19	0.47	3.47	0.80
NWM (original)	**1.13**	<u>0.35</u>	4.12	0.96	**5.63**	**1.18**	1.28	0.33	**3.04**	**0.71**
CompAct	1.33	0.39	-	-	-	-	1.36	0.34	-	-
RAE-NWM	1.36	0.37	**2.91**	**0.70**	-	-	1.14	**0.28**	-	-
NWM (pretrained)	1.38	<u>0.35</u>	5.08	1.08	5.79	1.21	<u>1.08</u>	<u>0.29</u>	3.33	<u>0.73</u>
SkipWorld	<u>1.19</u>	**0.34**	4.86	1.04	<u>5.74</u>	<u>1.20</u>	**0.98**	**0.28**	<u>3.19</u>	**0.71**

3.5 Complexity Analysis

We compare planning complexity in terms of world-model rollout calls. Let N be the number of sampled candidate trajectories, H the prediction horizon, s the skip stride, P the number of candidates retained after coarse screening, and R the number of stochastic repeats used during dense refinement.

The original dense planner evaluates every candidate using stride-1 autoregressive rollout. Since each of the N candidates requires H world-model calls for each of the R stochastic evaluations, its rollout complexity is

$$\mathcal{C}_{\mathrm{NWM}} = O(NRH). \tag{8}$$

SkipWorld replaces the dense rollout used during coarse candidate evaluation with the skip student. Evaluating all candidates with stride s requires approximately H/s student calls per trajectory, while the expensive dense teacher is used only for the retained P candidates. The resulting complexity is

$$\mathcal{C}_{\mathrm{SkipWorld}} = O\left(N\frac{H}{s}\right)_{\mathrm{student}} + O(PRH)_{\mathrm{teacher}}. \tag{9}$$

With $N = 120$, $H = 8$, $s = 8$, $P = 20$, and $R = 3$, the dense planner requires 2880 dense rollout steps, whereas SkipWorld uses 120 skip-student steps and 480 dense teacher steps. This reduces expensive teacher rollout computation by $6\times$. In practice, the reduced rollout cost translates to a $2.51\times$ end-to-end speedup, with the remaining overhead mainly arising from latent decoding, LPIPS scoring Zhang et al. (2018), data movement, and prediction saving.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate SkipWorld on four visual navigation benchmarks: RECON Shah et al. (2021), HuRoN Hirose et al. (2023),TartanDrive Triest et al. (2022), and SCAND Karnan et al. (2022). These benchmarks cover diverse navigation scenarios, including indoor navigation, human-centered navigation, off-road driving, and social navigation. We follow the standard planning evaluation protocol used by prior navigation world model methods.

Evaluation Metrics. We report absolute trajectory error (ATE) Sturm et al. (2012) and relative pose error (RPE) Sturm et al. (2012). ATE measures global trajectory deviation, while RPE measures local motion consistency. Lower values indicate better performance.

Table 2: Component ablation averaged over four datasets. Time is seconds per query. Lower is better. The best result is shown in bold and the second best is underlined.

Variant	AutoRoll	SkipDistill	Refine	ATE	RPE	Time
NWM	–	–	–	3.33	0.73	2220.5
Latent AutoRoll	✓	–	–	3.26	0.72	3807.9
AutoRoll + SkipDistill	✓	✓	–	3.26	0.73	**465.3**
AutoRoll + Refine	✓	–	✓	<u>3.20</u>	**0.71**	991.7
SkipWorld	✓	✓	✓	**3.19**	**0.71**	<u>834.5</u>

Table 3: Ablation on skip stride with candidate budget fixed to $K_c = 20$. Time is seconds per query. Lower is better. The best result is shown in bold and the second best is underlined.

Stride	RECON		HuRoN		TartanDrive		SCAND		Avg.		Time
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	
2	1.55	0.39	4.97	1.06	5.90	1.23	1.06	0.29	3.37	0.74	1257.7
4	<u>1.30</u>	<u>0.36</u>	<u>4.94</u>	<u>1.05</u>	**5.72**	**1.19**	<u>1.02</u>	**0.28**	<u>3.24</u>	<u>0.72</u>	<u>973.7</u>
8	**1.19**	**0.34**	**4.86**	**1.04**	<u>5.74</u>	<u>1.20</u>	**0.98**	**0.28**	**3.19**	**0.71**	**834.5**

Baselines. We compare SkipWorld with representative visual navigation methods. **GNM** Shah et al. (2023a) is a general goal-conditioned navigation model trained on heterogeneous robot data. **NoMaD** Sridhar et al. (2024) employs a unified diffusion policy for goal-directed navigation and exploration. **NWM** Bar et al. (2025) is a navigation world model that evaluates action candidates by autoregressively predicting future visual observations. **CoMPACT** Kim et al. (2026) performs efficient world-model planning using compact discrete visual tokens. **RAE-NWM** Zhang et al. (2026) models navigation dynamics in a dense visual representation space using a representation autoencoder, improving future prediction and downstream planning.

Implementation Details. For **Latent AutoRoll**, we train for 50K steps from the 100K-step NWM checkpoint with a 16-frame horizon on four datasets. For **SkipDistill**, the student starts from the same 100K-step checkpoint and is distilled from the frozen 150K-step teacher using $\mathcal{S} = 2, 4, 8$ for 50K steps. For **planning**, unless otherwise stated, the planner samples $N = 120$ candidates, keeps $K = 5$ elites, and performs one CEM update step. The default SkipWorld configuration uses coarse stride $s_c = 8$, preselection size $P = 20$, and $R = 3$ teacher repeats. Intermediate rollout states remain in latent space, and only the final predicted latent is decoded for scoring.

4.2 Main Results

Table 1 summarizes quantitative comparisons across four visual-navigation benchmarks. Our SkipWorld achieves competitive overall ATE and RPE performance against prior world-model-based navigation methods. Note that NWM (original) is fully trained for 100K steps from the original work, whereas our NWM (pretrained) baseline only receives an additional 50K-step fine-tune, which partially accounts for its performance gap versus NWM (original). Even under this limited training budget, SkipWorld obtains strong average-tracking metrics and yields clear improvements over our retrained NWM (pretrained) baseline.

4.3 Ablation Studies

Does each component contribute? We first ablate the three main components of SkipWorld: Latent AutoRoll, SkipDistill, and Skip-then-Refine Planning. Table 2 shows that Latent AutoRoll improves planning accuracy over NWM, but it is expensive when used as a dense autoregressive planner. SkipDistill greatly reduces runtime, although using the skip model alone is less reliable than using refinement. SkipWorld combines these two behaviors: it stays close to the accuracy of AutoRoll-based refinement while reducing the cost of planning.

How large should the skip stride be? We further ablate the skip stride for coarse candidate screening. As shown in Table 3, smaller strides yield negligible accuracy improvements at the cost of

Table 4: Ablation on coarse candidate budget with skip stride fixed to $s = 8$. Time is seconds per query. Lower is better. The best result is shown in bold and the second best is underlined.

Budget	RECON		HuRoN		TartanDrive		SCAND		Avg.		Time
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	
$P = 5$	1.32	0.36	4.92	1.06	**5.67**	1.19	1.07	0.29	3.25	0.73	**460.5**
$P = 10$	1.33	0.36	4.89	1.05	**5.67**	**1.18**	**0.96**	0.28	3.21	0.72	610.1
$P = 20$	1.19	0.34	**4.86**	**1.04**	5.74	1.20	0.98	0.28	**3.19**	**0.71**	834.5
$P = 40$	**1.15**	**0.33**	4.88	1.05	5.83	1.21	0.98	**0.27**	3.21	0.72	1436.1

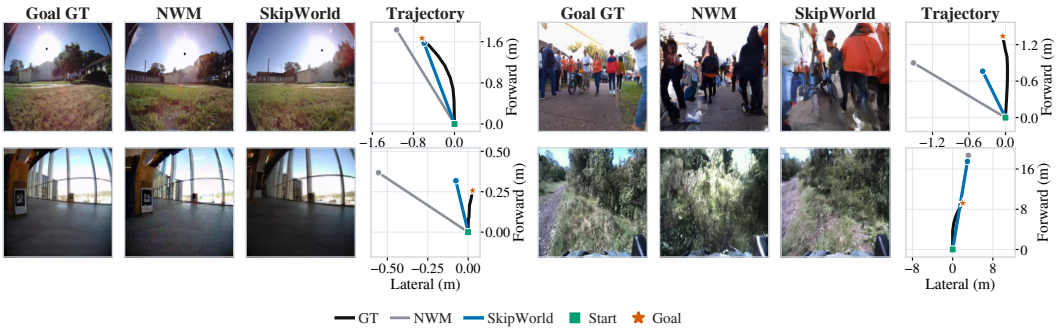

Figure 3: Predicted trajectories on the four evaluation datasets. SkipWorld produces trajectories that better follow the ground-truth path across diverse indoor, outdoor, and driving scenes.

more rollout calls. Stride-8 achieves favorable runtime with near-peak accuracy, verifying the skip model's effective long-horizon ranking capability. We adopt stride 8 in our final SkipWorld setup.

How many candidates should be retained? Finally, we vary the number of candidates retained after coarse screening. Table 4 shows the expected trade-off: a small budget is fast but can remove useful candidates too early, while a larger budget gives the refinement stage more choices at a higher cost. The improvement from increasing the budget is not uniform across datasets, which indicates that candidate filtering quality depends on the environment and trajectory distribution. In practice, a moderate budget offers a reasonable balance between preserving good candidates and avoiding excessive teacher evaluation.

4.4 Qualitative Results

Figure 3 visualizes representative predicted trajectories on the four evaluation datasets. Compared with the original NWM planner, SkipWorld tends to produce trajectories that better follow the ground-truth future motion, especially when the original planner selects candidates with noticeable lateral drift. These examples provide qualitative evidence that latent skip scoring does not merely accelerate planning, but can also guide the planner toward more plausible action candidates before final refinement.

5 Conclusion

We presented SkipWorld, a latent skip planning framework for efficient visual navigation. SkipWorld first trains a latent autoregressive world model to reduce the mismatch between training and closed-loop inference, then distills it into a skip student that directly predicts long-stride future latent states. During planning, the skip student efficiently filters candidate trajectories, while the autoregressive teacher refines only the selected candidates. Experiments on four navigation datasets show that this design improves the speed–accuracy trade-off over dense world-model planning. These results suggest that latent skip prediction is a practical way to make visual world-model planning more efficient without sacrificing trajectory quality.

AI USE STATEMENT

During the preparation of this paper, we used generative AI tools for grammar checking, and formatting assistance. All technical contributions, including problem formulation, methodology design, experimental implementation, result analysis, and conclusions, were developed and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work does not involve human subjects, personally identifiable information, or sensitive data collection. The datasets used in our experiments are publicly available benchmarks for visual navigation research. We do not identify any additional ethical concerns related to the methodology or applications presented in this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our methodology, model architecture, training procedures, and experimental settings in the main paper and supplementary materials. We report the datasets, evaluation protocols, implementation details, and hyperparameters necessary to reproduce our experiments. We will release the source code and configurations to facilitate further research and verification of our results.

REFERENCES

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801. IEEE, 2025.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Amirhosein Chahe and Lifeng Zhou. Policy-guided world model planning for language-conditioned visual navigation. *arXiv preprint arXiv:2603.25981*, 2026.

Yifei Dong, Fengyi Wu, Guangyu Chen, Zhi-Qi Cheng, Qiyu Hu, Yuxuan Zhou, Jingdong Sun, Jun-Yan He, Qi Dai, and Alexander G Hauptmann. Unified world models: Memory-augmented planning and foresight for visual navigation. 2025.

Sourav Garg, Dustin Craggs, Vineeth Bhat, Lachlan Mares, Stefan Podgorski, Madhava Krishna, Feras Dayoub, and Ian Reid. Objectreact: Learning object-relative control for visual navigation. *arXiv preprint arXiv:2509.09594*, 2025.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. In *Forty-third International Conference on Machine Learning*, 2026.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

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

Haihong Hao, Lei Chen, Mingfei Han, Changlin Li, Dong An, Yuqiang Yang, Zhihui Li, and Xiaojun Chang. Latentpilot: Scene-aware vision-and-language navigation by dreaming ahead with latent visual reasoning. *arXiv preprint arXiv:2603.29165*, 2026.

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 9(1):49–56, 2023.

Changxin Huang, Lv Tang, Zhaohuan Zhan, Lisha Yu, Runhao Zeng, Zun Liu, Zhengjie Wang, and Jianqiang Li. Unemo: Collaborative visual-language reasoning and navigation via a multimodal world model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18315–18323, 2026a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026b.

Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7 (4):11807–11814, 2022.

Dongwon Kim, Gawon Seo, Jinsung Lee, Minsu Cho, and Suha Kwak. Planning in 8 tokens: A compact discrete tokenizer for latent world model. *arXiv preprint arXiv:2603.05438*, 2026.

Yanghong Mei, Longteng Guo, Ming-Ming Yu, Guiyu Zhao, Xingjian He, and Jing Liu. Navwm: A unified navigation world model for foresight-driven planning. *arXiv preprint arXiv:2606.24101*, 2026.

Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34:27580–27591, 2021.

Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. *arXiv preprint arXiv:2104.05859*, 2021.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226–7233. IEEE, 2023a.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023b.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Sebastian Stapf, Pablo Acuaviva, Aram Davtyan, and Paolo Favaro. Composition of memory experts for diffusion world models. In *International Conference on Learning Representations*, volume 2026, pp. 15377–15403, 2026.

Jürgen Sturm, Wolfram Burgard, and Daniel Cremers. Evaluating egomotion and structure-from-motion approaches using the tum rgb-d benchmark. In *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, volume 13, pp. 6, 2012.

Shengbang Tong, David Fan, John Nguyen, Ellis Brown, Gaoyue Zhou, Shengyi Qian, Boyang Zheng, Théophane Vallaeys, Junlin Han, Rob Fergus, et al. Beyond language modeling: An exploration of multimodal pretraining. *arXiv preprint arXiv:2603.03276*, 2026.

Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2546–2552. IEEE, 2022.

Ziyang Yu, Wenbing Huang, and Yang Liu. Unisim: a unified simulator for time-coarsened dynamics of biomolecules. *arXiv preprint arXiv:2506.03157*, 2025.

Mingkun Zhang, Wangtian Shen, Fan Zhang, Haijian Qin, Zihao Pei, and Ziyang Meng. Rae-nwm: Navigation world model in dense visual representation space. *arXiv preprint arXiv:2603.09241*, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

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

A Appendix

A.1 Reproducibility and Evaluation Details

We provide additional implementation and evaluation details for reproducibility. Unless otherwise stated, all planning experiments use the same candidate optimization setting summarized in Table 5. The planner samples $N = 120$ candidate action trajectories, retains the top $K = 5$ candidates, and performs one CEM update step. Dense autoregressive scoring is repeated $R = 3$ times to reduce stochastic variance. The default SkipWorld configuration uses a coarse stride $s_c = 8$ and retains $P = 20$ candidates for dense refinement.

Table 5: Default planning hyperparameters used in our experiments.

Hyperparameter	Value
Candidate samples N	120
Elite candidates K	5
CEM update steps	1
Batch size	1
Number of workers	12
Dense repeat evaluations R	3
Coarse repeat evaluations	1
Fine rollout stride	1
Default coarse stride s_c	8
Default preselection size P	20
Prediction horizon H	8 frames
Evaluation frame rate	4 FPS

All methods are evaluated on the same test split for each dataset. The default prediction horizon is 8 future frames, corresponding to 2 seconds at 4 FPS. Runtime is measured as the average wall-clock planning time per query and includes candidate rollout, final latent decoding, perceptual scoring, and evaluation overhead under the same implementation setting.

A.2 Full Component Ablation Results

The main paper reports the averaged component ablation results. Here, Table 6 provides the full dataset-wise results.

Latent AutoRoll improves average trajectory accuracy over NWM by reducing the mismatch between training and closed-loop rollout, although dense autoregressive planning remains computationally expensive. Skip-only planning substantially reduces runtime by replacing dense rollout with long-stride prediction during candidate evaluation. Adding dense refinement improves robustness by reranking the shortlisted candidates with the stronger autoregressive teacher. Overall, SkipWorld achieves the best runtime among the refined variants while maintaining comparable trajectory accuracy.

Table 6: Full dataset-wise component ablation. Time is average planning time per query. Lower is better. The best result is shown in bold and the second best is underlined.

Method	RECON		HuRoN		TartanDrive		SCAND		Avg.		Time
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	
NWM	1.381	0.351	5.076	1.083	5.791	1.209	1.081	0.290	3.332	0.733	2220.5
Latent AutoRoll	1.245	**0.336**	4.909	1.048	5.846	1.216	1.020	0.277	3.255	0.720	3807.9
AutoRoll + SkipDistill	1.487	0.380	4.846	1.043	**5.638**	**1.178**	1.067	0.297	3.260	0.725	**465.3**
AutoRoll + Refine	**1.191**	**0.336**	**4.839**	**1.038**	5.798	1.208	0.987	**0.271**	3.204	**0.713**	991.7
SkipWorld	**1.191**	0.344	4.859	1.039	5.736	1.196	**0.980**	0.277	**3.192**	0.714	834.5

Table 7: Horizon-wise error analysis averaged over four datasets. Lower is better. The best result is shown in bold and the second best is underlined.

Method	0.5s		1.0s		1.5s		2.0s	
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
NWM	1.134	0.708	1.884	0.706	2.607	0.716	3.332	0.733
Latent AutoRoll	1.119	0.694	1.854	0.693	2.551	0.700	3.255	0.720
AutoRoll+SkipDistill	1.092	0.682	1.821	0.687	2.532	0.701	3.260	0.725
AutoRoll+Refine	1.098	0.681	1.816	0.682	2.505	0.693	3.204	**0.713**
SkipWorld	**1.084**	**0.673**	**1.797**	**0.677**	**2.487**	**0.691**	**3.192**	0.714

A.3 Horizon-wise Error Analysis

In addition to the final 2-second planning metric reported in the main paper, we evaluate trajectory accuracy at intermediate prediction horizons. Specifically, each saved trajectory prediction is truncated at the corresponding horizon, after which ATE and RPE are recomputed. Table 7 summarizes the resulting horizon-wise errors. No additional planning or model inference is performed for this analysis; all results are derived from the same saved predictions used in the main experiments.

A.4 Loss Ablation for SkipDistill

We further study the training objective of the skip student in Table 8. Given a latent context $C_t = \{z_{t-K+1}, \dots, z_t\}$ and a skip stride s, the student predicts the future endpoint latent at $t+s$ conditioned on the accumulated action $\bar{a}_{t,s} = \sum_{i=1}^{s} a_{t+i}$ and the relative temporal condition $s\tau$.

Direct endpoint supervision. A straightforward choice is to directly supervise the skip student using the ground-truth future latent z_{t+s}. The corresponding diffusion objective is

$$\mathcal{L}_{\text{direct}} = \mathcal{L}_{\text{diff}}\left(F_\phi, z_{t+s}, C_t, \bar{a}_{t,s}, s\tau\right), \tag{10}$$

where F_ϕ denotes the skip student and $\mathcal{L}_{\text{diff}}$ is the diffusion denoising loss.

Teacher endpoint distillation. Alternatively, we supervise the student using the endpoint produced by the frozen Latent AutoRoll teacher. Starting from the same context C_t, the teacher performs s dense one-step transitions to obtain $\hat{z}^T_{t+s}$. The student prediction $\hat{z}^S_{t+s}$ is then aligned with this teacher endpoint using

$$\mathcal{L}_{\text{distill}} = \text{SmoothL1}\left(\hat{z}^S_{t+s}, \text{sg}\left(\hat{z}^T_{t+s}\right)\right), \tag{11}$$

where $\text{sg}(\cdot)$ denotes stop-gradient.

Ablation setting. We compare three training objectives: $\mathcal{L}_{\text{direct}}$, $\mathcal{L}_{\text{direct}} + \mathcal{L}_{\text{distill}}$, and $\mathcal{L}_{\text{distill}}$. All variants are evaluated using the same Skip-then-Refine planner with skip stride $s = 8$ and preselection budget $P = 20$, so that the comparison isolates the effect of the skip-student training objective.

Table 8: Loss ablation of SkipDistill under the same Skip-then-Refine planning setting. Time is average planning time per query. Lower is better. The best result is shown in bold and the second best is underlined.

Training Loss	RECON		HuRoN		TartanDrive		SCAND		Avg.		Time
	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	
$\mathcal{L}_{\text{direct}}$	1.235	0.345	4.945	1.056	5.807	1.208	<u>0.996</u>	**0.275**	3.246	0.721	**824.5**
$\mathcal{L}_{\text{direct}} + \mathcal{L}_{\text{distill}}$	**1.168**	**0.331**	<u>4.896</u>	<u>1.048</u>	<u>5.779</u>	<u>1.201</u>	0.999	<u>0.276</u>	<u>3.210</u>	**0.714**	839.9
$\mathcal{L}_{\text{distill}}$	<u>1.191</u>	<u>0.344</u>	**4.859**	**1.039**	**5.736**	**1.196**	**0.980**	0.277	**3.192**	**0.714**	<u>834.5</u>

Results. As shown in Table 8, distillation-only achieves the best average ATE while remaining competitive in RPE. In contrast, direct endpoint supervision performs worse on average, indicating that the endpoint generated by the autoregressive teacher provides a more effective long-stride training target than the ground-truth endpoint alone. Combining the two objectives does not provide a consistent improvement over distillation-only. We therefore use $\mathcal{L}_{\text{distill}}$ as the default SkipDistill objective in all main experiments.

A.5 Latent Alignment Error Visualization

To visualize the endpoint matching learned by SkipDistill, we compute the latent alignment error between the skip student and the autoregressive teacher. Given the same latent context C_t and action sequence, the teacher autoregressively rolls out s one-step transitions to obtain the endpoint latent $\hat{z}^T_{t+s}$, while the skip student directly predicts the endpoint latent $\hat{z}^S_{t+s}$ using the accumulated action. For each spatial location (u, v) in the latent feature map, we compute

$$E(u,v) = \sqrt{\frac{1}{C}\sum_{c=1}^{C}\left(\hat{z}^S_{t+s,c,u,v} - \hat{z}^T_{t+s,c,u,v}\right)^2}. \tag{12}$$

The resulting heatmap shows where the student prediction is well aligned with the teacher rollout in latent space. Lower values indicate closer agreement between the long-stride student transition and the teacher's autoregressive dynamics.

A.6 Additional Qualitative Results

The main paper presents representative qualitative examples across the evaluated datasets. Here, we provide additional per-dataset visualizations for RECON, SCAND, HuRoN, and TartanDrive in Figures 4, 5, 6, and 7, respectively.

Each row contains two planning queries. For each query, we show the goal image, the predictions produced by NWM and SkipWorld, and the corresponding planned trajectories. These examples complement the quantitative results by illustrating how the two planners behave across different visual scenes and navigation conditions.

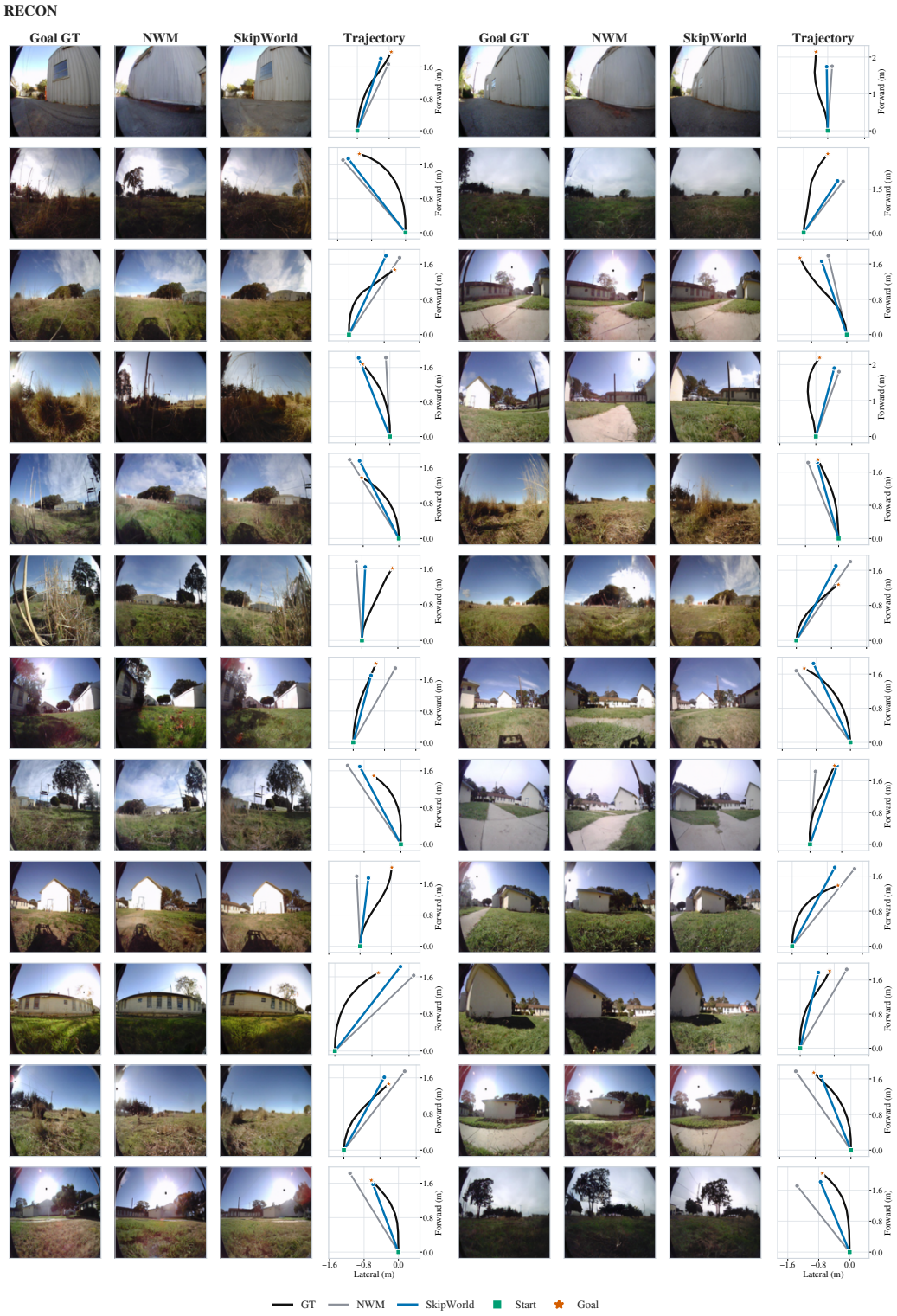

Figure 4: Additional scene-aware qualitative results on RECON.

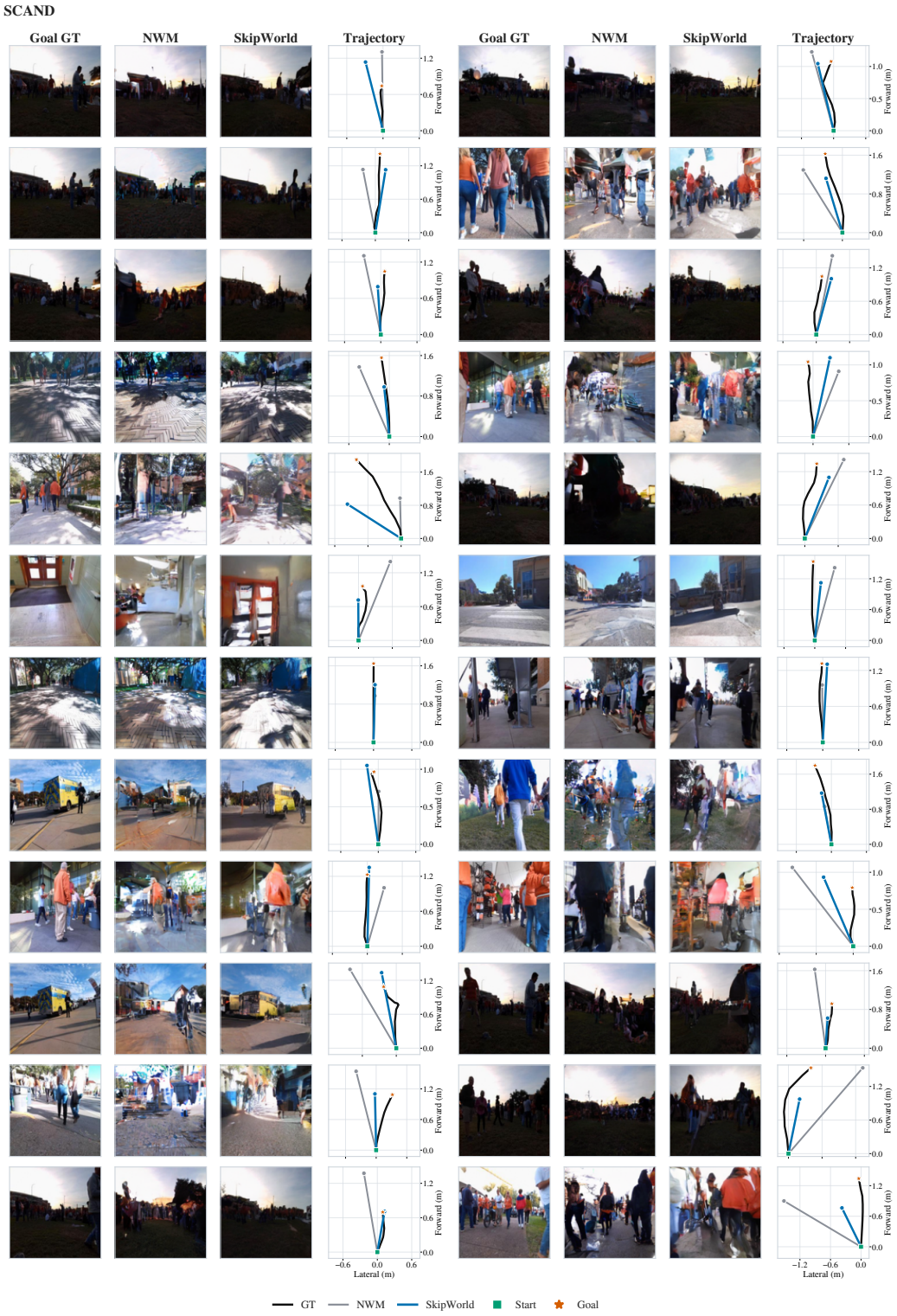

Figure 5: Additional scene-aware qualitative results on SCAND.



Figure 6: Additional scene-aware qualitative results on HuRoN.



Figure 7: Additional scene-aware qualitative results on TartanDrive.