

WorldFly: A World-Model-Based Vision-Language-Action Model for UAV Navigation

Shengtao Zheng^{1*} Kai Li^{1*} Weichen Zhang¹ Yu Meng¹

Chen Gao^{2†} Xinlei Chen^{1†} Yong Li² Xiao-Ping Zhang¹

¹Tsinghua Shenzhen International Graduate School

²BNRist, Tsinghua University

{st_zheng24, likai24, zhangwc23, mengy24}@mails.tsinghua.edu.cn

chgao96@tsinghua.edu.cn, chen.xinlei@sz.tsinghua.edu.cn

liyong07@tsinghua.edu.cn, xpzhang@ieee.org

Abstract

End-to-end Vision-Language-Action (VLA) models have shown promise in UAV navigation. However, existing approaches typically rely on historical observations to directly predict actions, often struggling in dense urban environments where severe occlusions and sharp turns result in drastic viewpoint transitions. We argue that the ability to “imagine” future states—inherent in World Models—is critical for robust decision-making under such partial observability. To address this, we construct a challenging Urban Canyon Traversal Benchmark, specifically designed to evaluate spatial understanding in scenarios characterized by severe occlusions and drastic viewpoint transitions. To this end, we propose WorldFly, a novel world-model-based VLA framework that employs a dual-branch coupled flow matching mechanism to jointly generate future video predictions and navigation actions, thereby explicitly guiding the agent’s policy via spatial imagination. Extensive evaluations on our benchmark demonstrate that WorldFly outperforms other baselines, particularly in unseen environments, validating the effectiveness of integrating world models into embodied aerial agents.

1 Introduction

In urban low-altitude environments, ground mobile robots are often constrained by complex factors such as traffic conditions, cluttered infrastructure, and limited fields of view. As a result, they typically require sophisticated behavioral strategies and extensive prior mapping to achieve effective navigation. In contrast, Unmanned Aerial Vehicles (UAVs) benefit from low-altitude flight capabilities that allow them to bypass many ground-level constraints, enabling more flexible and efficient navigation in urban spaces and demonstrating superior accessibility.

Recent works, such as OpenFly [5] and UAV-Flow [14], have begun to explore UAV navigation driven directly by human natural language instructions without expert intervention. Under the VLA paradigm, these methods aim to end-to-end generate navigation actions based on first-person-view (FPV) visual observations and language instructions. Despite promising progress, existing UAV VLA approaches, such as OpenVLA [7] and PI-0-UAV [14], primarily rely on historical and current observations to predict future actions. They lack explicit estimation of future scene states, which significantly limits their ability to perform long-horizon action prediction in previously unseen environments. Moreover, when UAVs execute large-magnitude motions, observations across

*Equal contribution.

†Corresponding authors.

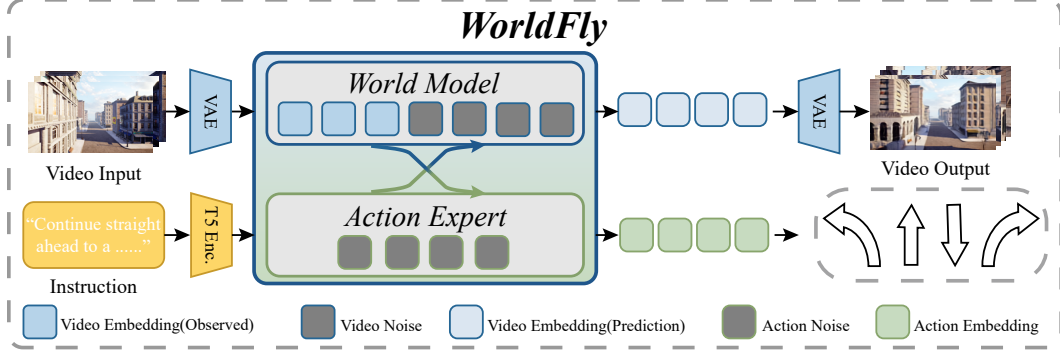


Figure 1: The overall architecture of our work. A dual-branch coupled flow matching mechanism is used to jointly generate future video predictions and navigation actions, integrating world modeling and policy learning in a unified Vision-Language-Action framework.

time steps can exhibit substantial semantic variation, making it difficult to temporally align visual observations with language instructions.

Meanwhile, in domains such as autonomous driving and video generation, world models have demonstrated strong capabilities in generating first-person-view videos conditioned on language descriptions, showcasing powerful alignment between text and visual modalities. Furthermore, interactive world models such as NWM [1], Matrix-Games [6], and YUME [11] have shown an impressive ability to model visual dynamics and action representations jointly. These advances motivate our core insight: by leveraging a world model to generate future visual trajectories conditioned on UAV instructions, we can extract motion cues from imagined videos to predict future actions and thereby enable more robust UAV navigation.

However, incorporating world models into UAV VLA frameworks presents several key challenges: 1) Existing UAV navigation datasets often overlook the extreme viewpoint variations and complex spatial constraints typical of urban "canyon" environments. Most benchmarks focus on tasks in low-clutter settings, failing to evaluate a model's ability to handle instructions under drastic perspective shifts. 2) Current UAV VLA frameworks often treat visual perception and action prediction as sequential or loosely coupled modules. These models typically rely on reactive mapping of historical observations to current actions, lacking an integrated mechanism to jointly optimize future scene "imagination" with control policy learning. 3) Purely reactive VLA models struggle with "short-sightedness" in unfamiliar environments. Without an explicit estimation of future states or a world model to provide dense motion priors, these models cannot maintain temporal consistency, leading to failure when language instructions require navigating across long distances with significant semantic variation.

To address these challenges, we propose **WorldFly**, a unified world-model-enhanced VLA framework for UAV navigation. First, to tackle the first limitation, we introduce a new benchmark named Urban Canyon Traversal, which is characterized by large viewpoint variations and extended navigation trajectories. This benchmark is specifically designed to stress-test the generative capability of world models in complex urban environments. Second, to improve architectural coupling between imagination and policy learning, we propose a dual-branch coupled architecture that enables the model to simultaneously generate future visual representations and navigation actions. This design jointly optimizes world modeling and action prediction within a single framework, avoiding a separately trained serial world-model-then-policy pipeline. Finally, we validate that our joint modeling of future video and action trajectories significantly enhances navigation robustness across diverse difficulty levels. Experimental results show that WorldFly outperforms existing baselines by utilizing imagined scenes as predictive guidance, successfully reducing the "short-sighted" failures typical of purely reactive VLA models. Our contributions are summarized as follows:

- We introduce a novel and challenging benchmark, **Urban Canyon Traversal**, designed to evaluate instruction-following capabilities of VLA models under significant viewpoint variations and long-horizon navigation requirements. The benchmark quantitatively measures UAV navigation performance in both seen and unseen urban environments.

- We propose **WorldFly**, the first framework to integrate world models into UAV VLA tasks. WorldFly employs a **Dual-Branch Coupled Architecture** that jointly optimizes future scene imagination and action generation, enabling future scene imagination while simultaneously generating navigation actions.
- Experiments show that **WorldFly** outperforms baseline models on the **Urban Canyon Traversal** benchmark, while jointly achieving future video generation and navigation action prediction within a unified framework.

2 Related Work

2.1 UAV Navigation VLA

UAV Vision-Language-Action (VLA) models generate future navigation actions by processing UAV visual observations and human language instructions. Existing works primarily adopt two paradigms within the VLA domain: the autoregressive action prediction architecture based on OpenVLA [7] (e.g., OpenFly [5], RaceVLA [12], OpenVLA-UAV [14]) and the Flow Matching architecture [10] based on π_0 [2] (e.g., Pi-0-UAV [14]).

RaceVLA [12] utilizes 4-DoF velocity vectors as UAV control signals. It constructs a UAV motion dataset in real-world environments and builds navigation tasks for racing scenarios based on the OpenVLA model. OpenFly [5], fine-tuned from OpenVLA, proposes a keyframe-aware navigation model. It extracts critical visual features during flight via an adaptive frame-level sampling mechanism and employs visual token fusion to effectively reduce information redundancy between adjacent frames while preserving historical visual memory. UAV-Flow [14] introduces a fine-grained, trajectory-guided instruction navigation benchmark. It trains OpenVLA-UAV and Pi-0-UAV to focus on understanding low-level flight semantics, thereby evaluating the UAV’s capabilities regarding spatial perception and motion correlation.

However, these methods typically map historical observations directly to actions, lacking the capability to explicitly predict or “imagine” future environmental states. This limits their robustness in complex urban scenarios where spatial foresight is crucial for handling severe occlusions and drastic viewpoint transitions.

2.2 World Model VLA

Currently, VLA models based on World Models predominantly emerge in the field of robotic manipulation. World Model VLAs construct a deep integration of visual understanding, action generation, and environmental simulation, incorporating the spatial reasoning capabilities of world models into VLA architectures to enhance adaptability in complex scenes.

Genie-Envisioner [9] comprises two structural components: GE-Base, a large-scale instruction-conditioned video diffusion model trained on the AgiBot World dataset to predict future manipulation scenes; and GE-Act, an action head that maps latent visual feature representations into executable action trajectories via a lightweight flow matching decoder. VideoVLA [13] employs the VAE encoder from CogVideoX for video prediction, utilizing a unified vision-action structure and a Flow Matching architecture to simultaneously predict future video frames and generate actions. UVA [8] focuses on the co-optimization of video generation and action prediction. It proposes a unified video-action latent feature representation alongside a decoupled diffusion decoding architecture. WorldVLA [3] integrates the VLA model and the world model into a single autoregressive framework, achieving unified understanding and generation of both actions and images. MinD [4] addresses the computational inefficiency of video generation models in real-time robotic control by proposing a dual-system diffusion architecture that decouples low-frequency visual imagination from high-frequency action generation.

Despite these advancements, existing World Model-based approaches remain largely confined to robotic manipulation within static, small-scale environments (e.g., tabletops). Applying them to UAV navigation presents unique challenges, primarily the need to handle large-scale urban topologies with drastic viewpoint changes and the stricter real-time inference requirements for aerial safety. WorldFly bridges this gap by extending the World Model VLA paradigm to the challenging domain of embodied aerial navigation.

primitive is executed; otherwise, it executes a “stop” action to terminate the navigation task. This design enables the flow matching model to operate in a continuous latent space while ensuring safe and precise discrete maneuvering. We do not include a backward primitive because the expert A* trajectories in our benchmark are forward-progressing, and reverse motion is typically unsafe under first-person low-altitude navigation with limited rear visibility. Simultaneously, the model imagines a sequence of future visual observations:

$$\mathbf{V}_t = \{v_j \in \mathbb{R}^{H \times W \times 3}\}_{j=t+1}^{t+K}, \quad (3)$$

where each frame v_j represents the predicted visual state resulting from the execution of the corresponding action a_{j-1} . Consequently, the task is formulated as learning a joint generative policy parameterized by θ :

$$p_\theta(\mathbf{A}_t, \mathbf{V}_t \mid \mathcal{O}_t, I), \quad (4)$$

which captures both future action execution and the spatiotemporal evolution of the scene conditioned on the language instruction.

4.2 Data Preprocessing

Language Encoding. We employ a text encoder based on T5 with full attention to encode the navigation instruction I . T5 provides context-aware token representations that capture long-range dependencies and fine-grained semantic structure, which is crucial for grounding high-level language instructions into sequential decision-making. We set the maximum token length to $L_{txt} = 250$. The encoded instruction embeddings are denoted as:

$$\mathbf{E}_I = \{e_l \in \mathbb{R}^{D_{txt}}\}_{l=1}^{L_{txt}}, \quad (5)$$

where D_{txt} is the embedding dimension.

Video Encoding. The UAV navigates via discrete motion primitives, where each step induces large spatial displacements and drastic viewpoint changes. Given these rapid transitions, temporal down-sampling would discard critical spatial context required for state estimation. Therefore, we encode both historical observations and future frames on a per-frame basis to preserve full spatiotemporal fidelity. We define the full target video sequence $\mathcal{X}$ by concatenating the historical buffer $\mathcal{O}_t$ and the future prediction targets $\mathbf{V}_t$:

$$\mathcal{X}_t = \{x_i\}_{i=t-L_h+1}^{t+K}, \quad x_i = \begin{cases} o_i, & t - L_h + 1 \leq i \leq t \quad (\text{History}), \\ v_i, & t < i \leq t + K \quad (\text{Future}). \end{cases} \quad (6)$$

where o_i denotes historical observations and v_i denotes future frames. We adopt the VAE encoder $E(\cdot)$ from LTX-Video to map each frame x_i into a compressed latent space:

$$z_i = E(x_i), \quad z_i \in \mathbb{R}^{C \times h \times w}, \quad (7)$$

where C, h, w denote the channel, height, and width of the latent feature map, respectively. The resulting latent sequence is denoted as $\mathcal{Z}_t = \{z_i\}_{i=t-L_h+1}^{t+K}$. Our WorldFly model operates directly in this latent space to predict the future latent states. During inference, the pixel-space frames are reconstructed frame-wise using the VAE decoder: $\hat{x}_i = D(z_i)$. This VAE applies aggressive spatial compression while preserving spatial layout, enabling efficient transformer-based modeling.

4.3 Dual-Branch Coupled Architecture

We propose a Dual-Branch Coupled Architecture that explicitly separates world modeling and action modeling into two parallel branches: a *World Model Branch* and an *Action Expert Branch*. Each branch maintains independent parameters while interacting through periodically inserted dual-branch coupling layers. Let g_{θ_W} and g_{θ_A} denote the networks for the world model and action expert branches, parameterized by θ_W and θ_A , respectively.

Flow Matching with Aligned Noising. In the world-model branch, the historical video latents serve as clean conditioning inputs, while Gaussian noise is added only to the future latents. Specifically, for each training step we sample a shared timestep τ and construct noised future latents $\tilde{z}_i(\tau)$. Note that the historical latents $\{z_i\}_{i \leq t}$ remain clean and act as conditioning context. Similarly, for the action expert, we corrupt the future action chunk $\tilde{a}_j(\tau)$ using the same timestep. By sharing τ , we ensure that the signal-to-noise ratio is consistent across both modalities, preserving the temporal alignment between future dynamics and actions during training.

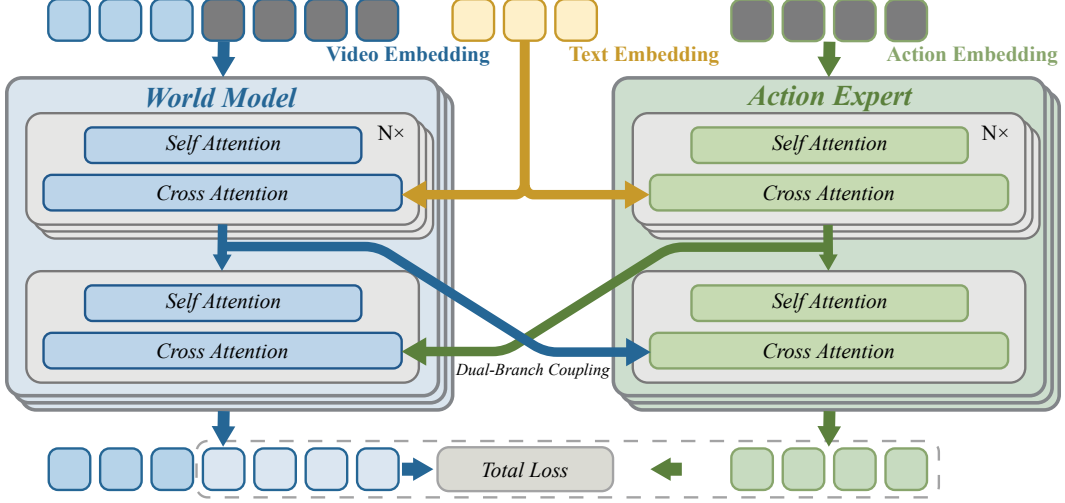


Figure 3: Dual-Branch Coupled Architecture. Historical observation frames and language instructions are provided as conditional inputs to a dual-branch flow-matching framework, enabling simultaneous future video latent prediction and action generation. To facilitate cross-modal information exchange, dual-branch coupling blocks are periodically inserted, where cross-attention allows the action expert to leverage imagined future dynamics while the world model is conditioned on planned trajectories.

Training Objective. The model is trained to predict the constant velocity vector field that flows from data to noise. Conditioned on the clean history $\mathcal{O}_t$ and instruction I , both branches are optimized via the flow-matching objective. Here, ϵ_i and η_j denote Gaussian noise sampled for the future latent z_i and action a_j , respectively. The world model loss over future latents is defined as:

$$\mathcal{L}_W = \mathbb{E}_{\tau, \epsilon, \mathcal{Z}} \left[\sum_{i=t+1}^{t+K} \|g_{\theta_W}(\tilde{z}_i(\tau), \tau \mid \mathcal{O}_t, I) - (\epsilon_i - z_i)\|_2^2 \right]. \quad (8)$$

Correspondingly, the action expert loss over the action chunk is:

$$\mathcal{L}_A = \mathbb{E}_{\tau, \eta, \mathbf{A}} \left[\sum_{j=t}^{t+K-1} \|g_{\theta_A}(\tilde{a}_j(\tau), \tau \mid \mathcal{O}_t, I) - (\eta_j - a_j)\|_2^2 \right]. \quad (9)$$

The overall training objective is a weighted sum:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_W + \lambda \mathcal{L}_A. \quad (10)$$

In our experiments, we set $\lambda = 1$ to balance the gradients from both branches.

Instruction Attention Block. Within each transformer block, the input latent feature H_{in} first undergoes self-attention, followed by cross-attention with the encoded instruction tokens $\mathbf{E}_I$:

$$H_{\text{out}} = \text{CrossAttn}(\text{SelfAttn}(H_{\text{in}}), \mathbf{E}_I). \quad (11)$$

This mechanism ensures that both visual generation and action planning are continuously guided by the high-level navigational goal.

Dual-Branch Coupling Block. To enable cross-modal interaction, a Dual-Branch Coupling Block is inserted every N layers. In this block, the latent features in each branch first undergo self-attention independently. Subsequently, cross-attention is performed, where the self-attended features of one branch act as queries and attend to the key-value representations provided by the other branch. Specifically, let H^W and H^A denote the features of the World Model and Action branches, respectively. The coupling is achieved via cross-attention:

$$H_{\text{out}}^W = \text{CrossAttn}(\text{SelfAttn}(H_{\text{in}}^W), H_{\text{in}}^A), \quad H_{\text{out}}^A = \text{CrossAttn}(\text{SelfAttn}(H_{\text{in}}^A), H_{\text{in}}^W). \quad (12)$$

This design allows the action policy to be informed by the imagined future context, while the world model’s generation is conditioned on the planned trajectory.

Table 1: Performance comparison across methods on TEST-EASY and TEST-HARD benchmarks.

Method	TEST-EASY			TEST-HARD		
	NE (m)	SR (%)	SPL (%)	NE (m)	SR (%)	SPL (%)
Random	46.49	2	1.79	35.30	0	0
Pi-0-UAV	34.07	29	27.08	41.35	10	9.43
OpenFly	14.69	72	58.55	35.32	16	14.92
WorldFly (ours)	7.92	87	73.25	31.08	31	27.86

Asymmetric Hidden Dimensions. To ensure efficient inference, the action expert branch utilizes a significantly smaller hidden dimension (D_{act}) compared to the world model branch (D_{world}). Our coupling mechanism naturally handles this dimensionality mismatch without requiring auxiliary projection layers. In the cross-attention mechanism $\text{CrossAttn}(Q, K, V)$, the query Q comes from the target branch, while keys K and values V come from the source branch. The learnable weight matrices W_K and W_V in the attention layer implicitly project the source features from their original dimension to the shared attention head dimension.

5 Experiments

5.1 Experimental Setup and Baselines

We conduct simulation-based evaluations on the **Urban Canyon Traversal Benchmark**. To comprehensively assess model performance, we evaluate on two distinct splits based on the novelty of the intersection layouts:

(1) **TEST-EASY**: 100 trajectory–instruction pairs sampled from intersections that appeared in the training set. This split tests the model’s ability to reproduce expert behaviors within familiar structural configurations. (2) **TEST-HARD**: 100 pairs sampled from novel intersections that were strictly excluded from the training data. This split evaluates the model’s zero-shot generalization capabilities to unseen spatial geometries and junction types.

Protocol. In all experiments, the UAV is initialized at the starting pose of the sampled trajectory. At each timestep, the model infers control actions conditioned on the natural language instruction and first-person visual observations. Crucially, to ensure a rigorous comparison, the action space is unified across all methods using the discrete OpenFly action primitives. OpenFly and Pi-0-UAV are both fully fine-tuned on the Urban Canyon Traversal benchmark under the same train/test split definition as WorldFly. Each episode is run for at most 100 steps. An episode is considered successful if the final position is within 12 meters of the target; otherwise it is treated as a failure when the maximum episode length is reached or when the agent predicts a stop action while still outside the success threshold.

Detailed hyperparameters and implementation specifics are provided in the Appendix A.

5.2 Experiment Results

We adopt the four standard evaluation metrics defined in OpenFly [5]: (i) **Success Rate (SR)**: The proportion of episodes where the UAV’s final position is within 12 meters of the target. (ii) **Navigation Error (NE)**: The average Euclidean distance between the final position and the target. (iii) **Success weighted by Path Length (SPL)**: A rigorous metric balancing success with path efficiency.

The quantitative results are presented in Table 1. WorldFly achieves state-of-the-art performance, consistently outperforming all baselines across both splits.

Generalization in Novel Environments. Most notably, in the **TEST-HARD** split (unseen intersections), WorldFly surpasses the strongest baseline (OpenFly) by a significant margin of 15% in Success Rate (31% vs. 16%) and 12.94% in SPL. While the absolute success rates decrease for all methods due to the difficulty of zero-shot transfer, WorldFly achieves a performance that is nearly double ($1.9\times$) that of OpenFly. Additionally, WorldFly reduces the Navigation Error (NE) by approximately

Table 2: Ablation study results on TEST-EASY and TEST-HARD benchmarks.

Method	TEST-EASY			TEST-HARD		
	NE (m)	SR (%)	SPL (%)	NE (m)	SR (%)	SPL (%)
WorldFly w/o DB	12.01	76	61.59	32.50	21	18.14
WorldFly	7.92	87	73.25	31.08	31	27.86

4.2 meters compared to OpenFly (31.08m vs. 35.32m). This result supports our hypothesis that joint future imagination and action reasoning improves robustness in unseen environments, rather than indicating that reactive policy learning alone is sufficient under large viewpoint transitions. While reactive baselines suffer from catastrophic performance drops (OpenFly’s SR drops from 72% to 16%), WorldFly demonstrates significantly stronger robustness.

Advantage of World Modeling. Compared to purely reactive VLA models (OpenFly and Pi-0-UAV), WorldFly exhibits drastically lower Navigation Error (NE) in familiar environments. In the TEST-EASY split, WorldFly reduces the navigation error by nearly $2\times$ compared to OpenFly (7.92m vs. 14.69m). We interpret this precision as evidence that the joint imagination-and-action design is beneficial for long-horizon reasoning. By explicitly imagining future states, WorldFly can anticipate obstacles and turnings in advance, leading to smoother and more proactive maneuvers. In contrast, reactive models often exhibit higher error rates due to missing turns due to a lack of foresight.

Effectiveness of Dual-Branch Coupling. Furthermore, WorldFly significantly outperforms Pi-0-UAV, which also employs a flow-matching backbone but relies on a standard single-stream VLA architecture. In unseen environments, WorldFly achieves a $3\times$ higher success rate than Pi-0-UAV (31% vs. 10%). This result is pivotal: it demonstrates that simply applying Flow Matching is insufficient for complex navigation. Our Dual-Branch Coupled Architecture—which explicitly disentangles world dynamics from action policies while ensuring interaction—is an important contributor to the final performance. The coupling mechanism allows the action expert to be guided by a dedicated, high-fidelity world simulator, resulting in higher SPL compared to the naive Pi-0-UAV implementation.

5.3 Ablation Study

We conduct ablation studies to evaluate the contribution of the Dual-Branch Coupled Architecture (DB), which enables bidirectional interaction between the world model branch and the action expert branch. To ensure a fair comparison, the variant WorldFly w/o DB was trained from scratch using the same dataset, optimization schedule, and hyperparameters as the full WorldFly model.

Table 2 presents the results. Comparing WorldFly with WorldFly w/o DB, the removal of bidirectional coupling leads to significant performance degradation, particularly on the more challenging TEST-HARD set. Specifically, while WorldFly w/o DB achieves a 76% Success Rate (SR) on TEST-EASY, its performance plummets to 21% on TEST-HARD (a 55-percentage-point decline). In contrast, the full WorldFly model experiences a more moderate decline from 87% to 31% (56 percentage points). This sharper performance decay in the absence of DB indicates that without the Dual-Branch Coupled Architecture, the model struggles to generalize and transfer learned visuomotor associations to difficult environments with larger distribution shifts. Overall, bidirectional coupling is crucial for capturing the intrinsic correlations between visual dynamics (world imagination) and the action space (policy reasoning), thereby enhancing robustness under challenging conditions.

6 Additional Analysis

6.1 Inference Latency and Memory.

We measure inference latency on a single NVIDIA A100 GPU using the same 50-step flow-matching denoising schedule as the main evaluation. For a fair measurement of model-side cost, we report the CUDA-event-timed forward latency of the action-only evaluation mode. WorldFly requires approximately 7.81 s per step, corresponding to a control frequency of approximately 0.5 Hz. The peak GPU memory is approximately 14.6 GB allocated.

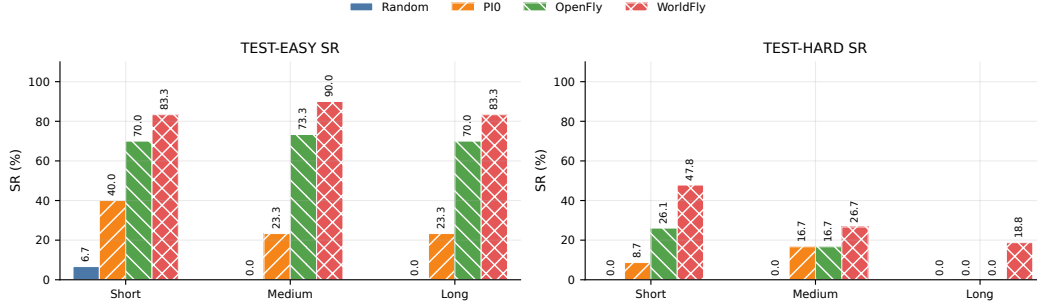


Figure 4: Success rates of different methods on short, medium, and long trajectory groups. Trajectories are divided into three groups according to their path lengths. The left figure shows the success rates on TEST-EASY, and the right figure shows the success rates on TEST-HARD.

Table 3: World-model future video prediction quality (trajectory-level aggregation). Higher is better for PSNR/SSIM, lower is better for LPIPS.

Split	PSNR	SSIM	LPIPS
TEST-EASY	14.33	0.342	0.432
TEST-HARD	13.17	0.248	0.518

6.2 Path-Length Analysis.

We further stratify the evaluation trajectories into short, medium, and long groups based on their path lengths within each split, with each group containing 30 navigation tasks. On TEST-EASY, WorldFly maintains robust performance across all three categories, achieving SR/SPL/NE of 83.3/74.7/7.57 for short, 90.0/73.0/6.76 for medium, and 83.3/68.7/11.10 for long routes. On TEST-HARD, performance exhibits a more pronounced degradation as route length increases: the SR/SPL/NE drops from 47.8/47.0/22.21 (short) to 26.7/23.9/34.66 (medium), and further to 18.8/11.5/35.82 (long). This trend aligns with the inherent challenges of long-horizon generalization. Notably, WorldFly consistently outperforms OpenFly across all categories in TEST-HARD. Even in the most challenging long-route group, where OpenFly fails to complete any tasks (0.0% SR), WorldFly retains an SR of 18.8%, demonstrating its superior capability in handling extended navigation trajectories through future-state imagination.

6.3 World Model Quality.

To directly quantify the fidelity of imagined future observations, we evaluate the predicted future frames against ground-truth future frames using standard video-prediction metrics. Ground-truth frames are resized to the model’s native operating resolution (224×224) before evaluation. As shown in Table 3, WorldFly achieves a trajectory-level PSNR/SSIM/LPIPS of 14.33/0.342/0.432 on TEST-EASY and 13.17/0.248/0.518 on TEST-HARD. The consistent degradation from EASY to HARD aligns with the larger distribution shift and more severe viewpoint variation in unseen intersections, providing direct evidence that the world model is non-trivial and produces structurally coherent future predictions even in the harder split.

7 Conclusion and Limitations

In this paper, we introduced **WorldFly**, a dual-branch coupled Vision-Language-Action model based on world model, designed to address embodied control for UAV in complex low-altitude urban environments. We proposed a novel architecture that explicitly decouples yet jointly optimizes future video generation and action planning through dense language interaction and periodic modality coupling. Experimental results on the Urban Canyon Traversal Benchmark demonstrate that WorldFly significantly outperforms strong baselines, including OpenFly and Pi-0-UAV. These findings

empirically validate that high-fidelity generative world modeling serves as a critical catalyst for enhancing the perception and decision-making capabilities of VLA models.

Limitations. The main limitation of our work is the unavoidable computational cost brought by future frame prediction. An important future research direction is to apply techniques such as model pruning and distillation to reduce computational costs.

References

- [1] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15791–15801, 2025.
- [2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.
- [3] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.
- [4] Xiaowei Chi, Kuangzhi Ge, Jiaming Liu, Siyuan Zhou, Peidong Jia, Zichen He, Yuzhen Liu, Tingguang Li, Lei Han, Sirui Han, Shanghang Zhang, and Yike Guo. Mind: Learning a dual-system world model for real-time planning and implicit risk analysis, 2025.
- [5] Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Peng'an Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, et al. Openfly: A comprehensive platform for aerial vision-language navigation. *arXiv preprint arXiv:2502.18041*, 2025.
- [6] Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.
- [7] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [8] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.
- [9] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.
- [10] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [11] Xiaofeng Mao, Shaoheng Lin, Zhen Li, Chuanhao Li, Wenshuo Peng, Tong He, Jiangmiao Pang, Mingmin Chi, Yu Qiao, and Kaipeng Zhang. Yume: An interactive world generation model. *arXiv preprint arXiv:2507.17744*, 2025.
- [12] Valerii Serpiva, Artem Lykov, Artyom Myshlyaev, Muhammad Haris Khan, Ali Alridha Abdulkarim, Oleg Sautenkov, and Dzmitry Tsetserukou. Racevla: Vla-based racing drone navigation with human-like behaviour. *arXiv preprint arXiv:2503.02572*, 2025.
- [13] Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025.
- [14] Xiangyu Wang, Donglin Yang, Yue Liao, Wenhao Zheng, Bin Dai, Hongsheng Li, Si Liu, et al. Uav-flow colosseo: A real-world benchmark for flying-on-a-word uav imitation learning. *arXiv preprint arXiv:2505.15725*, 2025.

A Model Configuration

A.1 Training Configuration

We evaluate three models, **WorldFly**, **OpenFly**, and **Pi-0-UAV**, under identical training settings to ensure a fair comparison. All three models are fully fine-tuned on the Urban Canyon Traversal benchmark using the same train/test split protocol.

We compare WorldFly against the following state-of-the-art baselines:

- **OpenFly**: A 7B parameter model fine-tuned from OpenVLA. To enhance spatiotemporal reasoning, it incorporates sparsely sampled historical visual frames. For a fair comparison with WorldFly, we provide the previous three frames as historical context.
- **Pi-0-UAV**: A flow-matching-based VLA model adapted from the π_0 architecture, utilizing PaliGemma as the language backbone. It generates continuous actions via multi-step denoising. Consistent with our method, it receives the previous three visual frames as input.

Table 4: Training configuration for WorldFly.

Parameter	Value
Hardware	$4 \times \text{A100 GPUs}$
Training Steps	40,000
Global Batch Size	128
Learning Rate	$3\text{e-}5$
LoRA Enabled	None

Table 5: Training configuration for OpenFly.

Parameter	Value
Hardware	$4 \times \text{A100 GPUs}$
Training Steps	40,000
Global Batch Size	128
Learning Rate	$5\text{e-}4$
LoRA Enabled	None

Table 6: Training configuration for Pi-0-UAV.

Parameter	Value
Hardware	$4 \times \text{A100 GPUs}$
Training Steps	40,000
Global Batch Size	128
Learning Rate	$5\text{e-}5$
LoRA Enabled	None

All models take three historical frames as input and predict four future frames during training. The action chunk size is set to 4 for **WorldFly** and **Pi-0-UAV**, while **OpenFly** predicts a single action per step (action chunk size = 1).

A.2 Action Primitives

We adopt the action primitives defined in **OpenFly** as the discrete action encoding scheme. Each action primitive represents an atomic low-level motion and is encoded as an 8-dimensional vector. The full set of action primitives is listed in Table 7.

Table 7: Definition of action primitives.

Index	Action Vector	Description
0	[1, 0, 0, 0, 0, 0, 0, 0]	Stop
1	[0, 3, 0, 0, 0, 0, 0, 0]	Move forward 3 m
2	[0, 0, 15, 0, 0, 0, 0, 0]	Turn left 30°
3	[0, 0, 0, 15, 0, 0, 0, 0]	Turn right 30°
4	[0, 0, 0, 0, 2, 0, 0, 0]	Move up 3 m
5	[0, 0, 0, 0, 0, 2, 0, 0]	Move down 3 m
6	[0, 0, 0, 0, 0, 0, 5, 0]	Move left 3 m
7	[0, 0, 0, 0, 0, 0, 0, 5]	Move right 3 m
8	[0, 6, 0, 0, 0, 0, 0, 0]	Move forward 6 m
9	[0, 9, 0, 0, 0, 0, 0, 0]	Move forward 9 m

A.3 Additional Experimental Details.

We include a **Random** baseline for reference. Specifically, **Random** denotes a policy that samples actions uniformly at random from the same discrete OpenFly action primitive set used by all methods. At each environment step, the agent independently draws one action from the **ten** available primitives and executes it, without conditioning on the instruction or visual observations.

WorldFly and **Pi-0-UAV** both adopt flow matching and perform the same 50-step denoising procedure to obtain the predicted action chunk; additionally, **WorldFly** predicts future video latents during the same denoising process. In contrast, **OpenFly** is an autoregressive model that predicts actions in an autoregressive manner. For all three methods, at each environment step we execute the *first* action of the predicted action chunk to interact with the environment.

At evaluation time, each episode is run for at most 100 steps. An episode is marked successful if the final UAV position is within 12 meters of the target. Otherwise, it is treated as a failure when the maximum episode length is reached or when the model predicts a stop action while still outside the success threshold. Navigation instructions are generated with an LLM-based instruction constructor built on the OpenFly toolchain using Qwen3-VL.

B Model Hyperparameters

As shown in Table 8, we report the relevant hyperparameters of our model. It is worth noting that the world model branch and the action expert branch adopt different hidden dimensions. All ablation experiments are trained using the same configuration.

Table 8: Model hyperparameters.

Parameter	Description	Value
<i>Data Processing</i>		
H, W	Input RGB frame resolution (height $\times$ width)	224×224
L_h	Length of historical observation buffer (frames)	3
K	Length of predicted future action/frame chunk	4
L_{txt}	Maximum token length for instruction encoding (T5)	250
D_{txt}	T5 text embedding dimension	4096
<i>WorldFly Architecture</i>		
D_{world}	Hidden dimension of World Model branch	2048
D_{act}	Hidden dimension of Action Expert branch	512
N_{layer}	Number of model layers	28
N	Frequency of Dual-Branch Coupling Block insertion	5
<i>WorldFly Inference</i>		
m	Size of Action Cache	4
α	Decay rate for cached action sequence voting weights	0.2

C Joint Video and Action Prediction Examples

As shown in Figure 5, we visualize a joint inference example of **WorldFly** for future video generation and action prediction. The input consists of 3 frames of historical observations and the natural language instruction; flow-matching denoising is performed for 50 steps, and 4 future frames are predicted. This case is selected from the **TEST-EASY** split to qualitatively demonstrate WorldFly’s capability of simultaneously performing future imagination and action execution. For visualization, the imagined future video is obtained by decoding the predicted future latents frame by frame using **LTX-Video VAE**.

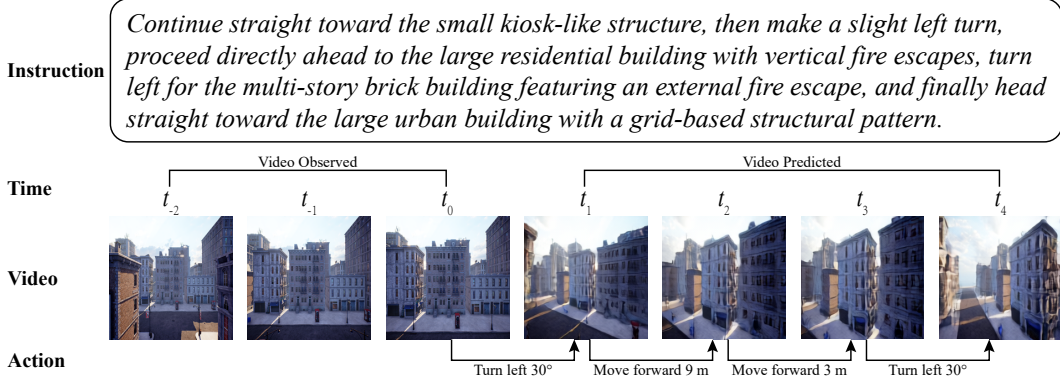


Figure 5: Joint video generation and action prediction with WorldFly.