Auxiliary future state generation framework using diffusion model

Yicheng Duan*
Duo Chen*
yxd245@case.edu
dxc830@case.edu
Computer and Data Sciences, School of Engineering
Case Western Reserve University
Cleveland, Ohio, USA

Abstract

Embodied AI aims to build agents that perceive, reason, and act within real-world environments. Vision-and-Language Navigation (VLN) tasks require agents to follow natural language instructions using visual input. While large vision-language models (VLMs) have advanced this field, they often suffer from inference-time hallucinations, leading to navigation errors. We propose a lightweight auxiliary framework that integrates a diffusion-based future state predictor to reassess and refine initial actions. This "future thought" mechanism enhances reliability with minimal computational overhead, offering a general plug-in to improve VLN performance.

CCS Concepts

• **Computing methodologies → Vision for robotics**.

Keywords

Computer Vision, State prediction, Vision Language Navigation

1 Project Overview

This project addresses a critical challenge in Embodied AI: enhancing the reliability of Vision-and-Language Navigation (VLN) agents that interpret natural language instructions to navigate complex visual environments. While large-scale vision-language models like ViLBERT [7] have advanced VLN capabilities, they are prone to inference-time hallucinations—generating plausible yet incorrect actions—due to their architectural complexity and training limitations [4]. To mitigate this, we propose an auxiliary framework that integrates a diffusion-based future state predictor, enabling agents to simulate and reassess potential outcomes before executing actions. This "future thought" mechanism enhances decision-making accuracy with minimal computational overhead. Evaluated on the Matterport3D (MP3D) dataset [2], our approach demonstrates the capability of improved navigation performance and robustness, aligning with the growing need for dependable embodied agents in real-world applications.

2 Problem Statement

Vision-and-Language Navigation (VLN) tasks require agents to interpret natural language instructions to navigate complex environments. However, these agents often struggle with ambiguous or erroneous instructions, leading to navigation errors. In the Room-to-Room (R2R) [2] navigation task, agents are required to follow natural language instructions to navigate indoor spaces. However, studies have indicated that VLMs can make navigation errors due to ambiguous landmark references. For instance, VLN-Trans introduces a translator module that converts original instructions into sub-instructions focusing on recognizable and distinctive landmarks based on the agent's visual abilities [14]. Similarly, Taioli et al. [13] propose a benchmark to detect and localize instruction errors, document instances where instructions like "turn right at the painting" result in errors when the agent confuses similar objects in the scene. These challenges underscore the necessity for systems that can interpret commands and predict real-world changes in the environment over time. Notably, the Room-Across-Room (RxR) dataset [5] highlights the complexities introduced by multilingual instructions and diverse environmental contexts.

To enhance environmental understanding, Wonderland generates high-quality 3-D scenes from a single image using a video diffusion model, though its lack of publicly available code limits reproducibility [6]. DIAMOND employs diffusion models for world modeling, preserving visual details crucial for agent performance, but requires scene-specific training, hindering scalability [1]. Additionally, integrating visual imaginations generated from text-to-image models has been shown to improve VLN agent performance by providing visual cues corresponding to described landmarks [8]. Despite these advancements, there remains a need for lightweight frameworks that can predict future environmental states based on inferred actions, allowing agents to reassess and refine their decisions in dynamic settings.

3 Methodology

An overview of the architecture is shown in Figure 1. Our proposed framework includes two primary stages:

Stage 1 (3D Reconstruction and Action-driven Navigation): Initially, given a text prompt and an initial real image captured at state $t = 0$, the system employs a single-shot 3D reconstruction module (Unik3D) to generate a coherent 3D representation of the environment. We utilize Habitat-Sim with the corresponding Matterport3D (MP3D) dataset as our environmental simulation platform. Concurrently, a vision-language model (VLM) agent interprets the provided textual instructions and visual context to infer a series of actionable instructions (action chunks). These inferred actions are subsequently executed within the reconstructed 3D environment, enabling navigation through virtual space to a new viewpoint. The final component of this stage involves viewpoint

*Both authors contributed equally to this work.

clipping, where the viewpoint for the subsequent state ($t = 1$) is precisely determined based on the executed navigation actions.

Stage 2 (Synthetic Image Generation via Diffusion): In the second stage, the viewpoint obtained from Stage 1 serves as the input for a diffusion-based synthesis process, specifically leveraging the RealVisXL_V4.0 inpainting model. This inpainting technique operates using an RGB image from the selected viewpoint along with a corresponding mask to define regions requiring synthesis or correction. The model generates a synthetic image at state $t = 1$ that is both realistic and contextually coherent with respect to the agent's predicted viewpoint. The resultant synthesized image is then directly compared with the real captured image at state $t = 1$, enabling an evaluation of the framework's effectiveness in simulating and accurately predicting future visual states.

We also experimented with **Janus-Pro** [3], an autoregressive framework that unifies multimodal understanding and generation, to synthesize novel viewpoints. However, the generated outputs were often inconsistent with the original inputs in terms of geometry and appearance, leading us to abandon this direction for the current project. A comparable alternative, **NVIDIA's COSMO**, which is trained on over 20 million hours of video data, has demonstrated impressive results in similar tasks. Nevertheless, its computational demands far exceed the constraints of our setup.

We use the Fréchet Inception Distance (FID) score to quantitatively compare the images generated by our framework against the corresponding real images. This metric evaluates both the structural coherence and overall realism of the generated images by measuring the statistical distance between feature representations extracted from a pretrained Inception network. In the context of vision-language model (VLM) based navigation, such evaluation is crucial, as VLMs are sensitive to the photorealistic fidelity and semantic consistency of input observations.

Our results yield an FID score of 105.41, indicating a significant divergence from ground truth images. This discrepancy is likely attributable to the 30-degree viewpoint shift, which removes substantial contextual information, making it challenging for the diffusion model to reconstruct the missing details accurately. Nevertheless, as illustrated in Figure 2, our framework still produces several qualitatively promising outputs, suggesting potential for further refinement and adaptation.

4 Tools and Technologies

For this project, we primarily utilized Python due to its extensive libraries and frameworks tailored for artificial intelligence and computer vision tasks. The core simulation environment leveraged was **Habitat-Sim** [11, 12, 10], chosen for its high-performance rendering capabilities and seamless integration with 3D datasets, particularly **Matterport3D (MP3D)** [2], which provides rich, photorealistic indoor scenes well-suited for navigation tasks.

To interact with our 3D-reconstructed point clouds, we used **Open3D** as a visualization and manipulation tool, allowing us to navigate within the reconstructed scenes and extract viewpoint images efficiently. Additionally, we employed **PyTorch** and **Hugging Face Diffusers** to implement diffusion models, chosen for their strong support of modern deep learning architectures and ease of integration into existing pipelines.

UNik3D: In the 3D reconstruction phase, we leverage Unik3D [9] to infer a complete three-dimensional representation from a single RGB image. This process yields three complementary outputs: (1) Point cloud (.ply file): A collection of discrete 3D points sampled from the surfaces visible in the scene; each point carries spatial coordinates (x, y, z) that together form a sparse but accurate model of object geometry. (2) Depth map: A dense per-pixel image in which each value encodes the distance from the camera to the corresponding surface point along the viewing ray; depth maps are essential for understanding relative object placement and for driving further geometry refinement. (3) Ray map: A per-pixel directional field where each vector describes the normalized direction of the camera ray passing through that pixel; ray maps capture the angular relationship between pixels and the camera center, ensuring that the reconstructed geometry aligns consistently with the original camera pose. By combining these outputs—a spatially distributed point cloud, a pixel-wise depth representation, and the underlying ray directions—Unik3D provides a rich, multi-view-aware reconstruction that can be further processed for visualization, mesh generation, or downstream tasks such as navigation and scene understanding.

Unik3D (shown in figure 3) uses two lightweight transformers: the angular module predicts per-pixel ray directions via spherical harmonic coefficients, and the radial module fuses those directions with image features to regress continuous depth. By unprojecting each pixel along its predicted ray and depth, it produces a sparse point cloud (.ply) plus dense depth and ray maps.

RealVisXL_V4.0 Inpainting Model: We employed the **RealVisXL_V4.0** inpainting model, a diffusion-based architecture designed for high-fidelity image synthesis and restoration. This model operates by taking an input RGB image along with a binary mask that defines the region to be inpainted. The diffusion process then fills in the masked area with semantically consistent and visually realistic content, conditioned on the surrounding context. In our framework, the masked regions are generated based on the viewpoint transformation from the agent's predicted state, allowing the model to simulate plausible future observations. RealVisXL_V4.0 was chosen for its superior performance in generating realistic textures, maintaining relative geometric alignment, and preserving scene semantics, outperforming traditional GAN-based methods and lower-resolution diffusion alternatives.

Image Sources and Selection Criteria: All images utilized in this project originated from the Matterport3D (MP3D) dataset, a publicly available dataset containing high-resolution RGB images, depth maps, and reconstructed 3-D meshes collected from diverse real-world indoor environments. MP3D was specifically chosen due to its comprehensiveness, realism, and widespread adoption within the embodied AI research community, thereby facilitating robust benchmarking and comparative analysis. For our image selection process, we sampled a single reference point within each scene provided by MP3D. From each sampled reference point, we defined an initial base state and subsequently generated four corresponding action-driven result images, forming our real-image dataset for validation and comparative evaluation.

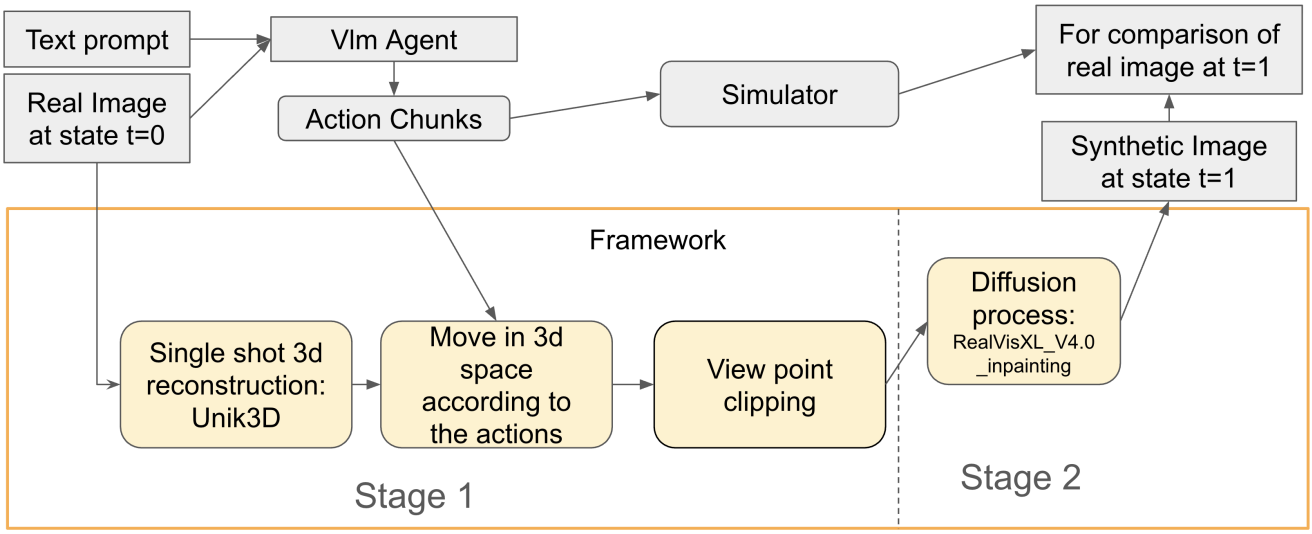

Figure 1: Overview of the proposed framework. Stage 1 involves single-shot 3-D reconstruction (Unik3D), action-driven 3-D navigation, and viewpoint clipping. Stage 2 synthesizes images using a diffusion process (RealVisXL_V4.0_inpainting)

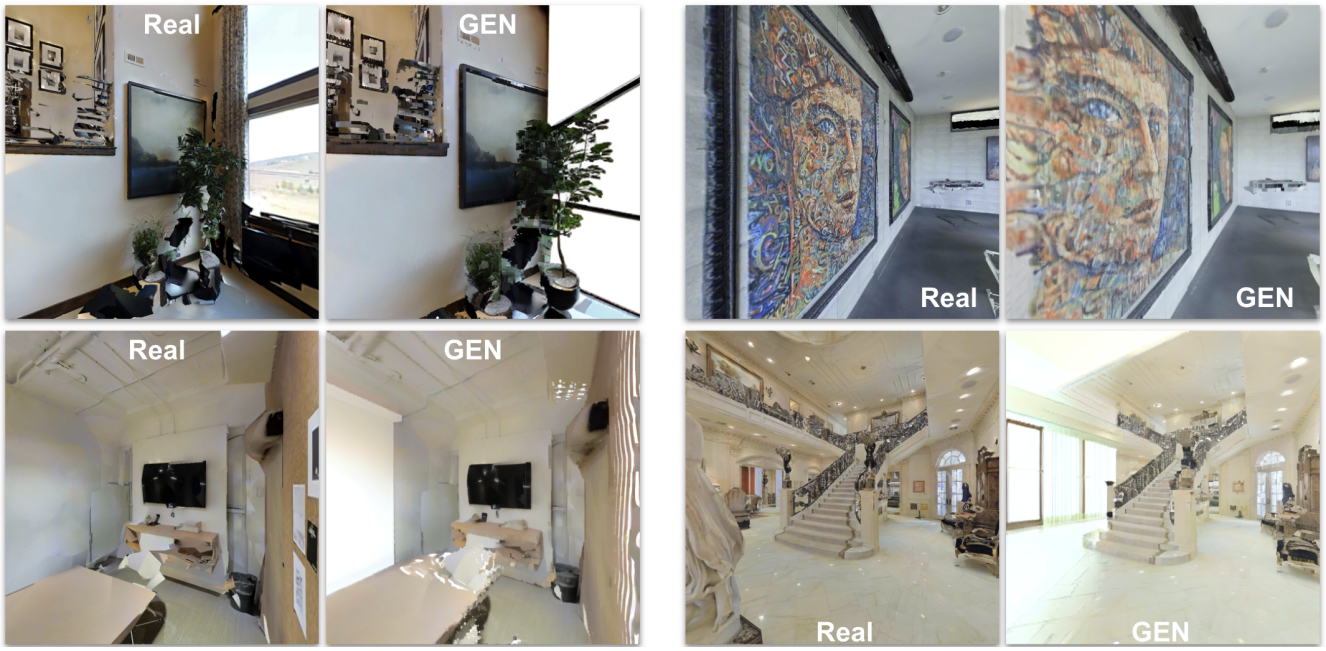

Figure 2: Qualitative comparison between real and generated images. Each pair of columns shows a real image (left) and its corresponding generated image (right) produced by our framework. While some structural discrepancies exist due to viewpoint shifts, the generated outputs often preserve key scene semantics and textures.

5 Timeline

- **[1] 3D Reconstruction & Initial Fine-Tuning**
  Use Unik3D to predict the scene's 3D representation and reconstruct the mesh in Open3D. ***Completed: April 1, 2025***
- **[2] Camera Mounting & View Acquisition**
  Moving the camera according to specifications and capture clippings via Open3D to get the new view image. ***Completed: April 10, 2025***

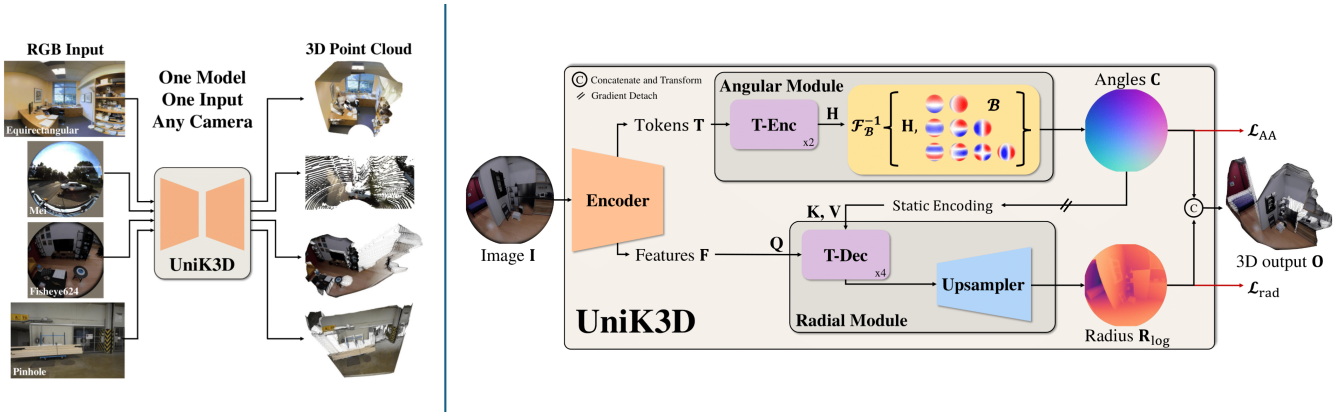

Figure 3: Unik3D framework. Given a single RGB image, a lightweight CNN backbone first extracts per-pixel feature tokens. The angular module augments these tokens with learned directional embeddings and predicts spherical-harmonic coefficients encoding each camera ray's orientation. These coefficients are then fused via cross-attention into the radial module, which regresses a continuous depth value for every pixel. Finally, each pixel is unprojected along its predicted ray and depth to produce a sparse point cloud (.ply), alongside dense depth and ray maps.

- **[3] Diffusion-Based Novel View Inpainting**
  Generate the mask from the new view image and inpaint occluded regions using the RealVisXL V4.0 model. ***Completed: April 20, 2025***
- **[4] VLM Fine-Tuning & Pipeline Integration**
  Attempt to fine-tune the Janus-Pro vision–language model and build an end-to-end workflow (approach deemed a dead end). ***Completed: April 27, 2025***
- **[5] Evaluation & Final Reporting**
  Run performance experiments, analyze results, and draft the final project report. ***Completed: May 3, 2025***

6 Outcome

Throughout the course of this project, we aimed to achieve the following specific objectives:

- **Objective 1:** Investigate and implement a 3D reconstruction model for our framework.
  We successfully integrated **Unik3D** as the single-shot 3D reconstruction module, enabling efficient conversion of 2D images into structured 3D scenes suitable for downstream 3D navigation and viewpoint synthesis.
- **Objective 2:** Investigate and implement a diffusion model for our framework.
  After comparative exploration, we selected **RealVisXL_V4.0** for its strong inpainting capability and incorporated it to generate high-fidelity viewpoint predictions.
- **Objective 3:** Successfully implement our proposed framework.
  The overall pipeline—spanning from 3D reconstruction, 3D navigation, to synthetic image generation—was successfully implemented and tested end-to-end.
- **Objective 4:** Evaluate our proposed framework on the Room-to-Room dataset.
  We used samples from the Room-to-Room (R2R) benchmark via the MP3D environment to evaluate the visual accuracy of predicted images. The framework's generated views were quantitatively assessed using the FID metric and qualitatively analyzed through side-by-side comparisons.
- **Objective 5:** Investigate and fine-tune the Janus-Pro model on the Room-to-Room dataset.
  Janus-Pro was explored for potential use in novel viewpoint synthesis. However, the outputs were inconsistent with ground-truth imagery, leading us to exclude this method from the final implementation.

As the project progressed, we also identified a new potential application. Inspired by the **TesserAct** framework [15], which proposes 4D generation by incorporating temporal dimensions, our framework could be adapted to synthesize future video chunks across multiple timesteps. This would enable predictive state generation not just for single images, but for temporally coherent sequences, expanding its utility in long-horizon planning and video-based navigation tasks.

7 Further Work

In future work, we plan to optimize **UniK3D** by fine-tuning its camera parameters—specifically focal length, principal point, and pose priors—to enhance both the accuracy and density of the reconstructed point clouds. This refinement is expected to improve the spatial realism and continuity of our 3D environments.

Concurrently, we will adapt our diffusion model to the indoor-scene domain by conducting domain-specific fine-tuning. This process aims to better capture the unique textures, lighting conditions, and structural characteristics inherent to indoor environments, enabling the generation of higher-fidelity synthetic views that integrate more effectively with the reconstructed 3D scenes.

Additionally, inspired by the **Tesseract** framework [15], we plan to extend our current system to support *future state prediction over time*. This would enable the generation of temporally consistent

video sequences, thereby broadening the applicability of our framework to tasks such as long-horizon planning, video prediction, and embodied agent simulation.

8 Work Division

Our team's work is split between two complementary roles: Duo Chen focuses on early model research and the core 3D reconstruction pipeline—using Unik3D for scene prediction and Open3D to manage camera movements and capture clips—while Yicheng Duan leads system architecture design, implements novel-view image generation with RealVisXL V4.0, and fine-tunes the Janus-Pro vision–language model. At the same time, we synchronize progress daily via Email and WeChat to ensure seamless collaboration.

Acknowledgments

The code and resources for this work are publicly available at: https://github.com/YichengDuan/axground3D

References

[1] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. 2024. Diffusion for world modeling: visual details matter in atari. In *Thirty-eighth Conference on Neural Information Processing Systems*. https://arxiv.org/abs/2405.12399.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*.

[3] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-pro: unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

[4] Meera Hahn, Amit Raj, and James M. Rehg. 2023. Which way is 'right'?: uncovering limitations of vision-and-language navigation model. (2023). https://arxiv.org/abs/2312.00151 arXiv: 2312.00151 [cs.CV].

[5] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4392–4412.

[6] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos Plataniotis, Sergey Tulyakov, and Jian Ren. 2024. Wonderland: navigating 3d scenes from a single image. *arXiv preprint arXiv:2412.12091*.

[7] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265. http://arxiv.org/abs/1908.02265 arXiv: 1908.02265.

[8] Akhil Perincherry, Jacob Krantz, and Stefan Lee. 2025. Do visual imaginations improve vision-and-language navigation agents? (2025). https://arxiv.org/abs/2503.16394 arXiv: 2503.16394 [cs.CV].

[9] Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. 2025. Unik3d: universal camera monocular 3d estimation. (2025). https://arxiv.org/abs/2503.16591 arXiv: 2503.16591 [cs.CV].

[10] Xavi Puig et al. 2023. Habitat 3.0: a co-habitat for humans, avatars and robots. (2023).

[11] Manolis Savva et al. 2019. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[12] Andrew Szot et al. 2021. Habitat 2.0: training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[13] Francesco Taioli, Stefano Rosa, Alberto Castellini, Lorenzo Natale, Alessio Del Bue, Alessandro Farinelli, Marco Cristani, and Yiming Wang. 2024. Mind the error! detection and localization of instruction errors in vision-and-language navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12993–13000. doi:10.1109/IROS58592.2024.10801822.

[14] Yue Zhang and Parisa Kordjamshidi. 2023. VLN-trans: translator for the vision and language navigation agent. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, (Eds.) Association for Computational Linguistics, Toronto, Canada, (July 2023), 13219–13233. doi:10.18653/v1/2023.acl-long.737.

[15] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. 2025. Tesseract: learning 4d embodied world models. (2025). https://arxiv.org/abs/2504.20995 arXiv: 2504.20995 [cs.CV].