I-Perceive: A Foundation Model for Vision-Language Active Perception

Yongxi Huang[1] **Zhuohang Wang**[2,3] **Wenjing Tang**[1] **Xinyu He**[2]
Cewu Lu[1,2]* **Panpan Cai**[1,2]*

[1]Shanghai Jiao Tong University [2]Shanghai Innovation Institute [3]Beihang University

Abstract

Active perception—the ability of a robot to proactively select viewpoints to acquire task-relevant information—is essential for robust operation in real-world environments. However, existing approaches are typically limited to fixed objectives or constrained settings, and struggle to generalize to open-ended perception intents specified in natural language. We propose I-Perceive, a foundation model for language-conditioned active perception in large-scale indoor environments. Given a query image, a set of context images, and a natural language instruction, I-Perceive predicts a 6D camera pose that fulfills the specified perception intent. The model integrates a vision-language pathway for semantic grounding with a geometric reasoning pathway for multi-view 3D understanding, connected via multi-layer semantic fusion to enable language-conditioned geometric reasoning. To support scalable training, we construct a large-scale dataset of language-viewpoint pairs from both real-world scene-scanning data and simulated environments using an automated pipeline. Extensive experiments demonstrate that I-Perceive significantly outperforms strong baselines on prediction accuracy, viewpoint feasibility, and instructions alignment. The model exhibits strong zero-shot generalization to unseen scenes and instructions, and enables closed-loop active perception, progressively refining viewpoints over sequential interactions. See our project page: `https://roboticsjtu.github.io/I-Perceive-Page/`.

1 Introduction

Active perception refers to a robot's ability to proactively control its camera pose in order to acquire task-relevant information for downstream decision-making and execution [1, 4, 5, 43]. Existing approaches, however, are largely fragmented. On the one hand, many methods are designed for fixed downstream tasks—such as object detection, pose estimation, or scene reconstruction [9, 27, 30, 40]—where the perception objective is hard-coded into task-specific optimization losses. As a result, models developed for different tasks are isolated, difficult to integrate, and lack generality. On the other hand, a large body of work assumes highly constrained environments, typically limited to table-top or small-scale scenes [19, 25, 47, 48], which restricts their applicability to real-world indoor settings.

In this work, we aim to provide a foundation model for active perception that overcomes these limitations, allowing (1) generalizing to open-ended perception intents specified by natural language instructions, and (2) scaling to large-scale indoor environments suitable for mobile manipulators. Achieving this goal poses several fundamental challenges. First, the model must infer perception intent from natural language instructions, which are inherently diverse and underspecified. Second, it

*Corresponding authors.

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

must recover the geometry of large scenes from sparse visual observations, bridging the gap from image space (2D) to geometry space (3D). Third, it must ground language-specified intents into the visual and geometric representations of the scene, requiring alignment across language, image, and geometry spaces. Finally, the model must transform the geometry of task-relevant regions or objects into actionable camera poses, mapping 3D scene understanding to 6D camera poses.

Recent advances provide partial solutions to these challenges. Vision-Language Models (VLMs) [3, 18, 33] excel at aligning language with visual observations, while geometric foundation models (e.g., VGGT) [23, 36, 37] enable multi-view 3D reasoning and camera pose estimation. However, these capabilities remain largely *disconnected*: existing systems lack a principled way to translate open-ended language instructions into viewpoint decisions, leaving a critical gap between semantic intent understanding and geometric action.

To bridge this gap, we propose I-Perceive, a foundation model for language-conditioned active perception, that enables end-to-end viewpoint decision-making in large-scale indoor environments. Given a query image, a set of context images, and a natural language instruction, I-Perceive predicts a 6D camera pose that fulfills the specified perception intent. At a high level, the model integrates two complementary pathways: a *vision-language pathway* that extracts semantic intent from multimodal inputs, and a *geometric reasoning pathway* that models multi-view spatial structure and camera relationships. The key challenge lies in tightly coupling these pathways to enable *language-conditioned geometric reasoning under partial observability*.

To this end, we introduce a unified architecture based on *cross-pathway fusion*. We represent viewpoint prediction as reasoning over a latent *target frame*, whose learnable tokens participate in joint multi-view attention with observed frames, enabling the model to infer unseen viewpoints under partial observability. Semantic information is injected via multi-layer cross-attention through dedicated *semantic tokens*, allowing language-grounded features from the vision-language pathway to progressively modulate geometric reasoning across views. This yields a unified model for viewpoint decision-making that jointly reasons about *what to observe* and *where to observe it* in 3D space.

To support scalable training, we construct a large-scale dataset of language-conditioned active perception tasks from both real-world scene-scanning datasets [10, 22] and simulated environments [21]. This results in 162K samples from real-world data and 70K from simulation, enabling both robust generalization and strong instruction-following capability. All tasks are annotated with high-quality camera viewpoints using a scalable and fully automated data generation pipeline.

In our experiments, we evaluate I-Perceive on both in-distribution simulation tasks and out-of-distribution environments. Results show that I-Perceive significantly outperforms strong baselines, including vision-language navigation methods [8], and next-best-view strategies [9, 27], and VLM prompting, in terms of viewpoint accuracy, feasibility, and task alignment. Moreover, I-Perceive exhibits strong zero-shot generalization to novel scenes and previously unseen perception intents, while robustly handling complex language instructions, rich visual contexts, and partial observability. The model naturally supports a variable number of context images, with performance improving as additional context is incorporated. These capabilities enable I-Perceive to perform sequential active perception: when executed multiple times in a closed-loop manner, it progressively refines camera viewpoints, producing progressively-refined views over successive perception steps.

2 Related Work

2.1 Active Perception

Active perception refers to an agent's ability to actively control its sensors to acquire task-relevant information. As formulated by Bajcsy [4, 5], an active perceiver must determine *what* to sense, and consequently decide *how* and *where* to perceive. This paradigm has motivated extensive research across robotics and computer vision [1, 43].

Reconstruction-Driven Approaches. A major line of active perception research focuses on 3D reconstruction, where the objective is to maximize geometric completeness. Classical Next-Best-View (NBV) methods adopt generate-and-test strategies, selecting viewpoints that reveal the largest amount of unknown geometry [9, 38]. Subsequent works introduced information-theoretic criteria to quantify viewpoint utility [11, 15], while learning-based approaches directly predict informative viewpoints from partial observations [27, 28]. Recent work further formulates reconstruction-oriented

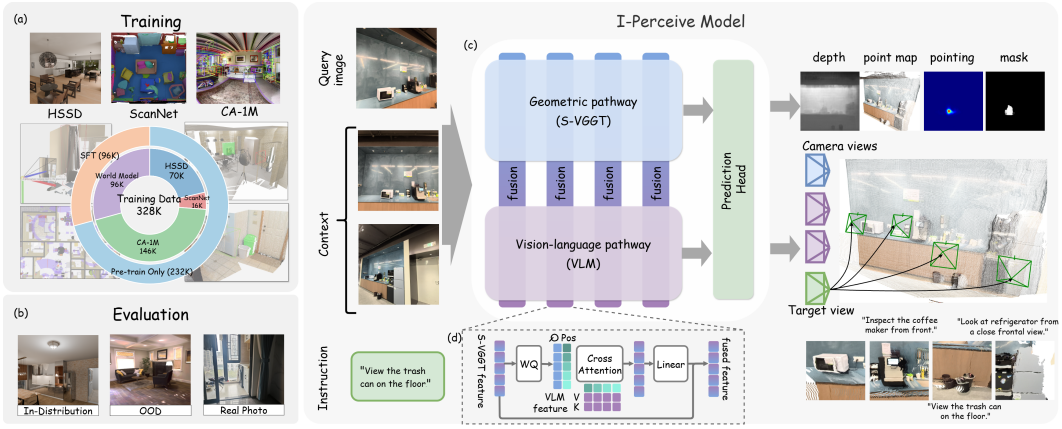


Figure 1: **I-Perceive** is a foundation model for vision-language active perception. Given context images and a natural language instruction, the model predicts a target camera pose that fulfills the observation intent. By fusing VLM semantic features into a geometric backbone, I-Perceive demonstrates strong zero-shot generalization to unseen real-world environments.

active perception as informative path planning under partial observability [29]. However, these methods primarily optimize geometric coverage and lack the semantic understanding required to interpret diverse user intents.

Perception-Driven Approaches. Active perception has also been studied for object recognition and pose estimation tasks. Early work integrated sensor planning with object hypothesis generation [14], while deep learning approaches such as 3D ShapeNets [40] select viewpoints that minimize classification uncertainty. Other methods leverage multi-view reasoning for recognition [17] and 6D pose estimation in cluttered scenes [16, 30]. Recent work [25, 48] further integrates active perception with manipulation by jointly learning camera motion and manipulation policies, but remains largely restricted to table-top environments. For large-scale environments, Vision-Language Navigation (VLN) [2, 12, 39] is also closely related, where agents navigate toward language-specified goals. Recent VLN methods leverage foundation models for semantic grounding and reasoning [20, 26, 42, 44]. However, although VLN implicitly involves viewpoint selection, its objective remains navigation-oriented, where views are primarily chosen based on exploration or proximity to goal locations rather than perception quality.

Positioning of Our Work. Existing active perception methods are typically designed for fixed objectives (*e.g.*, reconstruction, recognition, or navigation) with task-specific optimization criteria. In contrast, our work studies *instruction-driven* active perception that generalizes across diverse perception intents specified in natural language. This requires jointly reasoning about semantic intent, visual observations, and 3D geometry under partial observability, a capability that remains largely absent in prior work.

2.2 Vision-Language and Geometry Foundation Models

VLMs and Spatial Understanding. Vision-Language Models (VLMs) have achieved remarkable success in semantic understanding and reasoning across vision and language modalities [3, 18, 33]. However, recent studies reveal that VLMs struggle with spatial reasoning and 3D geometric understanding [41, 45]. Some works apply VLMs directly to active perception [19, 31, 47], but these typically operate in constrained settings (*e.g.*, table-top scenes) with limited action spaces (*e.g.*, image cropping or zooming), lacking fundamental 3D scene understanding.

Geometric Foundation Models. Recently, transformer based geometric foundation models [23, 36, 37] have shown impressive capabilities in recovering 3D structure and camera poses from sparse views. By leveraging large-scale training data and powerful attention mechanisms, VGGT [36] achieves robust spatial reasoning across diverse scenes and viewpoints, enabling effective mapping from discrete images to holistic 3D scene understanding and camera pose inference. However, while VGGT-like models are strong at estimating camera poses from given observations, they are not designed to perform viewpoint decision-making under partial observability, and do not accept language input.

Bridging Semantics and Geometry. Recent work has explored integrating VLM semantics with geometric reasoning. One line of research enhances VLMs' spatial understanding through language-

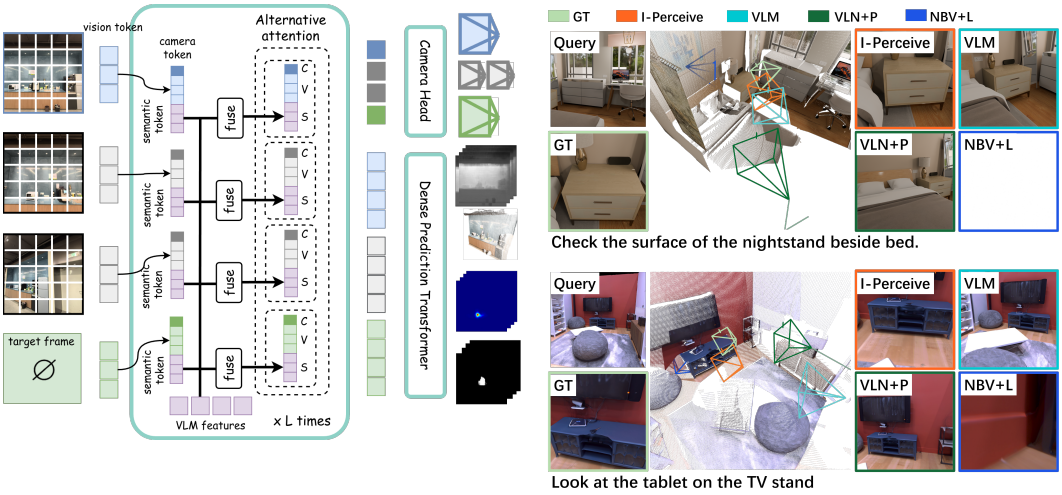


Figure 2: **Left:** A zoom-in view of I-Perceive's geometric-reasoning pathway. **Right:** Qualitative results of I-Perceive.

level supervision from geometry models [6, 7, 46], while another incorporates explicit geometric features from geometric foundation models into VLMs to improve downstream tasks such as spatial QA and manipulation action generation [24]. These approaches primarily leverages geometry to enhance VLM-centric reasoning, demonstrating the feasibility of joint semantic-spatial reasoning.

Positioning of Our Work. I-Perceive performs the integration in an *geometry-centric* manner: it injects VLM-derived semantic features into a geometric reasoning backbone as conditions. The core insight is that viewpoint decision-making is fundamentally a geometric problem, requiring reasoning in 3D space. Images and language only specify the semantic and task contexts.

3 Overview

I-Perceive formulates active perception decision-making as a mapping from multi-view scene observations and language tasks to executable camera poses. The input include a sequence of RGB images of the environment and a natural language instruction specifying the perceptual intent. The first image frame is the referred to as the *query frame*, representing the current view of the robot; the subsequent frames serve as *context frames*, providing auxiliary information of the global scene. A model should output a target camera pose that is both physically feasible (executable by a physical robot) and accurately fulfills the intent of language task.

The core challenge lies in jointly reasoning over language, visual observations, and 3D scene geometry under partial observability. To address this, I-Perceive integrates two complementary components: a vision-language pathway for extracting semantic intent from multimodal inputs, and a geometric reasoning pathway for modeling 3D structure and camera pose relationships across views. Crucially, semantic features are progressively injected into the geometric pathway via multi-layer cross-attention, enabling *language-conditioned geometric reasoning*. I-Perceive is trained on a large-scale dataset of active perception tasks constructed from both real-world scene-scanning data and simulated environments, enabling both robust generalization and strong instruction-following performance. At inference time, the model supports both single-step prediction and closed-loop execution, where newly observed views are iteratively incorporated as context to refine future predictions. Fig. 1 provides an overview of the method. The accompanying video provides a detailed narrative.

4 Architecture of the I-Perceive Model

I-Perceive maps visual context and language instructions to a target camera pose in an end-to-end manner, performing joint semantic and geometric reasoning using two tightly-integrated pathways. The detailed architecture is as follows.

Input and Output. The model takes as input N RGB frames $\{I_i\}_{i=0}^{N-1}$ and a natural language instruction ℓ. The goal is to predict the target camera parameters $\mathbf{g}_N$ that fulfill the instruction, defined w.r.t. the coordinate system rooted at the camera pose of I_0. Each camera is parameterized as $\mathbf{g}_i = [\mathbf{q}_i, \mathbf{t}_i, \mathbf{f}_i] \in \mathbb{R}^9$, representing the rotation quaternion, translation vector, and field of view, respectively. In addition to camera pose prediction, the model optionally produces dense geometric outputs (depth and point maps) and semantic masks (2D pointing and target object masks), comprising auxiliary tasks to facilitate spatial reasoning and semantic grounding during training.

Vision-Language Pathway. We use a pretrained VLM (Qwen3-VL-4B) backbone to encode multimodal semantic information from images and language. To extract semantic representations, instead of using the final-layer features, we extract a rich set of intermediate key-value (KV) features from selected transformer layers $\mathcal{L} = \{l_j\}_{j=1}^{N_Q}$:

$$K^{(l_j)}, V^{(l_j)} = \text{ExtractKV}(\text{VLM}, l_j, \mathcal{I}, \ell) \tag{1}$$

These multi-layer representations reflect the progressive semantic reasoning structure, allowing smooth integration to progressive geometric reasoning.

Geometric Reasoning Pathway (Fig. 2). We adopt a multi-view transformer architecture for geometric reasoning, inspired by a prior geometric foundation model, VGGT [36], that recover 3D structure and camera poses from visual observations. However, unlike VGGT, which focus on reconstructing geometry and estimating poses of observed views, our goal is to infer unseen, task-driven viewpoints conditioned on language and partial observations.

Similar to [36], the geometric backbone of I-Perceive maps each input frame as a set of tokens encoding visual appearance (*vision tokens,* v) and camera state (*camera tokens*, c), using a stack of attention layers that alternate intra- and inter-frame attentions, to perform joint reasoning across all frames and model their spatial relationships. This enables the model to aggregate multi-view information and build an implicit 3D understanding of the scene.

To enable language-conditioned active perception, we uniquely introduce two key design elements:

(1) Target-frame tokens. We append a latent target frame I_T with learnable tokens to represent an unseen viewpoint. Its vision and camera tokens participate in the same alternating attention pathway with the start and context frames, allowing to aggregate information from the observed frames. This effectively casts viewpoint decision-making as extrapolation from seen to unseen viewpoints.

(2) Semantic tokens. We introduce *semantic tokens* to each frame, denoted as s, that serve as carriers of language-grounded information. In intra-frame attention, semantic tokens interact with both vision tokens (v) and VLM features via Eq. (2), enabling local grounding of semantics in visual content. These fused representations are then propagated globally via inter-frame attention.

Cross-Pathway Fusion. Language-conditioned information are repeatedly injected into semantic tokens, via cross-attention:

$$s_i' = s_i + \gamma \cdot \text{Proj}\left(\text{Attn}\left(W_Q s_i, K^{(l)}, V^{(l)}\right)\right), \forall l \in \mathcal{L}. \tag{2}$$

Here, $\mathcal{L}$ denote layers where we extract VLM features, $K^{(l)}$ and $V^{(l)}$ represent the extracted features, respectively. Compared to shallow or late fusion strategies, injecting semantic information at multiple layers enables deep integration of language and geometry, allowing semantic intent to progressively modulate spatial reasoning throughout the network. See more implementation details in Appendix B.1.

Prediction Heads. The final token representations are decoded to predict camera poses, dense, and semantic output for both observed frames and the target frame. Concretely, camera tokens at selected layers are aggregated and passed through a regression head to estimate camera parameters of all frames; vision tokens at selected layers are aggregated and fed to transformer-based dense prediction heads for depth map and point map generation, in order to enhance 3D structural reasoning. Vision tokens are also fed to semantic grounding heads, to generate 2D pointing heatmaps and target-object masks, in order to encourage semantic understanding. At inference time, only the camera pose prediction head is required; other predictions are optional.

5 Training I-Perceive

We train I-Perceive using supervised learning. Importantly, there exists no public dataset that directly supports language-conditioned active perception learning, which requires pairing multi-view observations and language tasks with camera-view annotations. To address this, we construct a large-scale I-Perceive dataset of vision-language-viewpoint tuples, through an automated data generation and labeling pipeline, which transforms indoor scene datasets into active perception tasks with minimal human intervention. Each sample in the dataset consists of a sequence of $N+1$ frames $(\mathcal{F}_i)_{i=0}^{N}$ and a natural language instruction ℓ. The first N frames $\{\mathcal{F}_i\}_{i=0}^{N-1}$ are observed views, and the last frame $\mathcal{F}_N$ is the target view that satisfies the intent. Each frame $\mathcal{F}_i = (I_i, D_i, M_i, \mathbf{g}_i)$ contains an RGB image $I_i \in \mathbb{R}^{3\times H\times W}$, a depth map D_i, a semantic mask M_i, and camera parameters $\mathbf{g}_i$.

5.1 The I-Perceive Dataset

We leverage two complementary sources of data: real-world scene-scanning datasets, including ScanNet [10] and CA-1M [22], and simulated indoor scenes from HSSD [21]. We use the following pipeline to construct active perception tasks:

(a) Target sampling. For each data point, we first sample a target frame $\mathcal{F}_N$, and the camera pose $\mathbf{g}_N$, corresponding to meaningful observation intents using object-centric heuristics, leveraging dataset-specific annotations (e.g., masks, bounding boxes, or asset templates). For ScanNet and CA-1M, target frames are sampled from the scanning video, by maximizing target visibility and centered framing; for HSSD, we construct reusable target-pose templates for each object type, with predefined sampling ranges and perception intents. Across all datasets, we apply filtering to reject frames with collision or insufficient visibility, producing semantically meaningful and geometrically feasible target views.

(b) Query and context sampling. Given a target viewpoint, we then sample the query and context frames, $\{\mathcal{F}_i\}_{i=0}^{N-1}$, together with the camera poses $\{\mathbf{g}_i\}_{i=0}^{N-1}$, to emulate partial scene observation. For real-world scanning datasets, since images are already collected along scanning trajectories, we sample query and context frames around the target while ensuring consecutive camera-pose proximity and the overall view diversity. For HSSD, we explicitly plan collision-free paths among sampled target poses, and sample query and context frames along the resulting trajectories. This trajectory-based sampling better mimics online robot perception, where context images are accumulated sequentially.

(c) Free-form instruction generation. We use a VLM to generate natural language instructions ℓ conditioned on the target view image, verbalized viewpoint attributes (e.g., relative offset, viewing direction, and framing), and semantic description of the target object when available. The model converts these inputs into free-form instructions that specify what to observe and how to view it, covering diverse intent types such as object-centric, spatial, viewpoint-specific, and functional descriptions. Examples and prompts are provided in Appendix A.4.

(d) Disambiguation with 2D pointing. When an instruction refers to multiple similar objects or regions, we use a VLM to detect ambiguity and augment the instruction ℓ with a 2D point reference ("*(x, y) in image*") to disambiguate the intended target.

(e) World-model-based view improvement. We refine target viewpoints using a generative world model to increase diversity and naturalness. A VLM first converts the instruction of the active perception task into a camera-motion prompt, which is used by a video generation model (Wan2.2[35]) to synthesize a intent-consistent video. We then use VGGT to extract the precise camera motion from the video, and apply it to adjust the target camera pose $\mathbf{g}_N$. This improves the realism of target views and their alignment with task instructions.

With the above pipeline, steps (a)–(d) yield a pre-training dataset with 162K samples from ScanNet and CA-1M, and 70K from HSSD. Step (e) is applied to $\sim$30% of CA-1M scenes, producing an additional 96K high-quality samples for post-training. See additional pipeline details in Appendix A.

5.2 Training Details

We first train I-Perceive on the pre-training dataset for 5 epochs, with VLM backbone frozen and all learning targets enabled. Then we fine-tune the model with the enhanced post-training dataset for 3 epochs, unfreezing VLM backbone with LoRA [13] and disabling semantic mask and point heatmap

losses to focus on instruction following. The training process takes 30 hours on 32 H200 GPUs. See further details in Appendix B.

6 Experiments

Benchmarks. We evaluate I-Perceive on both in-distribution and out-of-distribution settings across diverse indoor environments.

(a) HSSD (in-distribution). We construct a set of 52 novel tasks with scene-level splits, where all test scenes are entirely unseen during training. To ensure credible and unbiased evaluation, target viewpoints are annotated by human experts rather than generated by the data pipeline.

(b) Replica (out-of-distribution). To assess zero-shot generalization, we construct a new test set on the Replica scene dataset [32], which provides reconstructed real-world indoor environments. We generate 89 novel language-conditioned tasks, with target views annotated by human experts.

Baselines. We compare I-Perceive against a range of relevant baselines. See detailed implementations of baselines in Appendix C. All VLM components used by the baselines are GPT-5.4.

(a) Language-Conditioned Next-Best-View (NBV+L). NBV methods are typically designed for coverage maximization and do not directly accept language tasks. To enable comparison, we implement a language-conditioned variant that samples candidate viewpoints around the target region and uses a VLM to select the best view based on instruction alignment.

(b) Vision-Language Navigation with Perception View Selection (VLN+P). We then implement a VLN-style baseline. It first navigates toward the target object based on language instructions using NaVILA [8], a recent open-source VLN model. A heuristic viewpoint selection step is then applied to orient the camera toward the target object.

(c) VLM Prompting (VLM). We also include a general-purpose VLM baseline. Given visual context and language instructions, the model is prompted to output a camera location and a look-at point on the image-space. This output representation leverages the strength of VLMs in 2D pointing.

Metrics. Evaluating language-conditioned active perception is inherently challenging due to the openness of language intents and the multi-modality of valid viewpoints. We design the following set of complementary metrics[2]. See full details in Appendix D.

(a) Constraint Violation (CV). We define the constraint violation rate as the fraction of predicted view that misses the target object or cause a collision. This metric captures the most obvious failures.

(b) Pose Error (e_t, e_r). We report translation and rotation errors between predicted and annotated camera poses. But note that, these values can be misleading under multi-modal solutions, as they over-commit to a single annotated pose.

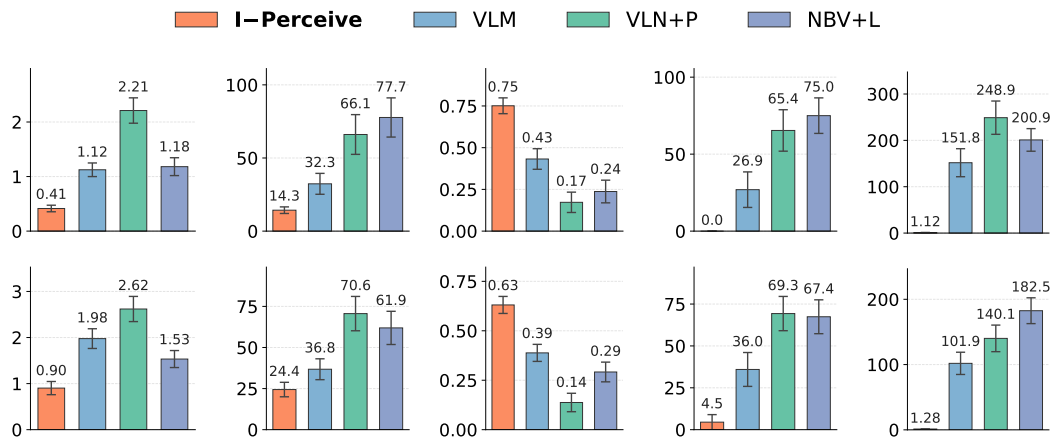
(c) Voxel-Matching F1 (F1). We further design a content-centric measure of viewpoint quality, with tolerance to minor disturbances. A 3D voxel covered by the ground-truth view is considered *matched* if there exists a voxel covered by the predicted view in its neighborhood (approximately 10 cm). We then compute precision, recall, and the F1 score over matched voxel sets. We regard Voxel-Matching F1 as the most reliable "absolute" measure.

(d) Judge Ranking (Rank). We further implement a list-wise ranking protocol to provide a reliable "relative" measure of viewpoint quality. Given a set of candidate views generated by different methods, we use GPT-5.4 and human experts to rank the candidates, and report the average ranks and winning rates. See detailed implementations in Appendix D.4.

6.1 Performance Comparison

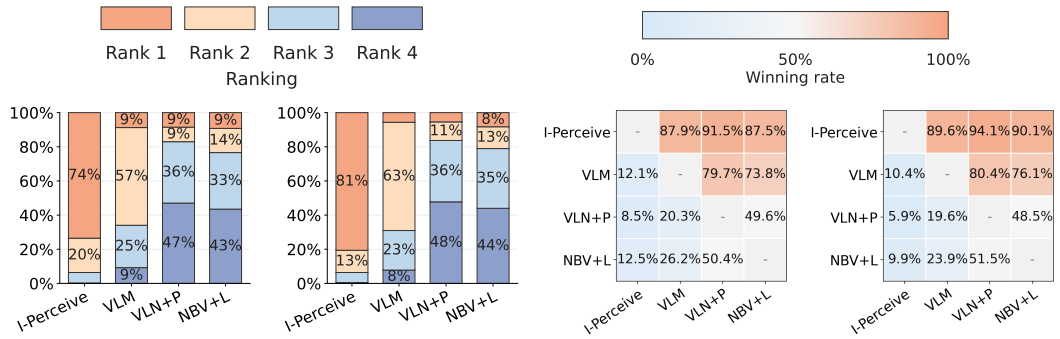
Absolute performance. Fig. 3 summarizes the performance of I-Perceive and all baselines on both in-distribution (HSSD) and out-of-distribution (Replica) settings. Across both settings, I-Perceive consistently achieves the best performance with the lowest runtime, yielding substantially lower

[2]Unless otherwise noted, uncertainty intervals are 95% non-parametric bootstrap intervals over test cases.



(a) $e_t(m) \downarrow$ (b) $e_r(deg) \downarrow$ (c) $F1 \uparrow$ (d) $CV(\%) \downarrow$ (e) Runtime $(s) \downarrow$

Figure 3: Absolute performance of I-Perceive and baselines on HSSD (Upper) and Replica (Bottom). Detailed numerical results are provided in Appendix F.



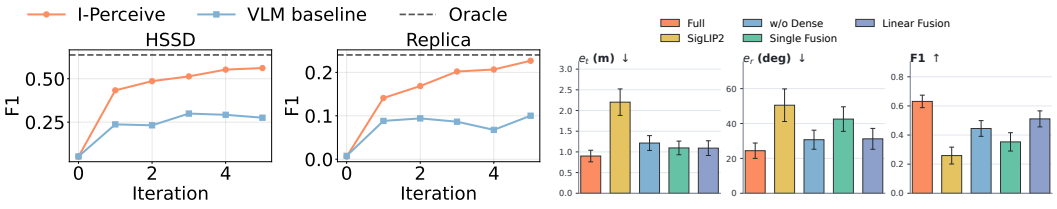
(a) GPT ranking (b) Human ranking (c) GPT winning rate (d) Human winning rate

Figure 4: Judge ranking results of I-Perceive and baselines on HSSD and Replica. (a,b) show the distribution of ranks assigned to each method by GPT and human judges. (c,d) show the pairwise winning computed from the rankings. Rank 1 is best; the winning rate matrix captures row-to-column comparison.

constraint violation, translation and rotation errors, and significantly higher voxel-matching F1 scores. Among the baselines, VLN+P is the worst-performing method because its heuristic view selection disregards language intent. NBV+L performs similarly, due to its assumption of fully observable targets. VLM prompting yielded reasonable task performance, but significantly lower view qualities than I-Perceive, despite using a much longer inference time. In contrast, I-Perceive produced highly consistent views aligned with human annotations, with near-zero constraint violations, through language-conditioned geometric reasoning trained on a wide set of active perception tasks.

Relative performance. Fig. 4 reports relative performance under the judge-ranking protocol. I-Perceive is ranked significantly higher than all baselines in both in-distribution and out-of-distribution settings, and achieves high pairwise winning rates (87.5%–94.1%) over all baselines. Human and GPT rankings are highly consistent.

Inference time. Finally, I-Perceive is substantially more efficient than all baselines, as it requires only a single forward pass for viewpoint prediction. The mean inference time is 1.1s on HSSD and 1.3s on Replica, which involve an average of 2.6 and 3.2 context frames per test case, respectively. Appendix D.5 further provides runtime and VRAM analysis under varying context sizes.



(a) (b)

Figure 5: Additional analysis on (a) closed-loop active perception and (b) ablation studies. (a) Closed-loop execution of I-Perceive progressively improves viewpoint quality across steps. (b) Ablation results show that replacing the VLM backbone with SigLIP2, removing dense supervision, or simplifying deep semantic fusion consistently degrades performance, highlighting their importance.

6.2 Performance on Close-Loop Active Perception

Fig. 5(a) shows I-Perceive's performance when operating in a closed-loop setting. In this setting, no initial context is provided (single start image), and the context must be progressively collected via closed-loop execution. The targets are completely out of view at start, requiring active exploration. Results show that I-Perceive consistently improves performance over successive steps, in the final step, nearly matches the oracle performance with full context, demonstrating effective closed-loop exploration. It is worth noting that I-Perceive is not explicitly trained for closed-loop operation. This phenomenon emerged from I-Perceive's capability to accept variable-length contexts. See detailed implementations in Appendix H.1.

6.3 Ablation Studies

To validate key design choices, we perform ablation studies on the evaluation tasks. Results are shown in Fig. 5(b). Replacing the VLM with a lightweight encoder (SigLIP2 [34]) leads to a large performance drop, indicating that pretrained VLMs is essential for semantic understanding. Removing auxiliary dense prediction heads (i.e., depth, point maps, and semantic masks) during training also degrades performance, due to weaker geometric and semantic reasoning. Simplifying the fusion mechanism (i.e., fusion on a single final layer or multi-layer fusion with linear modules) harms performance, underscoring the importance of cross-attention-based progressive semantic fusion for effective language conditioning. See details of the ablated variants in Appendix G.

6.4 Qualitative Results

Qualitative results, including those on in-distribution, out-of-distribution, and real-world evaluation, as well as closed-loop execution, are available at `https://roboticsjtu.github.io/I-Perceive-Page/`.

7 Conclusion

We presented I-Perceive, a foundation model for language-conditioned active perception in large-scale indoor environments. By integrating semantic grounding from vision-language models with multi-view geometric reasoning, I-Perceive directly predicts task-conditioned camera poses from visual context and natural language instructions. Extensive experiments demonstrate strong performance across both in-distribution and out-of-distribution environments, significantly outperforming existing active perception, VLM prompting, and VLN-based baselines in viewpoint accuracy, constraint satisfaction, preference-based evaluation, and runtime. In addition, I-Perceive naturally supports closed-loop active perception, progressively improving viewpoint quality through sequential interactions under partial observability.

Limitations and Future Work. Our current constraint checking only detects collisions of generated viewpoints without correcting them. Future work will incorporate collision-aware viewpoint optimization through gradient-based or sampling-based refinement. In addition, the current training pipeline remains closely tied to object annotations in scene-scanning datasets and simulation environments. Extending to richer and more open-ended targets will require stronger integration with

generative world models. Finally, the dual-pathway architecture relies heavily on pretrained VLMs and geometric foundation models. While effective in limited-data regimes, future work will explore unified architectures that can better exploit large-scale training data.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (Grant No. 2024YFB4707600) and NSFC under grant No. 62303304.

References

[1] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International journal of computer vision*, 1(4):333–356, 1988.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[5] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018.

[6] Meng Cao, Xingyu Li, Xue Liu, Ian Reid, and Xiaodan Liang. Spatialdreamer: Incentivizing spatial reasoning via active mental imagery. *arXiv preprint arXiv:2512.07733*, 2025.

[7] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[8] A.-C. Cheng et al. Navila: Legged robot vision-language-action model for navigation. *arXiv:2412.04453*, 2024.

[9] Cl Connolly. The determination of next best views. In *Proceedings. 1985 IEEE international conference on robotics and automation*, volume 2, pages 432–435. IEEE, 1985.

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[11] Jeffrey Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A comparison of volumetric information gain metrics for active 3d object reconstruction. *Autonomous Robots*, 42(2):197–208, 2018.

[12] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653, 2021.

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

[14] Seth A Hutchinson, Robert L Cromwell, and Avinash C Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1068–1075. IEEE, 1988.

[15] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016.

[16] Snehal Jauhri, Sophie Lueth, and Georgia Chalvatzaki. Active-perceptive motion generation for mobile manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1413–1419. IEEE, 2024.

[17] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3813–3822, 2016.

[18] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

[19] Justin Kerr, Kush Hari, Ethan Weber, Chung Min Kim, Brent Yi, Tyler Bonnen, Ken Goldberg, and Angjoo Kanazawa. Eye, robot: Learning to look to act with a bc-rl perception-action loop. *arXiv preprint arXiv:2506.10968*, 2025.

[20] Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838, 2022.

[21] Mukul Khanna*, Yongsen Mao*, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint*, 2023.

[22] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22225–22233, 2025.

[23] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[24] Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, Yuxin Du, Jiting Liu, Encheng Gu, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025.

[25] Mengzhen Liu, Enshen Zhou, Cheng Chi, Yi Han, Shanyu Rong, Liming Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Sapave: Towards active perception and manipulation in vision-language-action models for robotics. *arXiv preprint arXiv:2603.12193*, 2026.

[26] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. Zson: Zero-shot object-goal navigation using multimodal goal embeddings. *Advances in Neural Information Processing Systems*, 35:32340–32352, 2022.

[27] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.

[28] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *European Conference on Computer Vision*, pages 558–573. Springer, 2020.

[29] Kaixian Qu, Han Wang, Victor Klemm, Cesar Cadena, and Marco Hutter. An efficient beam search algorithm for active perception in mobile robotics. *arXiv preprint arXiv:2604.23327*, 2026.

[30] Juil Sock, S Hamidreza Kasaei, Luis Seabra Lopes, and Tae-Kyun Kim. Multi-view 6d object pose estimation and camera motion planning using rgbd images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2228–2235, 2017.

[31] Venkatesh Sripada, Samuel Carter, Frank Guerin, and Amir Ghalamzan. Ap-vlm: Active perception enabled by vision-language models. *arXiv preprint arXiv:2409.17641*, 2024.

[32] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[33] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[34] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[35] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[36] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[37] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[38] Laurana M Wong, Christophe Dumont, and Mongi A Abidi. Next best view system in a 3d object modeling task. In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No. 99EX375)*, pages 306–311. IEEE, 1999.

[39] Wansen Wu, Tao Chang, Xinmeng Li, Quanjun Yin, and Yue Hu. Vision-language navigation: a survey and taxonomy. *Neural Computing and Applications*, 36(7):3291–3316, 2024.

[40] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[41] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025.

[42] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 42–48. IEEE, 2024.

[43] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 6(3):225–245, 2020.

[44] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.

[45] Wanyue Zhang, Yibin Huang, Yangbin Xu, JingJing Huang, Helu Zhi, Shuo Ren, Wang Xu, and Jiajun Zhang. Why do mllms struggle with spatial understanding? a systematic analysis from data to architecture. *arXiv preprint arXiv:2509.02359*, 2025.

[46] Enshen Zhou, Cheng Chi, Yibo Li, Jingkun An, Jiayuan Zhang, Shanyu Rong, Yi Han, Yuheng Ji, Mengzhen Liu, Pengwei Wang, et al. Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2512.13660*, 2025.

[47] Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Zheng Huang, Mingyu Liu, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, et al. Active-o3: Empowering multimodal large language models with active perception via grpo. *arXiv preprint arXiv:2505.21457*, 2025.

[48] Yanwen Zou, Chenyang Shi, Wenye Yu, Han Xue, Jun Lv, Ye Pan, Chuan Wen, and Cewu Lu. Activeglasses: Learning manipulation with active vision from ego-centric human demonstration. *arXiv preprint arXiv:2604.08534*, 2026.

A Dataset and Data Generation Details

This appendix provides implementation-level details for the data, baselines, and evaluation protocol used in the main paper. Prompt files are referenced by compact aliases rather than copied in full; the complete prompts are included in the supplementary prompt folder. The alias mapping is shown in Table 1.

Table 1: Prompt aliases used in this appendix.

Alias	File	Role
P-DATA-INST	`prompts/data_instruction_generation.txt`	Generates short active-perception instructions.
P-DATA-VIDEO	`prompts/data_video_motion_translation.txt`	Translates instructions into first-person video descriptions.
P-NBVL-LOC	`prompts/baseline_nbvl_target_localization.txt`	Localizes the target frame and 2D box for NBV+L.
P-NBVL-AZ	`prompts/baseline_nbvl_azimuth_selection.txt`	Selects NBV+L azimuth candidates.
P-NBVL-EL	`prompts/baseline_nbvl_elevation_selection.txt`	Selects NBV+L elevation candidates.
P-NBVL-DIST	`prompts/baseline_nbvl_distance_selection.txt`	Selects NBV+L distance candidates.
P-NBVL-REFINE	`prompts/baseline_nbvl_pose_refinement.txt`	Applies final NBV+L pose refinement.
P-VLNP-NAV	`prompts/baseline_vlnp_instruction_translation.txt`	Converts active-perception requests into VLN commands.
P-VLNP-PITCH	`prompts/baseline_vlnp_pitch_refinement.txt`	Applies VLN+P pitch refinement.
P-VLM-POSE	`prompts/baseline_vlm_camera_lookat_prediction.txt`	Predicts camera and look-at points for the direct VLM baseline.

A.1 Unified Data Format

All training and evaluation samples are represented as language-conditioned viewpoint tuples. A sample contains a start frame, zero or more context frames, a target frame, and metadata:

- RGB images for the start, context, and target views.
- Camera poses ordered as start, context frames, and target frame.
- Optional depth maps and semantic segmentation maps, used for auxiliary supervision and geometric evaluation.
- A natural-language instruction and optional 2D point reference in the start image.
- Camera intrinsics and scene metadata, including a scene identifier for grouped evaluation.

Each case is stored as a self-contained sample containing RGB frames, camera poses, optional dense annotations, and metadata. Camera poses follow the OpenCV camera convention: X right, Y down, and Z forward. Unless otherwise stated, camera poses are represented as camera-to-world transformations in the corresponding scene coordinate frame, while model-predicted camera parameters are expressed relative to the start frame.

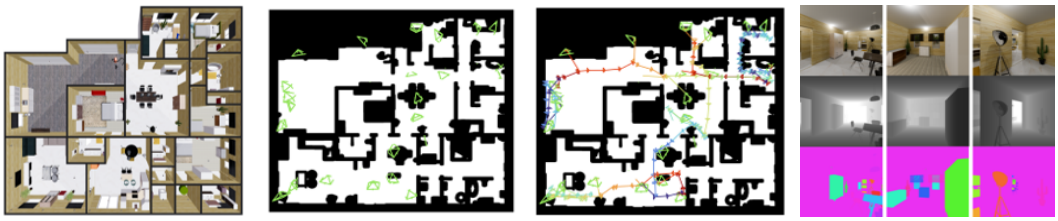
A.2 Real-World Scanning Data

For ScanNet and CA-1M, target viewpoints are sampled from existing scene-scanning trajectories. Since these datasets were not captured specifically for active perception, we use object-centric heuristics to identify useful observations. For ScanNet, semantic masks are used to select frames where a target category occupies a sufficiently visible, centered region. For CA-1M, object bounding boxes and semantic captions are used instead. Candidate target frames are filtered to remove views where the target is too small, heavily occluded, poorly centered, or geometrically inconsistent with the sampled observation intent.

Given a target viewpoint, start and context frames are sampled from nearby frames in the original scanning sequence. For the CA-1M world-model generation pipeline, the generator considers three start/context strategies: retaining the original query frame, selecting another nearby target-visible frame, or selecting nearby target-invisible frames to create harder partial-observability cases. The neighborhood window is currently set to approximately ± 150 frames, corresponding to about five seconds for a 30 FPS video.

A.3 HSSD Simulation Data

For HSSD, we generate active-perception samples from high-quality 3D indoor scenes with controllable rendering and geometry. The HSSD pipeline starts from reusable target-pose templates attached to object categories or object instances. Each template specifies a look-at region, allowable



(a) HSSD Scene (b) Annotated target frame (c) Generated trajectories (d) Rendered frame

Figure 6: **Data generation pipeline in simulated HSSD environments.** (a) Top-down view of a representative 3D house scene. (b) Target viewpoints (green frustums) are uniformly sampled across the navigable floor map. (c) To mimic real-world online exploration, robot movement trajectories towards the target viewpoints are generated using path planning. (d) Context frames are sampled along the movement history, providing dense multi-modal learning signals including RGB, depth maps, and semantic masks.

viewing directions, distance ranges, and a short objective description. These templates support both category-level annotation for geometrically regular object classes and instance-level annotation for object categories with large shape variation.

For each scene, collision-free paths are planned between sampled target poses. Start and context frames are then sampled along these trajectories to mimic how a mobile robot accumulates observations over time. RGB images are rendered photorealistically, while depth and semantic segmentation maps are rendered from the simulator geometry. Human annotators use an interactive 3D interface to place target views, inspect start/context/target observations, and verify target visibility for benchmark cases.

A.4 Instruction Generation and 2D Point Disambiguation

Natural-language instructions are generated by a VLM conditioned on the target view and available semantic information. For ScanNet, the semantic category label is provided. For CA-1M, the object caption and bounding box are provided. For HSSD, the prompt can additionally use target-pose objective descriptions and rendered start/target images.

The instruction generator is constrained to produce short, action-oriented active-perception commands that describe where the agent should look or move its camera, rather than long analytical goals. It also avoids ego-centric references such as "turn left" because the same instruction may be paired with different start frames during training. The relevant prompt aliases are:

- P-DATA-INST: active-perception instruction generation.
- P-DATA-VIDEO: translation from an active-perception instruction to a first-person video description for the world model.

When an instruction is ambiguous because multiple objects or regions could satisfy it, the data pipeline can add a 2D point reference in the start image. The text instruction and optional point are represented with a compact JSON-style schema so the model observes a consistent instruction format.

A.5 World-Model Generated Post-Training Data

The enhanced post-training data is generated with PerceiveBench. The pipeline first samples a key frame from CA-1M whose center points to a meaningful object. A VLM receives the image, the target bounding box, and the semantic caption, and generates a short active-perception instruction. A second prompt translates the instruction into a first-person video description. Wan2.2 then synthesizes a static-scene, camera-motion video, and VGGT estimates the camera motion from sampled video frames. The final frame pose is used as an improved target pose for post-training.

The generation pipeline is run as an offline data-production workflow using remote VLM, video-generation, and pose-estimation services. Generated CA-1M samples are validated before use; invalid samples are separated by failure reason, including malformed model outputs, failed pose recovery, and geometric validation failures.

A.6 Manual annotation in evaluation set

To ensure the high fidelity and semantic consistency of our evaluation suite, we conducted a rigorous manual annotation process for the HSSD and Replica datasets. All annotations were performed by researchers specialized in computer vision and robotics to guarantee professional-grade data quality.



Figure 7: **Active Perception Annotation Interface.** The tool is designed to ensure rigorous evaluation set construction. **(A)** The main viewport allows annotators to navigate 3D environments and determine the optimal field of view (highlighted central square). **(B)** The keyframe tracker records the precise 6D poses of the *start*, *context*, and *target* views. And the instruction module where annotators provide natural language guidance and assess object visibility.

Task Definition For each data point, annotators were tasked with identifying three key viewpoints: the start view, context view, and target view. Specifically, given a scene, annotators had to:

- Identify a coherent sequence of views representing the task's progression.
- Formulate a natural language instruction that accurately describes the transition to the target state.
- Verify that the selected target view represents the optimal viewpoint for fulfilling the given instruction within that scene.

Training and Standardization Before the formal annotation phase, all annotators underwent a unified training session. This training focused on standardizing the subjective understanding of the "optimal viewpoint" and "active perception." We established clear criteria for viewpoint quality, such as occlusion-free visibility of target objects and appropriate camera-to-object distances, ensuring that the subjective bias across different annotators was minimized.

Quality Control We implemented a multi-stage quality assurance pipeline:

1. Self-Inspection: Annotators were required to re-evaluate each sequence immediately after labeling to ensure the instruction-view alignment.
2. Secondary Verification: Upon completion, a secondary screening was performed by a separate expert. Any sequences with ambiguous instructions or suboptimal target poses were either refined or discarded. This rigorous filtering ensures that our evaluation set serves as a gold standard for assessing active perception capabilities.



(a) Input frame (b) Video's endframe (c) Recovered target pose (d) Resampled contexts

Figure 8: **The world model data generation pipeline.** (a) The initial frame from the scanning dataset. (b) The endframe of the video synthesized by the world model, which serves as the new target view. (c) The recovered camera trajectory and target pose estimated via VGGT. (d) 3D scene reconstruction and context frame resampling to formulate the final training data point.

B Model and Training Details

B.1 Backbones and Fusion Layers

The vision-language pathway uses Qwen3-VL-4B to encode the image sequence and instruction. Rather than using only the final hidden state, we extract intermediate key-value features from selected transformer layers and inject them into the geometric reasoning pathway. We collect VLM KV features from layers $[8, 15, 23, 35]$ in zero-based indexing and fuse them into geometric layers $[0, 5, 12, 18]$. In one-based notation, this corresponds to VLM layers $[9, 16, 24, 36]$ and geometric layers $[1, 6, 13, 19]$.

The geometric pathway is based on a VGGT-style multi-view transformer. Each frame contains a camera token, register tokens, semantic tokens, and patch tokens. Semantic tokens are placed on the same patch grid as visual tokens and serve as carriers for language-grounded information. For observed image frames, semantic tokens attend to VLM image KV features with 2D rotary position embeddings. For the latent instruction/target frame, semantic tokens attend to instruction KV features with 1D rotary position embeddings on the keys. The cross-attention update is gated by a learnable LayerScale parameter initialized to zero for stable training.

B.2 Prediction Heads and Losses

The camera head predicts the target camera parameters and the observed-frame camera parameters. Auxiliary heads produce depth maps, 3D point maps, 2D grounding heatmaps with visibility logits, and optional target-object semantic masks. During inference, only the camera prediction is required. During training, the total objective is a weighted sum of the following components:

- camera pose loss for translation, rotation, and focal representation;
- depth loss and confidence loss;
- 3D point-map loss and confidence loss;
- grounding heatmap, visibility, and soft-argmax coordinate losses;
- semantic-mask loss combining pixel-wise and region-level supervision.

B.3 Training Schedule

We first train on the pre-training set for 5 epochs with the VLM backbone frozen and all learning targets enabled. We then fine-tune on the enhanced post-training set for 3 epochs, using LoRA to adapt the VLM attention projections and disabling the semantic mask and point-heatmap losses to emphasize instruction following. The model is trained with mixed precision: parameters are kept in fp32 during training, forward computation uses bf16 autocast by default, and released inference checkpoints are stored in bf16.

C Baseline Designs

All methods are evaluated through the same pose-evaluation protocol. Each method receives the same start/context images, camera metadata, and instruction, and returns a single predicted camera-to-world pose. The returned pose is rendered and scored with the same metric definitions across methods.

C.1 NBV+L: Language-Conditioned Next-Best-View

NBV+L is a training-free, VLM-guided next-best-view baseline. It first asks a VLM to select the relevant observed frame and target bounding box. A frozen VGGT-1B model estimates geometry from the start and context frames; the prediction is refined by aligning the start-frame depth scale to the known data depth. The selected 2D target is back-projected to 3D, and a sparse occupancy model is built for collision and line-of-sight checks.

After target localization, the baseline decomposes view selection into interpretable low-cardinality decisions: azimuth selection over eight compass directions, elevation selection over five vertical angles, distance selection over the metric grid $\{0.3, 0.8, 1.3, 1.8, 2.3, 2.8\}$ meters, and a final semantic micro-refinement over yaw, pitch, and distance. The corresponding prompt aliases are P-NBVL-LOC, P-NBVL-AZ, P-NBVL-EL, P-NBVL-DIST, and P-NBVL-REFINE.

C.2 VLN+P: Vision-Language Navigation with Perception Refinement

VLN+P reformulates active perception as a Vision-Language Navigation (VLN) problem. NaVILA is a VLM-based VLN model that maps visual observations and navigation instructions to discrete movement actions. A VLM first translates the active-perception instruction into a NaVILA-style navigation command using P-VLNP-NAV. NaVILA then performs an online rollout from the start pose, using rendered observations at each step. Actions are parsed into `stop`, forward moves at 25 cm granularity, or left/right turns at 15 degree granularity. The rollout stops when NaVILA predicts stop, after at most 20 steps, or after repeated collision-blocked forward moves. Since VLN policies target goal-reaching rather than viewpoint selection, a final VLM call applies a pitch correction clamped to $[-45°, 45°]$ using P-VLNP-PITCH.

C.3 VLM Prompting

The VLM baseline removes VGGT geometry, candidate sampling, and occupancy checks. Given the start image, context images, instruction, and optional target point hint, the VLM emits two 3D marks in the start-frame coordinate system: a camera point and a look-at point. Each mark is represented as a per-mille image coordinate (u, v) and a metric depth. These points are back-projected through the known start-frame intrinsics and pose, and the final camera is constructed with a gravity-up look-at transform. Degenerate predictions where the two points nearly coincide are repaired by nudging the look-at point forward along the start camera's $+Z$ axis. The prompt alias is P-VLM-POSE.

C.4 Human Evaluation for Model Predictions (Rank-H)

To conduct a rigorous human evaluation of the generated target views, we developed a dedicated listwise ranking interface, as shown in **Figure 9**. Consistent with the evaluation set construction process, all Rank-H judgments were performed by the same group of domain experts to ensure high-quality and reliable assessments.

Interface Design and Anonymization To prevent evaluation bias, all model predictions are strictly anonymized and their spatial positions (A, B, C, D) are randomly shuffled for each test instance. As illustrated in the interface, the evaluator is provided with comprehensive context:

- **Task Instruction:** The natural language command defining the semantic goal (e.g., "Prepare to open the oven").
- **Reference Sequence:** The complete trajectory including the start view, intermediate context views, and the ground-truth (GT) target view.
- **Anonymized Predictions:** Four candidate target views comprising the output of our proposed method and three established baselines.

Ranking Protocol and Criteria Evaluators are tasked with providing a strict, forced-choice ranking of the four candidates from 1 (best) to 4 (worst), with **no ties permitted**. The primary evaluation criterion is the semantic alignment between the generated view and the text instruction, ensuring the output is visually consistent with the underlying 3D scene geometry. Crucially, the GT target view is provided merely as a contextual reference to help evaluators understand the spatial progression of the scene; it does not serve as the absolute ranking standard. Evaluators are explicitly instructed to reward models that successfully fulfill the text instruction, even if the predicted viewpoint differs from the GT. To ensure efficiency during large-scale evaluation, the interface supports seamless keyboard shortcuts for ranking assignment and navigation.

D Evaluation Metrics and Ranking Protocol

D.1 Pose Errors

For each test case, we compare the predicted pose $\mathbf{g}_{\text{pred}}$ to the human-annotated reference pose $\mathbf{g}_{\text{gt}}$. The translation error is the Euclidean distance between camera centers. The rotation error is the geodesic distance on $SO(3)$, reported in degrees in the main table.

D.2 Voxel-Matching F1

Let $\mathcal{V}(\mathbf{g})$ denote the set of scene voxels visible from camera pose $\mathbf{g}$ under frustum and occlusion testing. A ground-truth voxel $v \in \mathcal{V}(\mathbf{g}_{\text{gt}})$ is considered recalled if there exists a predicted voxel $u \in \mathcal{V}(\mathbf{g}_{\text{pred}})$ satisfying $\|u - v\|_\infty \le k$. Precision is defined symmetrically. We use $k = 2$ voxel units in the main experiments, corresponding to approximately 10 cm under the 5 cm voxel pitch used for HSSD:

$$\begin{aligned} \mathrm{sR}_k &= \frac{|\{v \in \mathcal{V}_{\text{gt}} : \exists u \in \mathcal{V}_{\text{pred}}, \|u - v\|_\infty \le k\}|}{|\mathcal{V}_{\text{gt}}|}, \\ \mathrm{sP}_k &= \frac{|\{u \in \mathcal{V}_{\text{pred}} : \exists v \in \mathcal{V}_{\text{gt}}, \|u - v\|_\infty \le k\}|}{|\mathcal{V}_{\text{pred}}|}, \end{aligned} \tag{3}$$

$$\mathrm{F1}_k = \frac{2\ \mathrm{sP}_k\ \mathrm{sR}_k}{\mathrm{sP}_k + \mathrm{sR}_k}. \tag{4}$$

D.3 Constraint Violation

Constraint violation (CV) measures the fraction of test cases where the predicted view either misses the target or violates physical feasibility. We track two flags: `target_visible`, annotated by human inspection of the rendered prediction, and `in_collision`, computed by a sphere-collision check around the predicted camera center against the scene voxel grid. A case counts as a violation if the target is not visible or the predicted camera is in collision. Only cases with both annotations available are included in the CV denominator.

D.4 List-Wise Judge Ranking Evaluation Details

To comprehensively evaluate the generated target views, we employ a list-wise ranking protocol. For a given test case, the judge is presented with the task instruction, the reference context, and the anonymized rendered outputs of all competing methods. The judges rank candidate views according to how well they satisfy the semantic instruction and provide a physically plausible, useful observation of the target. We report the *mean rank*, where a lower value indicates better performance.

Evaluation Groups and Aggregation The main results reported in our experiments are based on two distinct judge groups: an automated LLM judge, GPT-5.4 (Rank-G), and human annotators (Rank-H). To mitigate position bias, the spatial ordering (A, B, C, D) of the anonymized candidates is randomly shuffled. For each evaluation group, three independent shuffled runs are aggregated. We first average the ranks for the same test case across the three runs, and then compute the dataset-level means along with their bootstrap confidence intervals.

Table 2: Runtime and VRAM usage of I-Perceive as a function of the number of input images on GTX 4090.

#Images	1	2	3	4	5
Runtime (s)	0.78	1.42	2.13	2.90	3.82
VRAM (GB)	16.07	16.92	17.84	18.79	19.74

Human Evaluation Interface (Rank-H) To conduct the human evaluation rigorously, we developed a dedicated ranking interface, illustrated in **Figure 9**. Consistent with the evaluation set construction process, all Rank-H judgments were performed by domain experts. As shown in the interface, the ground-truth (GT) target view is provided merely as a contextual reference (alongside the start and context views) to illustrate the spatial progression of the scene; it is **not** shown as a candidate to be ranked.

The interface enforces a strict, forced-choice ranking from 1 (best) to 4 (worst) with **no ties permitted**. Evaluators are explicitly instructed to prioritize the semantic alignment with the text instruction, rewarding models that successfully fulfill the task even if the predicted viewpoint differs from the GT reference.

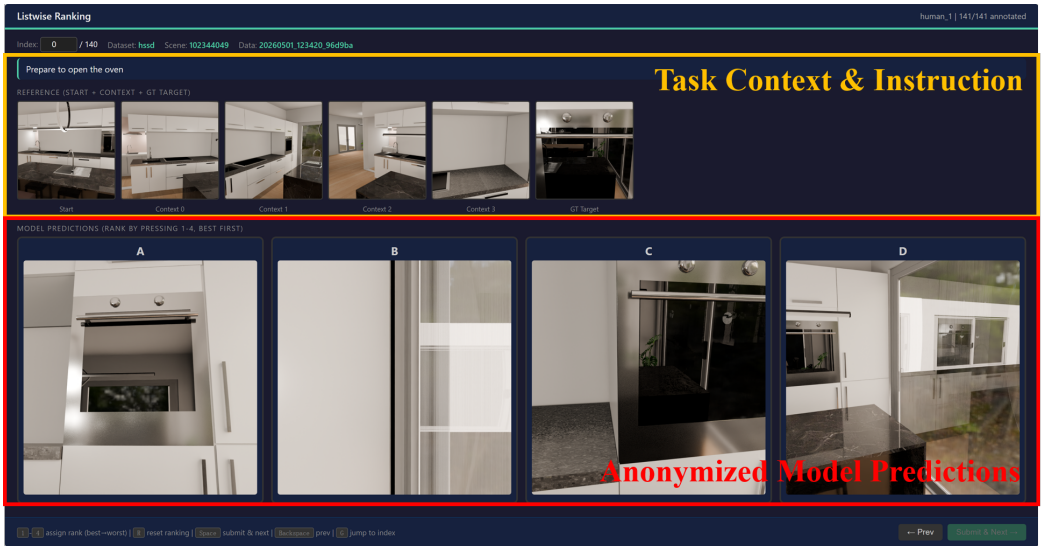


Figure 9: **The Listwise Ranking Interface for Human Evaluation (Rank-H). (A)** Evaluators are provided with the natural language instruction and the reference temporal sequence (from start to ground-truth target) to understand the task context. **(B)** Four anonymized and randomly shuffled model predictions are presented for side-by-side comparison.

D.5 Confidence Intervals and Runtime

For e_t, e_r, F1, Rank-G, and Rank-H, confidence intervals are computed with a non-parametric percentile bootstrap over test cases using 10,000 resamples. The table reports the mean and a conservative half-width of the 95% interval. Runtime is reported as the mean runtime returned by each algorithm implementation and is not assigned a confidence interval. For I-Perceive, runtime includes model forward and pose-refinement post-processing, but excludes data adapter preprocessing and host-device transfer.

E Experimental Details

E.1 Benchmarks

HSSD is used as the in-distribution benchmark. The HSSD test set contains 52 language-conditioned tasks from held-out 5 scenes. Target viewpoints are manually annotated by human experts rather than generated by the training-data pipeline.

Table 3: Performance comparison on HSSD and Replica. We report constraint violation (CV), voxel-matching F1 (F1), pose error (e_t, e_r), mean judge rank from GPT-5.4 (Rank-G) and human annotators (Rank-H), and mean runtime. ↑: higher is better; ↓: lower is better.

Dataset	Method	$\mathbf{e_t}$ (m) ↓	$\mathbf{e_r}$ (deg) ↓	F1 ↑	Rank-G ↓	Rank-H ↓	CV (%) ↓	Time (s) ↓
HSSD	NBV+L	1.18 ± 0.16	77.70 ± 13.39	0.2373 ± 0.0679	3.28 ± 0.25	3.38 ± 0.22	75.0 ± 11.5	200.9 ± 24.4
HSSD	VLN+P	2.21 ± 0.23	66.06 ± 13.59	0.1726 ± 0.0610	3.21 ± 0.21	3.23 ± 0.19	65.4 ± 13.5	248.9 ± 35.9
HSSD	VLM	1.12 ± 0.12	32.33 ± 7.15	0.4322 ± 0.0617	2.25 ± 0.19	2.21 ± 0.16	26.9 ± 11.5	151.8 ± 30.2
HSSD	**I-Perceive**	$\mathbf{0.41 \pm 0.06}$	$\mathbf{14.31 \pm 2.25}$	$\mathbf{0.7511 \pm 0.0474}$	$\mathbf{1.26 \pm 0.15}$	$\mathbf{1.18 \pm 0.14}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{1.1 \pm 0.1}$
Replica	NBV+L	1.53 ± 0.18	61.92 ± 10.10	0.2917 ± 0.0499	3.01 ± 0.20	3.01 ± 0.19	67.4 ± 10.1	182.5 ± 19.8
Replica	VLN+P	2.62 ± 0.27	70.61 ± 10.45	0.1374 ± 0.0466	3.22 ± 0.21	3.28 ± 0.19	69.3 ± 10.2	140.1 ± 20.4
Replica	VLM	1.98 ± 0.21	36.77 ± 6.35	0.3883 ± 0.0431	2.40 ± 0.16	2.40 ± 0.14	36.0 ± 10.1	101.9 ± 17.0
Replica	**I-Perceive**	$\mathbf{0.90 \pm 0.14}$	$\mathbf{24.41 \pm 4.40}$	$\mathbf{0.6306 \pm 0.0430}$	$\mathbf{1.37 \pm 0.12}$	$\mathbf{1.31 \pm 0.13}$	$\mathbf{4.5 \pm 4.5}$	$\mathbf{1.3 \pm 0.0}$

Table 4: Ablation results on the Replica dataset using the complete training recipe. $\pm$ denotes the 95% bootstrap half-width. The full-model F1 value is rounded from 0.6306 in Table 3 to three decimals.

Method	e_t (m) ↓	e_r (deg) ↓	F1 ↑
I-Perceive full	$\mathbf{0.90 \pm 0.14}$	$\mathbf{24.41 \pm 4.40}$	$\mathbf{0.631 \pm 0.043}$
Replace VLM with SigLIP2	2.202 ± 0.320	50.48 ± 9.34	0.259 ± 0.058
w/o Dense Tasks	1.214 ± 0.179	30.71 ± 5.46	0.445 ± 0.054
Single Fusion Layer	1.096 ± 0.165	42.53 ± 7.04	0.353 ± 0.063
Linear Fusion	1.091 ± 0.178	31.21 ± 5.98	0.511 ± 0.055

Replica is used as the out-of-distribution benchmark. It contains 89 language-conditioned tasks in reconstructed real-world indoor environments.

E.2 Evaluation Pipeline

All quantitative methods are evaluated by a shared pipeline. The evaluator enumerates test cases, groups cases by scene to reuse voxel grids, calls the method-specific predictor, renders the predicted view through the corresponding HSSD or Replica renderer, computes metrics, and writes per-case and aggregate summaries.

E.3 Result Analysis

The main results are analyzed from per-case evaluation records. The analysis aggregates all four methods and both datasets, computes bootstrap intervals, and exports the statistics used to format the main result table.

F Detailed Main Results

See Table 3.

G Ablation Details

The ablation study compares the full reference model against four variants: replacing the VLM with SigLIP2, removing dense auxiliary tasks, using only a single fusion layer, and replacing spatial cross-attention fusion with linear fusion. All variants use the complete training recipe and are evaluated with the same metric protocol.

On Replica, the full model is clearly better across all reported metrics. Replacing the VLM with SigLIP2 causes the largest overall degradation, while removing dense auxiliary supervision and simplifying the fusion mechanism also reduce performance.

H Additional Qualitative and Closed-Loop Results

H.1 Closed-Loop Evaluation Protocol

For closed-loop evaluation, we adapt each test case to a harder partial-observability setting by removing all initial context frames. The model therefore starts from a single image, and iteration 0 reports the coverage of this initial view before any model action. At each later iteration, the method predicts a target camera pose from the current start image, instruction, and accumulated context views.

To emulate execution on a robot, the predicted pose is passed through the same lightweight execution wrapper for all compared methods, including I-Perceive and the VLM baseline. The wrapper applies collision-avoidance correction and rule-based exploration before rendering or collecting the next observation, and the resulting view is appended to the context for the following iteration. This design reflects a realistic deployment pipeline: a robot would not execute raw network outputs directly, but would combine them with safety checks and simple exploration heuristics before acquiring the next camera view.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions: a language-conditioned active perception model, a large-scale vision-language-viewpoint dataset, and evaluations on in-distribution and out-of-distribution indoor benchmarks. The reported experiments, ablations, closed-loop evaluation, and limitations qualify the scope of these claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The conclusion includes a Limitations and Future Work paragraph discussing collision detection without correction, the dependence of the training pipeline on object annotations, and the reliance on pretrained VLMs and geometric foundation models.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs; its contributions are empirical, architectural, and dataset-oriented.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The main paper and appendix describe the model architecture, training data construction, baseline designs, metrics, confidence intervals, runtime protocol, and benchmark splits. The supplemental material provide prompts, and the project page provides access to the trained model. Code, data, and model checkpoints will be released after review to allow for reproduction of the main experimental results.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The paper includes a project page and supplemental material with the prompts and access to trained model. Code, checkpoints and data will be made public after review.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The Training section and Appendix describe the data sources, sample format, generation pipeline, training schedule, mixed-precision setting, LoRA fine-tuning, loss components, baseline implementations, benchmark splits, and evaluation protocol. Additional implementation details and exact reproduction commands are provided with the code release after review.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: The main table reports mean values with 95% non-parametric bootstrap intervals for pose errors, voxel-matching F1, and judge rankings. Appendix D.5 specifies that intervals are computed over test cases with 10,000 bootstrap resamples.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The Training Details section reports the training time and GPU configuration, and Appendix D.5 reports inference runtime and VRAM usage as a function of the number of input images.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and believe the work conforms to it. The work uses existing indoor-scene datasets and author-created annotations, does not involve deceptive data collection or human-subject experiments.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper and supplemental material discuss positive impacts for more capable assistive and service robots, as well as risks from unsafe physical execution, privacy-sensitive sensing, and deployment outside validated indoor settings. The discussion highlights mitigation through motion planning, collision checking, consent-aware deployment, and respecting dataset and model usage restrictions.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, a scraped Internet dataset, or personal data. Released assets are intended for indoor active-perception research and are distributed subject to the licenses and terms of the underlying datasets and models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The paper cites the existing datasets, models, and software used in the work, including ScanNet, CA-1M, HSSD, Replica, Qwen3-VL, VGGT, NaVILA, Blender, Genesis, and related baselines. The project page and supplemental material document the corresponding licenses, access conditions, and terms of use, and our released assets respect these restrictions.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces a language-conditioned active-perception dataset, benchmark annotations, prompts, evaluation protocol, code, and model checkpoints. The appendix and supplemental material document the data format, generation and validation pipelines, benchmark annotation protocol, metrics, model usage, limitations, licenses, and release instructions.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: Although this work does not involve external crowdsourcing or traditional human-subject experiments, expert researchers constructed the evaluation set and performed the Rank-H judgments. We provide the full task definitions, training guidelines, and

screenshots of the custom annotation interfaces in Appendix A.6 (for evaluation set construction) and Appendix D.4 (for list-wise Rank-H evaluation). Compensation details are not applicable as the annotation was conducted voluntarily by the authors and internal project members.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The work does not conduct human-subject research or collect personal information from study participants. Internal annotations are used only to define target viewpoints and rank rendered candidate views for model evaluation.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The paper explicitly describes LLM/VLM usage in the core method and experiments, including the Qwen3-VL vision-language pathway, VLM-assisted instruction generation, world-model prompt translation, prompt-based baselines, and GPT-5.4 judge ranking. The prompt aliases and baseline prompt roles are documented in the appendix and supplemental prompt files.

    Guidelines:

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