

HouseTour: A Virtual Real Estate A(I)gent

Ata Çelen^{1,2} Marc Pollefeys^{1,3} Dániel Baráth^{1,4} Iro Armeni²

¹ETH Zürich ²Stanford University ³Microsoft Spatial AI Lab ⁴HUN-REN SZTAKI

<https://house-tour.github.io/>



Figure 1. **HouseTour.** Given a set of images captured in an existing 3D space and their corresponding camera poses, our method tackles the task of 3D camera trajectory and textual summary generation. We focus on generating human-like trajectories and descriptions that can be used for real estate remote video tours of properties, thus highlighting spatial characteristics such as layout, functionality, architectural features, static building elements (e.g., appliances, windows, and doors), materials, and ambiance. To support the task, we present a novel dataset with real estate video tours, descriptions, and 3D reconstructions.

Abstract

We introduce *HouseTour*, a method for spatially-aware 3D camera trajectory and natural language summary generation from a collection of images depicting an existing 3D space. Unlike existing vision-language models (VLMs), which struggle with geometric reasoning, our approach generates smooth video trajectories via a diffusion process constrained by known camera poses and integrates this information into the VLM for 3D-grounded descriptions. We synthesize the final video using 3D Gaussian splatting to render novel views along the trajectory. To support this task, we present the *HouseTour* dataset, which includes over 1,200 house-tour videos with camera poses, 3D reconstructions, and real estate descriptions. Experiments demonstrate that incorporating 3D camera trajectories into the text generation process improves performance over methods handling each task independently. We evaluate both individual and end-to-end performance, introducing a new joint metric. Our work enables automated, professional-quality video creation for real estate and touristic applications without requiring specialized expertise or equipment.

1. Introduction

Recent advances in vision-language models (VLMs) [1, 39, 59] have enabled zero-shot generalization across a wide range of real-world image, video, and text applications, bridging the gap between curated research tasks and practical scenarios. However, generating videos grounded in an existing 3D space and describing spatial qualities in unstructured language—beyond merely listing contents—remains a challenge. This task requires geometric reasoning capabilities that current models lack.

We present the novel task of spatially-aware 3D camera trajectory generation and textual summarization from a collection of images depicting an existing 3D space. This task is closely related to creating house-tour videos, a popular format on YouTube, where over 624 million videos feature real estate agents and occupants showcasing their homes. This practice surged during the COVID-19 pandemic due to travel and interaction restrictions, providing renters and prospective buyers with remote access to properties. It remains a critical tool today in the U.S. real estate market, valued at 3.43 trillion dollars. However, providing such videos

is labor-intensive, requiring expert real estate agents to visit properties with high-end videography equipment and manually craft detailed descriptions. Unlike scene captioning methods [12, 14, 48, 63], these descriptions focus on spatial layout, functionality, architectural features, static building elements (*e.g.*, appliances, windows, and doors), materials, and ambiance rather than simply enumerating furniture.

To address these challenges, we introduce HouseTour (Fig. 1), a method to automatically generate house-tour videos from a set of captured images with known camera poses. Our approach enables users—without specialized expertise or equipment—to create professional-quality videos for real estate and touristic purposes. We extend an existing VLM [59] by fine-tuning it for 3D-grounded real estate descriptions. Our method generates smooth 3D camera trajectories using a diffusion process [31] constrained by the known camera poses and integrates this information directly into the VLM to ensure text alignment with the spatial path. To visualize the results, we synthesize the final video using 3D Gaussian splatting [35] to render novel views from the generated camera poses. We design the input to our method to accommodate practical use cases where typical end-users may struggle to capture smooth videos with smartphones. Additionally, using images instead of video enhances privacy by allowing selective content capture.

To support this task, we introduce the HouseTour dataset, a curated collection of over 1,200 house tour videos featuring diverse properties ranging from apartments to multi-storey houses. Each video is accompanied by professionally captured smooth camera trajectories and real-estate-oriented textual descriptions, and generated 3D reconstructions. We compute ground-truth 3D camera pose information using off-the-shelf methods [38, 51] and manually verify all dataset information to ensure accuracy, while removing visual and textual content that may infringe on privacy. Our dataset fills a gap in existing 3D visio-linguistic datasets [2, 6, 13, 26, 32, 44, 65], where camera trajectories are tailored to the 3D reconstruction task (close-up poses to object surfaces and jerky movements) and descriptions enumerate scene objects and their in-between relationships.

Our contributions are threefold:

- We present a novel task: spatially-aware 3D camera trajectory and textual summary generation from a collection of images, with the goal to resemble house tour videos.
- We propose a new method, HouseTour, that jointly models camera trajectory and language description generation, incorporating geometric constraints in the process.
- We release the HouseTour dataset to support this task, comprising house-tour videos with 3D reconstructions, and real-estate-style textual descriptions.

The dataset, code, and trained models publicly available at house-tour.github.io.

2. Related Work

Long-Horizon Understanding and Captioning. The Video-to-Text task traditionally emphasizes generating descriptive narratives from short video clips, focusing primarily on salient actions or prominent objects [57, 66]. Despite their effectiveness in general-purpose applications, these approaches typically lack detailed spatial reasoning and long-term contextual coherence, critical for describing spatial layouts and architectural details. Recent advances [12, 48] explicitly model spatial relationships and semantic context to improve captioning. Nevertheless, these methods primarily operate on 3D scans or pre-recorded video sequences without dynamically integrating novel camera viewpoints or trajectories. This limits their utility for generating cohesive, spatially-aware narratives, *e.g.*, for navigational videos, such as real estate tours.

Recent advances in 3D vision-language tasks recognize that real-world spatial context is crucial. Works like DenseCap [34] and Scan2Cap [16] detect and describe objects within images or 3D scans, incorporating relational cues (*e.g.*, “A couch next to a table”). Furthermore, large-scale 3D datasets [11, 53, 60] have spurred research on embodied tasks (*e.g.*, vision-and-language navigation [5]) where an agent must navigate and describe the environment. For instance, frameworks like EnvDrop [54] address navigation instructions grounded in real indoor scans, partially overlapping with our objective of context-aware commentary.

Lastly, only few Video-to-Text or Multi-Image-to-Text models can effectively handle very long sequences. Most models are designed for short-horizon videos that can be represented with a limited number of visual tokens [7, 42, 58, 64]. TimeChat [50] notably extends the manageable video length by using a sliding window approach, but it lacks the foundational expertise required for architectural captioning of interior scenes. Only a handful of models meet both criteria—handling larger sequences of visual data while also incorporating domain knowledge for our task. Examples include Qwen2-VL [59] (and its latest iteration, Qwen2.5-VL [8]) as well as LLaVa-OneVision [39].

Trajectory Generation and Human-like Motion. Traditional camera trajectory estimation, fundamental to SLAM and SfM pipelines [22, 45, 51], focuses on accurately reconstructing camera poses from existing image sequences. Recent work, however, has explored generative approaches to trajectory planning, aiming to synthesize realistic camera movements that mimic human behavior or cinematic styles [18, 36, 68]. These generative approaches typically utilize learned priors from human-recorded video datasets to produce plausible trajectories, but they often lack explicit geometric grounding, resulting in potential inaccuracies or physically infeasible paths.

In robotics, existing literature often focuses on egocentric and allocentric motion as well as trajectory predic-

tion [4, 40, 47], typically aiming to forecast short-term decisions given prior environment data. More recent methods such as SceneDiffuser [29], MotionDiffuser [33], and Decision Diffuser [3] utilize diffusion-based models specifically designed for sequential decision-making tasks. In contrast, our approach, inspired by the Diffuser [31], leverages diffusion-based generative modeling explicitly conditioned on known 3D scene geometry. We formulate trajectory planning holistically rather than sequentially, enabling improved long-horizon decision-making. Such a strategy is particularly advantageous in our setting, where prior knowledge of the scene geometry is available, unlike many robotics scenarios that require simultaneous exploration and planning. While our technique draws on Diffuser’s inpainting-like paradigm for conditioning sparse observations, it mainly departs in two key ways. First, because our interaction spaces vary with different real estate layouts, we move away from learning an absolute trajectory in non-constant environments. Instead, we model human-like motion as a residual to spline interpolation, which proves more effective as a learning task. Second, we introduce a custom loss function tailored to our trajectory generation objectives, enhancing the quality of the resulting paths.

Datasets for Spatially-Aware Video Generation. Existing 3D datasets [11, 19, 62] predominantly feature environments captured using tripod systems or videos optimized for reconstruction purposes (*i.e.*, staying close to object surfaces, lacking smooth movements, and failing to capture the entirety of rooms in single frames). In recent years, several datasets have extended these to connect 3D spaces with language [2, 6, 13, 26, 32, 44, 65], supporting tasks such as object referral, scene captioning, vision-language navigation, and reasoning. However, these datasets focus on describing furniture and object relationships while overlooking broader spatial aspects such as scene layout, architectural features, materials, and ambiance. They also lack real-world video trajectories designed to observe and highlight entire spaces and are not paired with professionally crafted textual narratives. Another related dataset, RealEstate10K [67], consists of 7000 video snippets from online real estate YouTube videos, but these clips are significantly shorter than ours (1-10 seconds versus several minutes) and do not provide textual summaries of the scenes.

3. HouseTour

Given a tuple prior $(\mathcal{C}, \mathcal{I})$, where $\mathcal{C} = [c_1, c_2, \dots, c_{N_c}]$ denotes a sequence of N_c sparse, temporally ordered, and known camera poses and \mathcal{I} the corresponding RGB frames for the camera poses, our objective is to generate a trajectory τ with $N > N_c$ frames and a scene-level summary Σ that emulate the motion and language narration of a professional real estate agent when touring a property. In short, the method takes posed images as input and returns (i) a

continuous camera trajectory anchored to those observations and (ii) a descriptive summary of the scene.

The generated trajectory is represented by $\tau = [\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^N]$ and each camera pose $p_i = \{l_i, r_i\}$ along the trajectory is represented by a translation and a quaternion rotation vector, where $l_i = [x_i, y_i, z_i]$ and $r_i = a + bi + cj + dk$. We assume prior spatial knowledge of the C camera poses. Specifically, each known camera pose is given by $c_i^{t_\tau} = \{l_i, r_i\}$ and t_τ represents the temporal order of the camera pose within the trajectory. Our generated summaries Σ are conditioned on both the sparse visual observations \mathcal{I} and the spatial features $f_\theta(\mathcal{C})$ that are provided by our trajectory generation framework, *Residual Diffuser*. In other words, the generation of Σ utilizes a tri-modal model—incorporating language, vision and 3D localization—referred to as *Qwen2-VL-3D*. An overview of our method is in Figure 2. For preliminaries see Supp.

3.1. Diffusion-based Camera Trajectory Planning

Our framework extends Diffuser [31] in terms of model architecture and inpainting-style conditioning approach. Analogous to image inpainting, where an incomplete image is denoised given a mask indicating known pixels, we denoise a trajectory conditioned on a set of sparsely known camera poses. Specifically, we treat $c_{1:N_c}$ as the “mask” of known camera poses along the trajectory and denoise the rest of the trajectory around these poses. The fixed pose constraints are represented by a Dirac function [21] during the forward and reverse processes, allowing for deterministic sampling of these known camera poses at the predetermined timepoints t_τ . Notably, our approach tackles trajectory planning purely within the “observation” space, distinguishing it from Diffuser’s original formulation, which operates within a joint “observation-action” space.

Additionally, Diffuser typically addresses a static interaction space, training the model to understand and plan within a fixed environment (such as navigating a constant maze with varying start and end points). In contrast, our scenario presents dynamically changing floor layouts, effectively creating a distinct maze to traverse in each iteration. This distinction makes the direct application of the Diffuser method effectively unsuitable for our case. The variability between scenes led us to change the formulation from learning absolute trajectory representations within each scene to learning residuals that mimic humane distinctions from spline interpolation. The spline solution is a robust initial approximation for moderately sparse observations.

The generated trajectory is calculated as: $\tilde{\mathbf{p}} = \mathcal{S} + \Delta\mathbf{p}$, where \mathcal{S} is the interpolated spline for translations (SLERP for rotations [37]) between known camera poses and $\Delta\mathbf{p}$ denotes the predicted residuals. Under this formulation, for timesteps with known camera poses, we set the residual vector to zero. The reverse process for Residual Diffuser is:

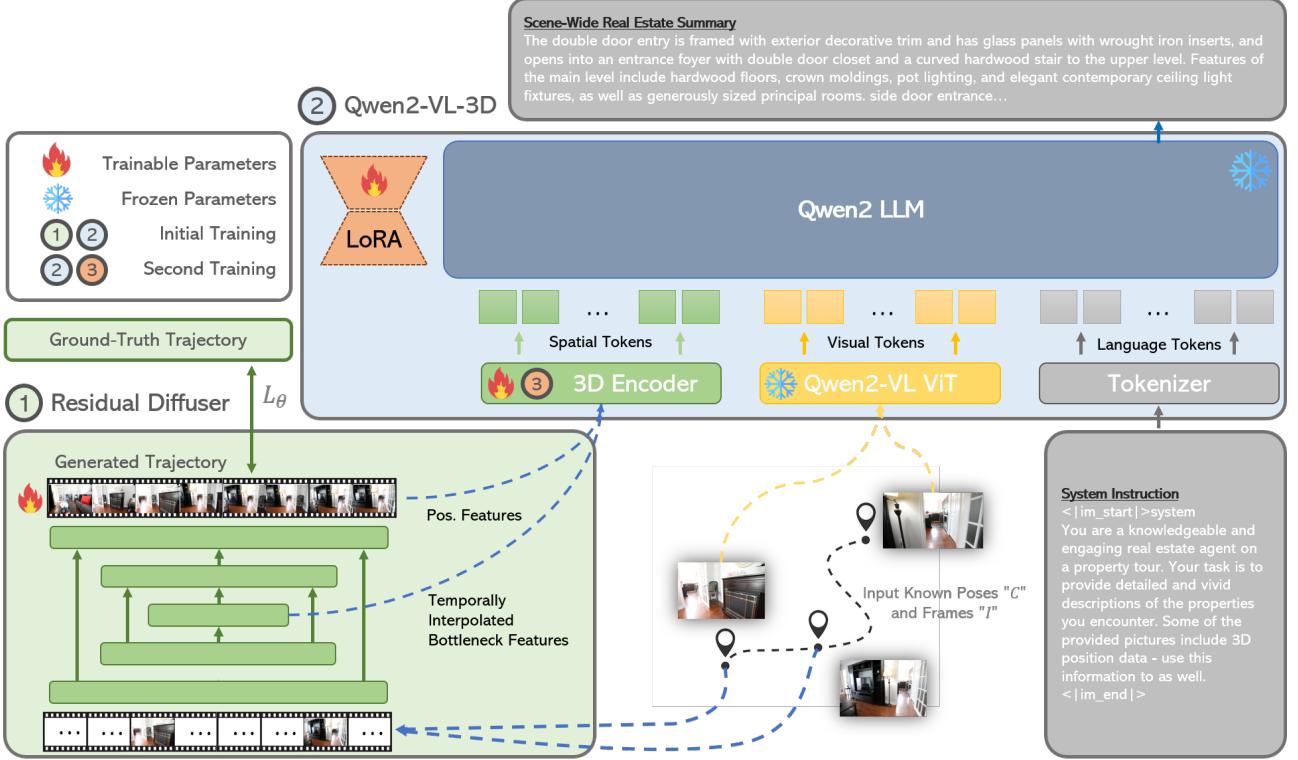


Figure 2. Pipeline Overview. Given an prior tuple $(\mathcal{C}, \mathcal{I})$ of camera poses and images, our goal is to generate scene-level real estate summaries alongside a continuous camera trajectory that emulates human navigation. The sparse camera pose observations \mathcal{C} are refined and completed by the proposed *Residual Diffuser* to obtain a smooth path. The resulting spatial features, along with the provided RGB frames \mathcal{I} , are then processed by the *Qwen2-VL-3D* model to generate a coherent real estate summary.

$$\begin{cases} \vec{0} = \delta(\mathbf{p}^i) & \text{if } i \in t_\tau \\ p_\theta(\Delta \mathbf{p}_{t-1}^i | \Delta \mathbf{p}_t^i, \mathcal{S}) = \mathcal{N}(\Delta \mathbf{p}_{t-1}^i; \mu_\theta, \Sigma_\theta) & \text{else} \end{cases} \quad (1)$$

During the forward process, we again set the residuals of known camera poses to a zero vector, while the remaining residuals are diffused using the traditional approach outlined in Equation 2 in Supp. We train a U-Net architecture built with 1D convolutions (Figure 2 (1)), as in [31], to predict the ground-truth residual from the diffused residual signal across uniformly sampled timesteps ranging from 1 to T_{diff} . Leveraging convolutions accommodates varying trajectory lengths during both training and inference. Furthermore, we significantly vary the sparseness of known camera poses during training for robust performance. The conditional diffusion process is as follows:

$$\begin{cases} \vec{0} = \delta(\mathbf{p}^i) & \text{if } i \in t_\tau \\ q(\Delta \mathbf{p}_t^i | \Delta \mathbf{p}_0^i, \mathcal{S}) = \mathcal{N}(\Delta \mathbf{p}_0^i; \sqrt{\alpha_t} x_0, (1 - \bar{\alpha}_t) I) & \text{else} \end{cases} \quad (2)$$

Trajectory Loss. In conventional denoising diffusion probabilistic model (DDPM) training, the loss is typically defined as the distance between the predicted and the ground-truth noise, a formulation that acts as a simplified vari-

tional bound [25]. However, for our application, directly minimizing the distance between the ground-truth and denoised camera poses may not be ideal for optimizing trajectories. This is partly because the ground-truth poses in our dataset were generated with 3D reconstruction and may contain inherent biases, such as denser sampling in low-texture areas and sparser sampling in high-texture regions. This could lead the model to learn undesirable patterns in order to minimize the objective. Moreover, a trajectory is a continuous function of camera poses that can be approximated by densely sampled points. We propose a loss function that adheres to these criteria and establishes a more effective distance measure between trajectories.

For a trajectory τ of length N , we define a spline segment (and SLERP for rotations) between each pair of consecutive camera poses, \mathbf{p}^i and \mathbf{p}^{i+1} . We then efficiently evaluate this spline on n uniformly sampled points along each interval using Horner's Method [27], denoting the resulting set of points as:

$$\mathcal{S}(\mathbf{p}^{i:i+1}) = (s_1^{i:i+1}, s_2^{i:i+1}, \dots, s_n^{i:i+1})$$

Next, we compute the total Euclidean length of the trajectory τ by summing the distances between successive

camera poses. We define $N_{eval} \gg N$ as the total number of evaluation points along the splines. These points are uniformly distributed along the trajectory based on Euclidean distance, with the precomputed spline values serving as indices for the evaluation points.

We compute the loss for translations using the L_2 norm on uniformly sampled dense spline points. At the same time, for rotations, we employ a geodesic loss that provides a more suitable measure of distance on the $SO(3)$ manifold. It is important to note that any residual vector added to a unit quaternion must be renormalized to ensure the quaternion remains of unit length. The resulting trajectory loss is formulated as follows:

$$\mathcal{L}_\theta = \mathbb{E}_{t, \tau, \epsilon} [\|\epsilon_{pos} - \epsilon_\theta(pos_t, t)\|^2 + d_{geo}(\epsilon_{rot}, \epsilon_\theta(rot_t, t))] \quad (3)$$

3.2. Generating Real Estate Summaries

Next, we generate real estate summaries that emulate professional house tours. These summaries emphasize the architectural features of the property rather than the objects visible in individual frames. Addressing this task requires a visual grasp of the vast architectural vocabulary; for example, differentiating among kitchen counter-top materials (quartz, ceramic, metal, wood, etc.) or recognizing various ceiling types (vaulted, cathedral, coffered, etc.). Moreover, this task demands the coherent processing of extensive, sparse multi-image data to accurately capture the complete layout of a property. Finally, we recognize that both visual and spatial information are essential to accurately locate each frame within the property and incorporate relevant spatial context into the summaries.

To this end, we integrate 3D spatial information as a third modality into a vision-language model, namely *Qwen2-VL-3D*, leveraging the spatial features provided by the Residual Diffuser to enhance summary quality (Figure 2 (2)). Our training builds on the *Qwen2-VL* [59] model, which serves as a robust foundation. Unlike many vision-language models that struggle to process large batches of images, *Qwen2-VL* provides a rich visual knowledge base that effectively connects with language and can attend to details across a large amount of visual tokens.

In the first step of training, we employ the parameter-efficient LoRA [28] fine-tuning method to train the *Qwen2-VL* model on the task of generating real-estate summaries. For this, we uniformly sample N_{frames} frames from each house tour video to serve as visual cues during multi-image training. We set N_{frames} to 96, which is chosen based on the memory requirements of training while ensuring sufficient scene coverage. This step ensures that the fine-tuned version effectively captures the language style of house tour summaries and incorporates the appropriate architectural terminology in its generated output.

In the second stage, we integrate spatial understanding

into the summary generation process. First, we add the special tokens `<|traj_start|>`, `<|traj_pad|>`, and `<|traj_end|>` to the VLM’s vocabulary. The start and end tokens define the boundaries where spatial features are inserted into the user prompt, and the pad token is replaced by the corresponding spatial token. Each spatial token is provided alongside the visual tokens from the corresponding scene location, though the visual tokens are not required to be paired with spatial tokens in return. This setup offers the flexibility to include frames with and without spatial features during training and inference.

Next, we build the adapter that transforms the raw features and absolute positional information coming from the Residual Diffuser into token representations compatible with the language processing components of the *Qwen2-VL* model. For each frame fed into the VLM, we denoise its corresponding pose p_0^i . We also get the temporally downsampled features from the bottleneck layer of the Residual Diffuser and upsample it with interpolation. We concatenate p_0^i with f_0^i , where f_0^i corresponds to the bottleneck feature from the last step of trajectory denoising. Utilizing the bottleneck layer features along with the denoised camera pose information ensures a high-level global representation of the trajectory. Lastly, the concatenated spatial features are passed through a linear layer, which maps them into the embedding space of *Qwen2-VL*’s language component. We use a single token to encode each frame’s spatial information. Implementation details are in Supp.

3.3. House Tour Videos via 3D Gaussian Splatting

To assess the visual quality of the generated trajectories, we train a Gaussian Splat [35] using all the ground-truth poses along with the reconstructed point clouds. We then render the camera poses, denoised by the Residual Diffuser, using only the sparse views. These videos are solely for visualizing the trajectories; during inference, the end user will have access only to the sparse views. While recent studies [15, 23, 61] address synthesizing scenes from sparse views, this problem falls outside the scope of our work.

4. HouseTour Dataset

We introduce the HouseTour dataset, which features scene-scale human trajectories, dense point clouds, and real estate descriptions, all derived from in-the-wild RGB real estate tour videos. The data is procured from professional real-estate agencies. Our dataset comprises 1639 videos showcasing properties ranging from condos to multi-storey apartments. Of these, 1298 videos are transcribed—half with timestamped descriptions—capturing the detailed professional language used by real estate agents use for both interior and exterior spaces. Additionally, we provide 3D reconstructions for 878 scenes, while the remaining scenes experienced partial or complete reconstruction failures. Further

details on the reconstruction pipeline, the dataset creation process, and statistics are in Supp.

5. Experiments

We evaluate HouseTour (*Residual Diffuser + Qwen2-VL-3D*) on human-like trajectory generation and multi-image scene summarization. In Section 5.2, we measure end-to-end performance with a novel joint metric and compare to the best performing baselines per task. We also provide per task analysis in Sections 5.3-5.4. All experiments are evaluated on the test set of our HouseTour Dataset.

5.1. Evaluation Metrics

Trajectory Generation. We report the recall-based metrics, $R@50cm$, $R@75cm$, and $R@1m$, which indicate the percentage of predictions with a translation error less than 50cm, 7cm, or 1m respectively. We then evaluate the trajectories using several distance- and shape-based metrics: *Euclidean Distance*, *Dynamic Time Warping* (DTW), *Hausdorff Distance*, *Fréchet Distance*, and the L_2 *Chamfer Distance*. We examine rotational quality via *Quaternion Distance* and *Geodesic Distance*; and use peak-signal-to-noise (*PSNR*) and structural similarity (*SSIM*) to evaluate the rendering performance on the 3D Gaussian splatting output.

Scene Summarization. We organize our metric selection around two primary goals: stylistic and factual alignment. For stylistic alignment, we use BLEU (B) [46], ROUGE-L (ROU-L) [43], METEOR (MTR) [9], and CIDEr (CDr) [56]. These n -gram based metrics are useful for assessing style, vocabulary, and syntax but cannot adequately capture aspects of factual correctness, coherence, or hallucinations—considerations that are critical for producing coherent summaries. To address these limitations, we adopt a commonly used preference-based evaluation approach: Bradley–Terry scores (BT). By comparing pairs of generated texts, this method captures their overall quality, including factuality and overall coherence. Building on recent approaches that use Bradley–Terry models to rank performance and employ LLMs as judges based on their own preferences [17, 24], we leverage GPT-4o [30] to compare the summaries produced by each method. In each iteration, the LLM is presented with the ground-truth summary alongside two generated summaries from different methods and is tasked with selecting the one that most closely matches the ground truth. To ensure fairness, the order that the summaries are presented to the LLM is randomized. For more details see the supplementary material.

End-To-End Performance. To measure end-to-end performance, we develop a new metric, the *Spatio-Linguistic score (SLS)*, that measures the joint performance of methods on the tasks of 3D camera trajectory and textual summary generation. More specifically, its role is to eval-

uate spatial geometry (translation and rotation) with respect to the ground truth trajectory and linguistic overlap with professionally annotated real estate summaries. It is computed as the harmonic mean of the Translation Recall at 75cm ($R@75cm$), the Rotation Score (Rot. Score), and the Bradley–Terry score (BT), and ranges from 0 to 100. Here, the Rotation Score is defined as $1 - \frac{\text{geo. dist.}}{\pi}$. The first two metrics evaluate trajectory and the latter summary generation. Together, they offer a comprehensive view of performance on the joint task. *Why R@75cm?* As shown in Table 2, our approach outperforms the baselines at larger recall thresholds but loses its comparative edge for tighter thresholds. We attribute this pattern to the increase in uncertainty as the distance to the closest known pose increases. Interpolation-based methods have high representation power where the evaluation points are close to the known poses due to the continuity and smoothness of trajectory curves. As the distance to the closest known pose grows, the uncertainty around trajectory prediction increases, causing interpolation-based methods to suffer greater performance degradation compared to our approach. Therefore, we select a *75cm* threshold as a practical balance between tightness and representational power.

5.2. End-to-End Performance

In Table 1, we compare the end-to-end performance of our method with a composed baseline. Due to the absence of a method that can solve this joint task, we devise one by using the best performing methods per task: Catmull–Rom Spline [55] for camera trajectory generation and Qwen2-VL-7B (SFT) for textual summary generation—a finetuned version of Qwen2-VL [59] for the task of generating *real-estate* summaries. HouseTour outperforms the baseline on all metrics, including the joint one.

Methods	$R@75cm \uparrow$	$\text{Rot. Score} \uparrow$	$BT \uparrow$	$SLS \uparrow$
Baseline	57.1	96.8	71.4	71.7
HouseTour	60.2	97.1	79.5	76.0

Table 1. **End-to-End Performance on 3D camera trajectory and textual summary generation.** The baseline method consists of Catmull–Rom Spline [10] and Qwen2-VL-7B (SFT), and HouseTour (ours) of Residual Diffuser and Qwen2-VL-3D.

5.3. Trajectory Generation

Table 2 presents a comparison of our *Residual Diffuser* against two interpolation-based baselines, *Linear Interpolation* and *Catmull–Rom Splines*, for trajectory generation. For rotational data, we adjust the Linear Interpolation baseline to linearly interpolate quaternions, while the Catmull–Rom one uses spherical linear interpolation (SLERP). We construct our spline-based baseline using the Catmull–

Methods	Translation							Rotation		Rendering	
	$R@50cm \uparrow$	$R@Im \uparrow$	<i>Euclidean Dist.</i> \downarrow	<i>DTW</i> \downarrow	<i>Hausdorff Dist.</i> \downarrow	<i>Frechet Dist.</i> \downarrow	<i>Chamfer Dist.</i> \downarrow	<i>Quaternion Dist.</i> \downarrow	<i>Geodesic Dist.</i> \downarrow	<i>PSNR</i> \uparrow	<i>SSIM</i> \uparrow
Lin. Interp.	41.2%	59.8%	145.8	192.1	118.7	126.7	109.5	0.0432	0.20	14.20	0.557
Catmull-Rom	45.9%	64.7%	106.2	146.3	89.3	95.8	83.4	0.0079	0.10	14.22	0.557
Residual Diffuser (Ours)	46.2%	69.4%	73.9	128.8	76.3	81.2	75.5	0.0073	0.09	14.24	0.556

Table 2. **Spatial Trajectory Generation Performance.** We compare our generated trajectories to interpolation-based baselines that do not account for “human-like” motion. *Translation* performance is reported in **cm**, *Quaternion Distance* is measured as the Euclidean distance between unit quaternions, and *Geodesic Distance* is reported in **radians**.

Rom spline because, unlike other spline variations such as the B-Spline[20], it passes through all the control points.

In order to evaluate methods on cases with varying number of frames that range from sparser to denser coverage of the property, we vary the frequency of the known camera poses between every 5th and 15th frame of the ground-truth video. Our method outperforms baselines across all translation and rotation metrics. In particular, the highest recall for $R@Im$ indicates that our method incurs fewer “large-errors”—errors substantial enough to be considered major misalignments—than the interpolation methods. In addition, significantly reduced distance metrics in translation, along with lower quaternion and geodesic distances in rotation ,imply more accurate pose generation. For rendering-based metrics, PSNR and SSIM remain comparable to baseline values; these metrics are less sensitive to slight pose differences—especially when the rendered image content (*e.g.*, lighting, texture, and overall scene structure) is preserved and, thus, are less informative on the quality of the final video.

These results confirm that incorporating data-driven diffusion into trajectory prediction better captures human-like motion tendencies and yields more robust and precise results than purely geometric interpolation. Figure 3 further supports this, showcasing more human-like and smooth camera trajectories. For more results see supp.

5.4. Scene Summarization

In Table 3, we compare our *Qwen2-VL-3D* method with zero-shot and supervised fine-tuned variants of foundation VLMs. Since many state-of-the-art VLMs struggle when faced with a large set of input images, thus limiting their scene-level understanding, we use *LLaVa-OneVision-7b* [39] and *Qwen2-VL-7b* [59] as zero-shot baselines, which have a demonstrated robustness to larger multi-image inputs. We also finetune *Qwen2-VL-7b* on our dataset without any interactions with the trajectory generation to learn the real-estate language style (*Qwen2-VL-7b (SFT)*).

By incorporating 3D-aware information, the resulting multi-image scene summaries are noticeably more coherent and descriptive, as evidenced by improvements across most n -gram metrics and preference-based evaluations. Fine-tuning also has a pronounced effect on preference scores, as the fine-tuned model decisively outperforms its zero-shot counterpart. While a performance gain is unsurprising, the



Figure 3. **Trajectory Visualization Within the 3D Reconstructions (top view).** Our method, Residual Diffuser, achieves a more human-like and smooth trajectory than the baseline Catmull-Rom Spline. **Black:** Ground-Truth, **Green:** Residual Diffuser and **Red:** Catmull-Rom Spline.

Methods	Multi-image Scene Summarization							
	B1 \uparrow	B2 \uparrow	B3 \uparrow	B4 \uparrow	ROU-L \uparrow	MTR \uparrow	CDr \uparrow	BT \uparrow
LLaVa-OneVision-7b	0.259	0.128	0.055	0.024	0.189	0.109	0.001	0.04
Qwen2-VL-7B	0.272	0.137	0.064	0.029	0.195	0.13	0.005	0.04
Qwen2-VL-7B (SFT)	0.363	0.220	0.126	0.070	0.231	0.175	0.026	0.71
Qwen2-VL-3D (Ours)	0.433	0.264	0.154	0.090	0.24	0.193	0.021	0.79

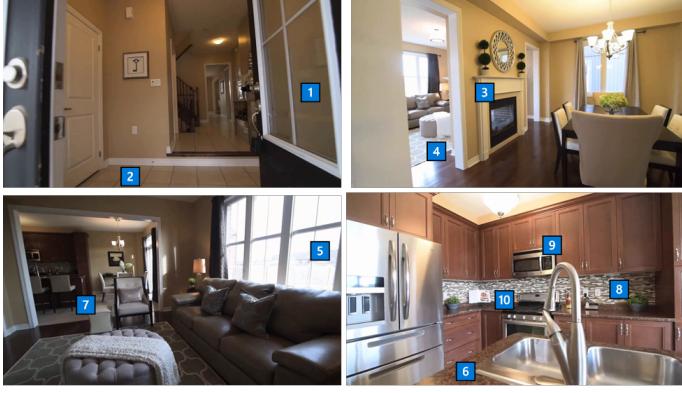
Table 3. **Linguistic Summary Evaluation.** We compare *Qwen2-VL-3D* against other foundation VLM models. (*SFT*) denotes that the model has been fine-tuned on our dataset without any input from the trajectory generation.

substantial gap underscores the limitations of off-the-shelf VLMs for generating more applied language styles.

Figure 4 shows an example of the summaries generated by our method. As shown, HouseTour can accurately identify the layout, materials, architectural elements, and ambiance. For more results, including a user study, see supp.

5.5. Ablation Studies

Table 4 examines how the frequency of observations impacts trajectory generation. In this experiment, observations are uniformly sampled every 5th, 10th, and 15th frame, and the corresponding performance metrics are reported. As anticipated, sparser sampling provides less information, which results in higher errors for all methods. Our findings indi-



Qwen2-VL-3D

The front door is double with decorative glass **1** and opens into a ceramic-tiled foyer **2** and a main level finished with hardwood and ceramic floors. The main level also features a gas fireplace in the formal dining area **3**, which has a walk-out to the backyard, and is open to the adjacent living room, a bright **4**, naturally well-lit room with a large picture window overlooking the front yard **5**. A french door offers privacy between these two rooms. The adjacent breakfast area has a sliding door walk-out to the backyard and overlooks the kitchen. This beautiful and functional workspace features granite countertops **6** and breakfast seating at the peninsula **7**, a tiled backsplash **8**, abundant cabinetry storage, and stainless steel appliances including a built-in microwave range hood **9**, oven with smooth cooktop **10**, dishwasher, fridge, and an upright freezer...

Figure 4. **Qualitative Results for Scene-Level Summary Generation.** Our method creates accurate real-estate descriptions that capture the architectural style and elements of the space. We showcase an example of our summary generation, including sampled images of the space and a mapping between generated text and spatial elements to facilitate the reader.

Pose Frequency (every N frame)	Methods	Translation (cm)						Rotation	
		Euclidean Dist. ↓	DTW ↓	Hausdorff Dist. ↓	Frechet Dist. ↓	Chamfer Dist. ↓	Quaternion Dist. ↓	Geodesic Err. ↓	
5 th	Lin. Interp.	87.5	151.6	95.4	99.9	89.6	0.0368	0.19	
	Catmull-Rom	56.0	103.5	64.2	67.4	61.5	0.0053	0.07	
	Residual Diffuser (Ours)	41.5	96.3	57.8	60.3	58.4	0.0052	0.07	
10 th	Lin. Interp.	268.3	278.1	166.3	182.2	149.5	0.0714	0.33	
	Catmull-Rom	219.0	238.4	140.6	154.9	128.6	0.0133	0.14	
	Residual Diffuser (Ours)	151.1	197.3	114.8	125.2	110.7	0.0131	0.13	
15 th	Lin. Interp.	559.9	414.9	235.9	272.9	202.4	0.1059	0.46	
	Catmull-Rom	510.9	388.5	217.3	254.7	189.6	0.0477	0.28	
	Residual Diffuser (Ours)	371.2	321.1	177.1	206.6	163.9	0.0471	0.27	

Table 4. **Ablation Study** on the impact on trajectory generation performance when varying the frequency of the known camera poses.

cate that our method performs reliably in both dense and sparse scenarios, establishing it as an overall better choice. Notably, the table reveals that at a moderate sampling frequency (every 10th frame) our method works the best. The Euclidean distance error is reduced by 32% compared to the closest baseline with moderate sampling frequency, while reductions of approximately 28% are observed when sampling every 5th and 15th frames.

our method achieves in the absence of 3D data suggests that the model may, to some extent, be learning to “localize” the images within the scene during training. As a reminder, during training, Qwen2-VL-3D receives as input data with and without 3D positioning.

Results on out-of-distribution scenes are in the supp.

6. Conclusion

We introduced HouseTour, a method for spatially-aware 3D camera trajectory and textual summary generation from a collection of images. Our approach addresses the limitations of existing VLMs by incorporating geometric reasoning through a diffusion process, enabling the creation of realistic house-tour videos without specialized equipment or expertise. We also presented the HouseTour dataset, which uniquely combines real-world house-tour videos, accurate 3D reconstructions, and professionally crafted textual descriptions, facilitating comprehensive evaluation of spatially-aware video methods. Future works can explore incorporating information from VLMs to jointly guide the trajectory diffusion process and the development of Gaussian splatting methods that can fill the gaps between images without generating non-existing content.

Methods	w/o 3D Pos.	w/ 3D Pos.
Qwen2-VL-7B (SFT)	44%	32.5%
Qwen2-VL-3D (Ours)	56%	67.5%

Table 5. **Ablation Study** on the impact of 3D positional information on the summarization performance, using Bradley–Terry probabilities. The values denote the **Win Percentages**.

In Table 5, we investigate the effectiveness of the presence of 3D information for the summary generation task. For this purpose, we partition our test set into scenes that include 3D information and those that do not. The results indicate that when 3D information is available, *Qwen2-VL-3D* is significantly favored over the finetuned *Qwen2-VL-7B* model, highlighting the value of spatial priors in summary generation. Additionally, the performance boost that

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1
- [2] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. 2, 3
- [3] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022. 3
- [4] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 3
- [5] 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. 2
- [6] 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. 2, 3
- [7] Kirolos Atallah, Xiaoqian Shen, Eslam Abdelrahman, Esam Sleiman, Deyao Zhu, Jian Ding, and Mohamed Elhoseiny. Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens. *arXiv preprint arXiv:2404.03413*, 2024. 2
- [8] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2
- [9] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 6
- [10] Edwin Catmull and Raphael Rom. A class of local interpolating splines. In *Computer aided geometric design*, pages 317–326. Elsevier, 1974. 6
- [11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgbd data in indoor environments. 2017. 2, 3
- [12] 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. 2
- [13] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgbd scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 2, 3
- [14] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11124–11133, 2023. 2
- [15] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvs-plat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2025. 5
- [16] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgbd scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021. 2
- [17] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024. 6
- [18] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. 27(8):2197–2218, 2008. 2
- [19] 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. 3
- [20] Carl De Boor. On calculating with b-splines. *Journal of Approximation theory*, 6(1):50–62, 1972. 7
- [21] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. Number 27. Oxford university press, 1981. 3
- [22] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. pages 611–625. IEEE, 2017. 2
- [23] Zhiwen Fan, Kairun Wen, Wenyan Cong, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantspat: Sparse-view sfm-free gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309*, 2024. 5
- [24] Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*, 2023. 6
- [25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4

- [26] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9202–9212, 2023. 2, 3
- [27] William George Horner. Xxi. a new method of solving numerical equations of all orders, by continuous approximation. *Philosophical Transactions of the Royal Society of London*, (109):308–335, 1819. 4
- [28] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 5
- [29] Siyuan Huang, Zan Wang, Puha Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023. 3
- [30] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6, 18
- [31] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 2, 3, 4
- [32] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Scenefeverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pages 289–310. Springer, 2024. 2, 3
- [33] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9644–9653, 2023. 3
- [34] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016. 2
- [35] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. pages 139–1, 2023. 2, 5
- [36] Jing Yu Koh, Honglak Lee, Yinfei Yang, Jason Baldridge, and Peter Anderson. Pathdreamer: A world model for indoor navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14738–14748, 2021. 2
- [37] Verena Elisabeth Kremer. Quaternions and slerp. In *Embolets dfki. de/doc/seminar ca/Kremer Quaternions. pdf*, 2008. 3
- [38] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. pages 71–91, 2024. 2, 12, 18
- [39] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 1, 2, 7
- [40] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in neural information processing systems*, 33:19783–19794, 2020. 3
- [41] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 12
- [42] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhui Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2
- [43] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 6
- [44] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sq3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 2, 3
- [45] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. pages 1147–1163. IEEE, 2015. 2
- [46] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 6
- [47] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric future localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4697–4705, 2016. 3
- [48] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025. 2
- [49] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023. 12
- [50] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024. 2
- [51] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 2, 12
- [52] Johannes L Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20–24, 2016, Revised Selected Papers, Part I* 13, pages 321–337. Springer, 2017. 12
- [53] 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. 2
- [54] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. 2019. 2
- [55] Christopher Twigg. Catmull-rom splines. *Computer*, 41(6): 4–6, 2003. 6
- [56] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 6
- [57] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015. 2
- [58] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 2
- [59] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1, 2, 5, 6, 7
- [60] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 2
- [61] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 5
- [62] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4927–4936, 2023. 3
- [63] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8573, 2022. 2
- [64] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2
- [65] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023. 2, 3
- [66] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018. 2
- [67] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 3
- [68] Zhenghong Zhou, Jie An, and Jiebo Luo. Latent-reframe: Enabling camera control for video diffusion model without training. *arXiv preprint arXiv:2412.06029*, 2024. 2

HouseTour: A Virtual Real Estate A(I)gent: Appendix

Abstract

In the supplementary material, we provide:

1. Details on the HouseTour dataset (Sec. 7)
2. More qualitative results (Sec. 8)
3. User Evaluation of Text Generation (Sec. 9)
4. Implementation details (Sec. 10)
5. Out-of-Distribution Scenes (Sec. 11)
6. Details on the Metric Scale Evaluation (Sec. 12)
7. Details on the Bradley-Terry evaluation (Sec. 13)
8. Recall Plots for the trajectory generation (Sec. 14)
9. Preliminary information on diffusion (Sec. 15)

7. HouseTour Dataset

7.1. Creation Details

At the start of our reconstruction pipeline, we select a subset of video frames for use in the process. Since our videos range from a few minutes to 15 minutes in length, using all frames would be computationally impractical. Our objective is to choose a minimal yet effective set of keyframes that maintain significant overlap to ensure accurate reconstruction. To accomplish this, we use an algorithmic approach that evaluates factors such as optical flow and key-point matches between consecutive views.

We then trim the beginning and end of the selected keyframe sequence to remove frames that induce spatial jumps—such as exterior drone shots, which could largely affect the 3D reconstruction. To accomplish this, we use an off-the-shelf vision-language model, BLIP2 [41], to classify the keyframes at the sequence boundaries as exterior shots.

For the 3D scene reconstruction, we employ the COLMAP [51] structure-from-motion approach. We generate image pairs from the keyframes leveraging their inherent sequential order and augment these pairs with additional ones identified through traditional image retrieval techniques [52] to simulate loop closure during the reconstruction process. We perform dense 2D-to-2D matching between paired frames using Mast3r [38] and subsequently map them with COLMAP after geometrically verifying the pixel correspondences.

Lastly, if a video contains speech, we use Whisper [49] to extract the transcriptions along with their timestamps. If there is no speech, we obtain video descriptions as they stylistically align with the transcribed scene descriptions. To protect privacy, we employ GPT-4o to automatically filter out sensitive details such as addresses, personal names, and phone numbers from the video transcriptions. Additionally, we manually edit sections that mention neighbor-

hood information or amenities, since such details are not visually represented in the videos.

7.2. Dataset Statistics

3D Reconstruction and Scene-Level Descriptions. The 3D reconstruction process for a single scene can take up to 40 hours, depending on the number of keyframes extracted from the videos. Reconstructing over 1,600 scenes may require 3 to 4 months of CPU time. Additionally, the keyframe extraction and matching processes are GPU-intensive. To manage this, we employ a high-performance computing cluster for dataset acquisition. Collecting the dataset typically takes about 7 days when using a job array with 20 parallel jobs. Each job requires 48GB of CPU memory and a 32GB NVIDIA Tesla V100 GPU. To handle extremely long runtimes, we limit each job to a maximum duration of 40 hours.

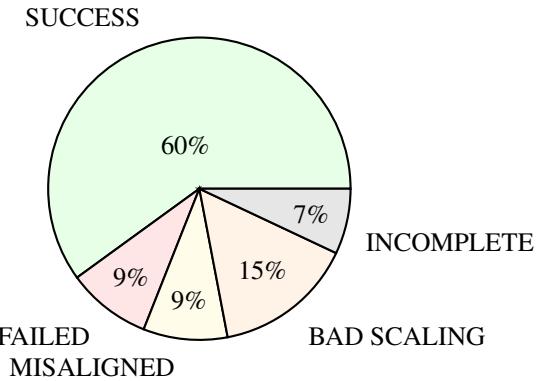


Table 6. Breakdown of the reconstruction outcomes.

Table 6 provides a detailed analysis of our pipeline’s reconstruction performance. Scenes marked as *SUCCESS* have been reconstructed successfully. *INCOMPLETE* refers to reconstructions that failed to register the entire scene due to tracking loss, often caused by textureless areas or the lack of complete covisibility graph data. *MISALIGNED* scenes have errors in reconstruction leading to incorrect rotations of some scene portions. Scenes labeled with *BAD SCALING* have errors resulting in discrepancies in scale across parts of the output model. We classify scenes as *FAILED* if they exhibit multiple of the aforementioned issues or have significant errors in the final model.

Each reconstructed scene includes a dense point cloud with more than one million vertices, 2D-to-3D correspondences, and outputs from COLMAP [51], including images, camera data, and 3D points in binary files. Additionally, the selected keyframes and their timestamps are listed. For scenes with descriptions, we provide either a CSV file containing text and timestamp information or a plain text file with the description.

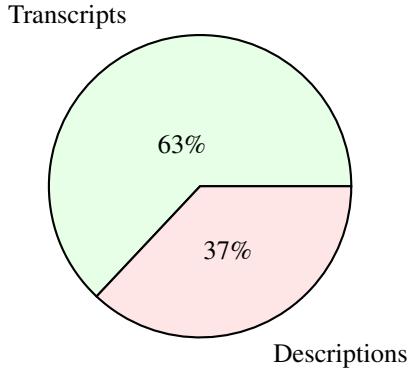


Table 7. Distribution of summary types in the HouseTour dataset. Transcripts include timestamped information.

Furthermore, we retrieved real estate descriptions for 1,298 out of 1,639 scenes. Of these, 813 descriptions were transcribed using the Whisper model, while the remaining ones were sourced from descriptions. We include all scenes in the dataset, even if they do not have 3D reconstructions or summaries. We utilize the scenes without 3D reconstruction as additional data to the VLM and those without summaries as additional data for the camera trajectory generation.

Contextual Analysis of the Dataset Descriptions. The videos primarily feature tours of detached houses rather than flats or apartments, with most properties located in a city within a developed, financially advanced country and its surrounding suburbs (we refrain from disclosing the location for privacy reasons). The showcased real estate is predominantly high-end, often including luxurious features. As shown in Table 8, these homes typically have three to five, with four being the most common. A substantial portion are multi-storey, usually with two or three floors, and many include outdoor spaces such as front or backyards. The interior design is largely modern or contemporary. It is important to note that our dataset is biased toward upscale properties in a specific region and does not capture the full diversity of global architectural styles.

Linguistic Analysis of the Dataset Descriptions. Table 9 presents an analysis of phrases based on constituency within the extracted descriptions. The constituency-based analysis reveals that the most common adjectives in the descriptions fall into categories such as scalar (e.g., “large”, “double”, and “ample”), directional (e.g., “main”, “upper”, and “lower”), and conceptual (e.g., “natural”, “open”, and “spacious”), all of which describe aspects of interior spaces. Additionally, there are adjectives related to material information, like “stainless” and “ceramic”.

When analyzing the most frequent adverbs in the video descriptions, the word “fully” stands out as the most fre-

quent adverb, occurring approximately 80 times, which may suggest these videos often showcase homes that are fully equipped or fully furnished. Following “fully,” we see a significant mention of “beautifully,” which indicates a focus on aesthetic appeal, highlighting beautifully designed spaces. Other adverbs like “away”, “incredibly”, and “graciously”, suggest descriptions of location, extraordinary features, or hospitality aspects. Adverbs such as “professionally”, “highly”, and “conveniently” point towards emphasizing quality and ease of living. The occurrence of these specific adverbs suggests that house tour videos prioritize aspects such as completeness, beauty, functionality, and unique features to attract potential buyers or viewers.

Lastly, the verbs within the descriptions give further linguistic cues on descriptions as a whole. The verb “features” appears most frequently, over 400 times, indicating a focus on highlighting key aspects or amenities of properties. Verbs like “finished”, “built”, and “found” suggest emphasis on quality, construction, and location. Words such as “offered”, “opens”, and “overlooking” reflect the dynamic aspects and views these properties provide. Other verbs like “leads”, “includes”, and “showcases” emphasize navigation, inclusivity, and presentation within the space. The use of “situated”, “covered”, and “updated” implies a focus on positioning, protection, and modern enhancements. Overall, these verbs underline the importance of showcasing distinctive features and conveying a sense of completeness and modernization in house tours.

Named-Entity Based Analysis of the Dataset Descriptions. When analyzing the first bar plot in Table 10 the *rooms and areas in a house* entities in video descriptions, the “kitchen” emerges as the most highlighted space, with nearly 250 mentions, underscoring its role as a central and significant feature in homes. The “master bedroom” follows closely, reflecting its importance in personal comfort and privacy. Other commonly referenced areas include the “family room,” “front foyer,” and “living room,” which are key spaces for gathering and welcoming guests. The plot also shows notable mentions of functional areas like the “laundry room” and “dining room,” indicating their relevance in daily living. The presence of terms like “lower level” and “breakfast area” suggests an emphasis on specific sections or niches that add value to the property’s layout. Overall, the focus on both communal and private spaces highlights a balanced presentation of essential living areas and unique features in house tours.

The second plot shows the frequency of *objects in a room*. “Stainless steel appliances” are the most prominent, appearing over 60 times, emphasizing the modern and desirable features of kitchens. Following closely are “granite countertops” and “gas fireplaces,” indicating a

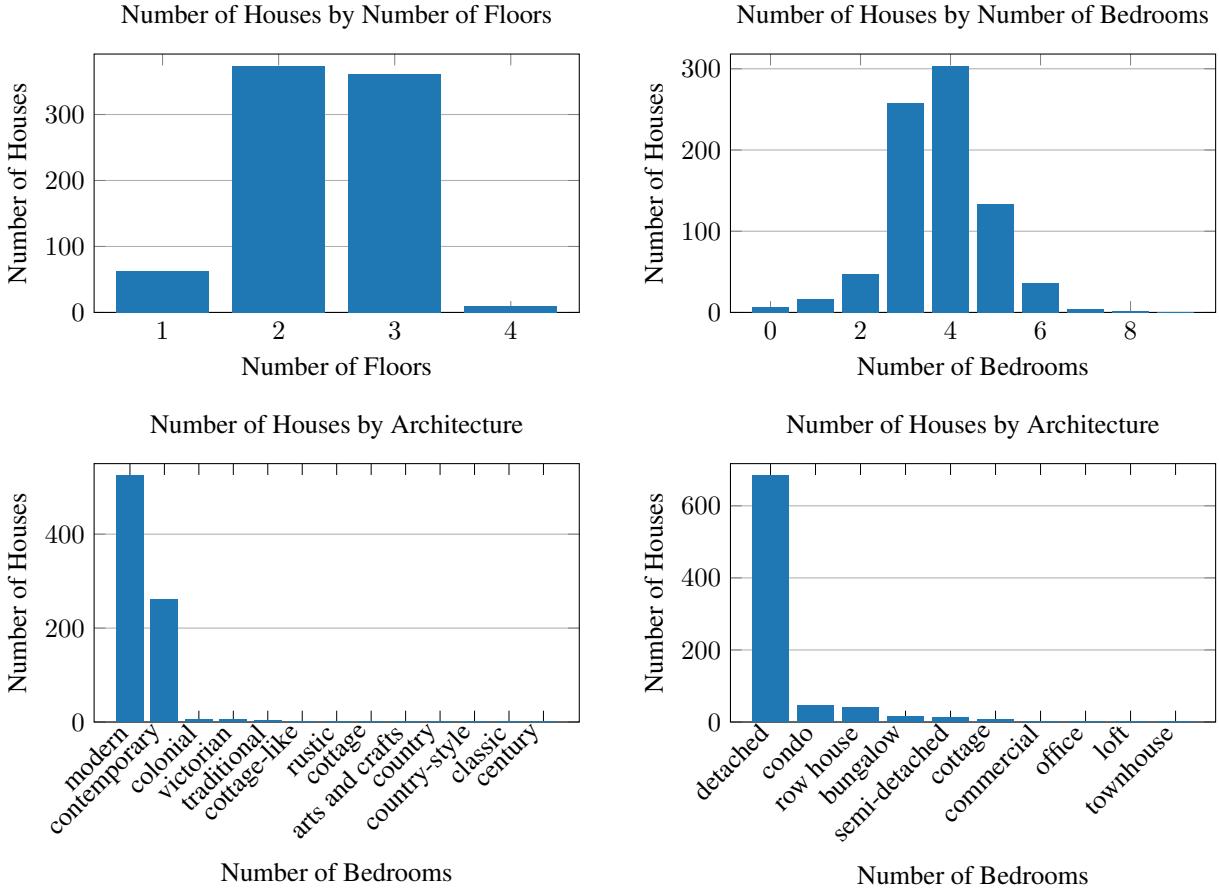


Table 8. **Contextual dataset statistics.** As shown in the data, most of the properties are modern, detached, with 2-3 floors, and 3-4 bedrooms.

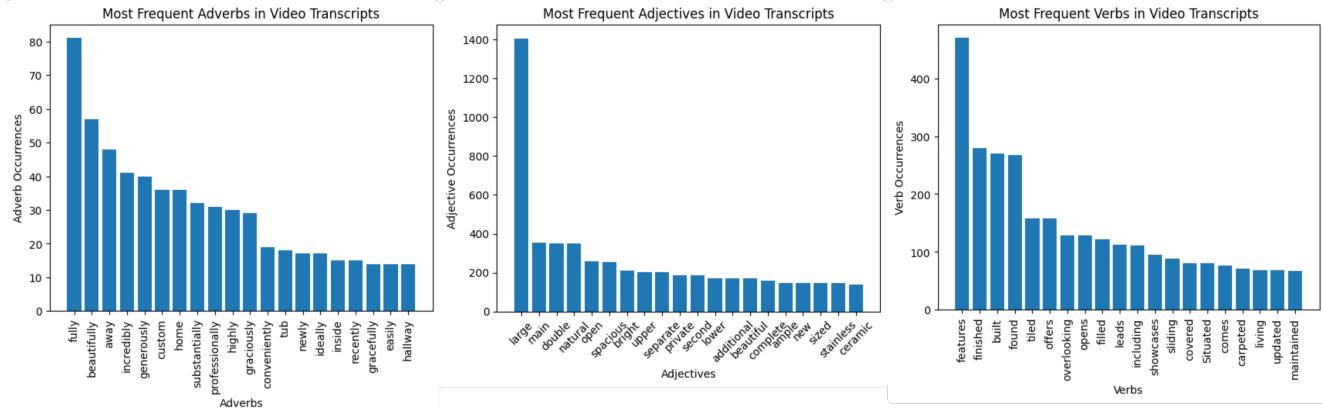


Table 9. **Constituency-based word frequency statistics.**

focus on quality materials and cozy elements. Decorative items like “chandeliers” highlight style and luxury in living spaces. Functional features such as “undermount sinks,” “fridges,” and “microwaves” underscore practical aspects of home living. The presence of “pot lights” and

“California shutters” suggests attention to lighting and window treatments. Additional mentions of “stoves,” “dishwashers,” and “tile backsplashes” reflect both essential and aesthetic components of kitchens.

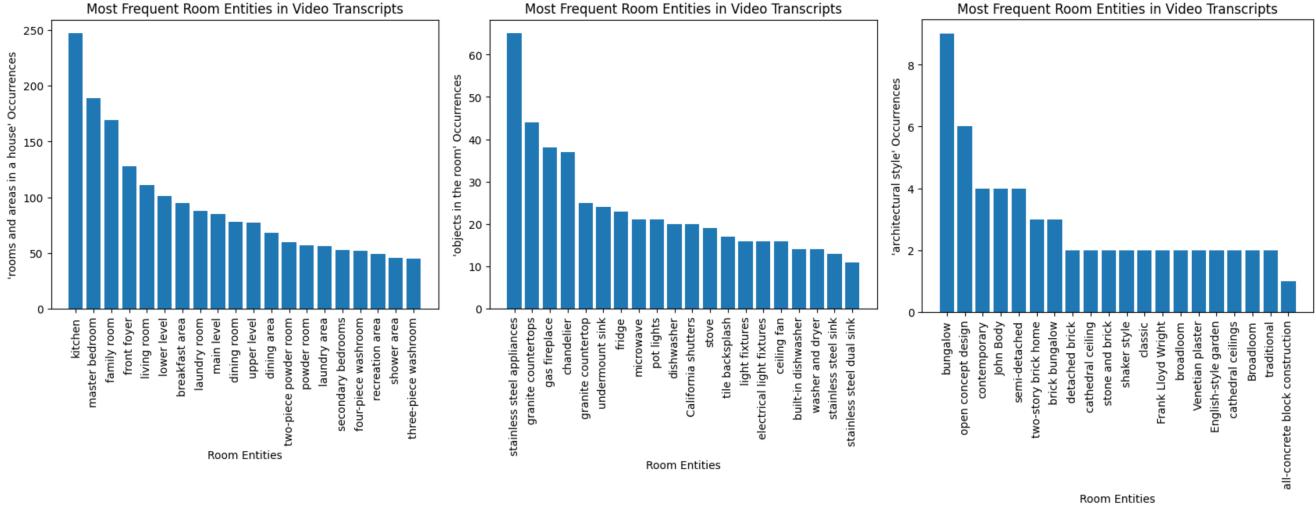


Table 10. Named Entity Frequency Statistics

The third and the last plot visualizes the *architectural style* found within the real estates. “Bungalow” emerges as the most frequently cited style, followed by “open concept design,” indicating a preference for single-storey (usually plus the lower floor) living and spacious, flowing interiors. Styles like “contemporary” and “semi-detached” also receive notable mentions, highlighting a mix of modern and practical designs. The inclusion of “two-storey brick home” and “brick bungalow” suggests an appreciation for classic, durable architecture. Terms like “cathedral ceiling” and “stone and brick” emphasize distinct design features and materials. Styles attributed to iconic architectural figures like “Frank Lloyd Wright” reflect a nod to renowned architectural influences. The mentions of “English-style garden” and “Venetian plaster” suggest an interest in incorporating thematic and textural elements. Overall, this plot underscores a variety of styles, balancing traditional and modern influences in architectural preferences.

Captures from the HouseTour Dataset. In Figure 5, we provide some captures from the dense point clouds of our HouseTour dataset.

8. Additional Qualitative Results

Figure 6 provides additional qualitative results on trajectory generation from *Residual Diffuser*, and Figure 7 on scene-level summary generation from *Qwen2-VL-3D*.

9. User Evaluation of Text Generation

We conducted a single-blind user study in which three different participants evaluated generated descriptions for 20 different scenes. Across all assessed categories (Tab. 11),

Methods	Lay.	Mat.	Fix.	Amb.	Ove.
Qwen2-VL-7B (SFT)	6.1	6.0	5.9	5.9	6.0
Qwen2-VL-3D (Ours)	7.0	7.3	7.2	6.9	7.3

Table 11. **User Study** evaluating text generation quality across five categories: **Layout** (Lay.), **Material** (Mat.), **Fixture** (Fix.), **Ambience** (Amb.), and **Overall** (Ove.). Score range: [0,10].

users consistently preferred our method. The results suggest that incorporating 3D positional information significantly enhances the perceived quality of the generated text. In our experimental setup, each participant watched a house tour video accompanied by two textual descriptions: one generated by a standard fine-tuned (SFT) baseline and the other by our Qwen2-VL-3D model. For every video, the order of the two descriptions was randomized to eliminate order bias. Six participants rated each description on a 0–10 scale, where 0 indicates no correspondence with the video and 10 indicates perfect alignment. The two descriptions were shown simultaneously, allowing for both absolute and comparative assessments. The definitions for each grading category is as follows:

Layout. Does the description demonstrate a good understanding of the spatial organization of the room? Consider references to walls, doors, windows, room shapes, and how space is structured.

Material. Does the description accurately capture the materials and finishes in the scene? Look for details about surface textures, colors, or types of materials (e.g., wood, glass, concrete).

Fixture. How well does the description mention relevant built-in or fixed elements? Examples include lighting fixtures, sinks, cabinetry, or any permanent installations.



Figure 5. **Gallery of sample captures from the HouseToura dataset.** The showcased regions originate from the 3D reconstructions.

Ambiance. Does the description convey the overall mood or atmosphere of the space? Consider lighting, color tone, and emotional or sensory impressions.

Overall. An overall grading on the quality of the description.

10. Implementation Details

Residual Diffuser. We employ a lightweight U-Net architecture with two downsampling and two upsampling layers, changing the trajectory length by a factor of two at each

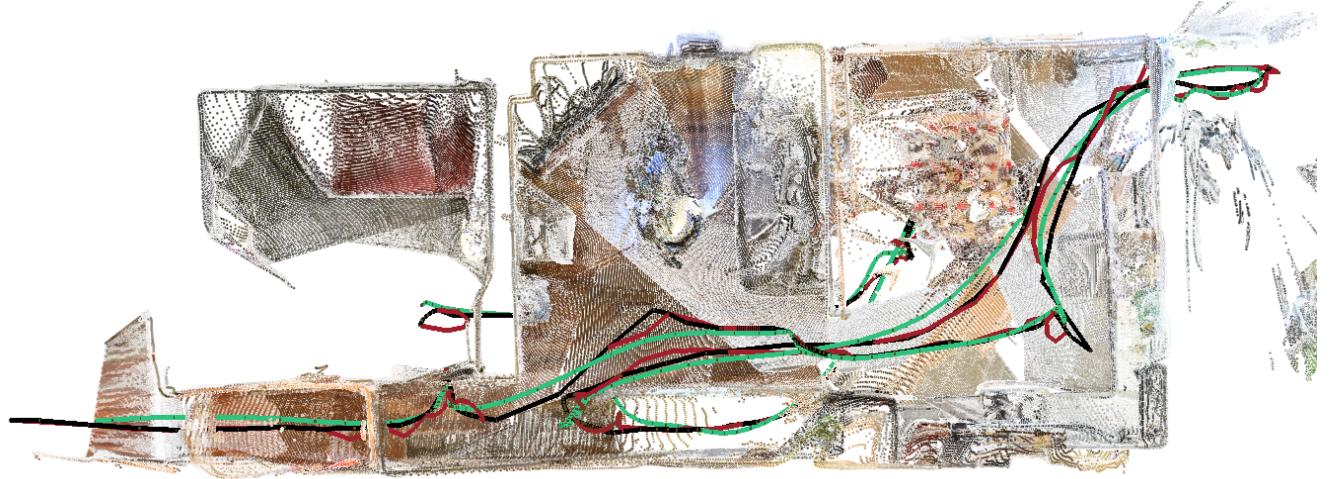


Figure 6. **Trajectory Visualization Within the 3D Reconstructions (top view).** Our method, Residual Diffuser, achieves a more human-like and smooth trajectory than the baseline Catmull-Rom Spline. **Black:** Ground-Truth, **Green:** Residual Diffuser and **Red:** Catmull-Rom Spline.

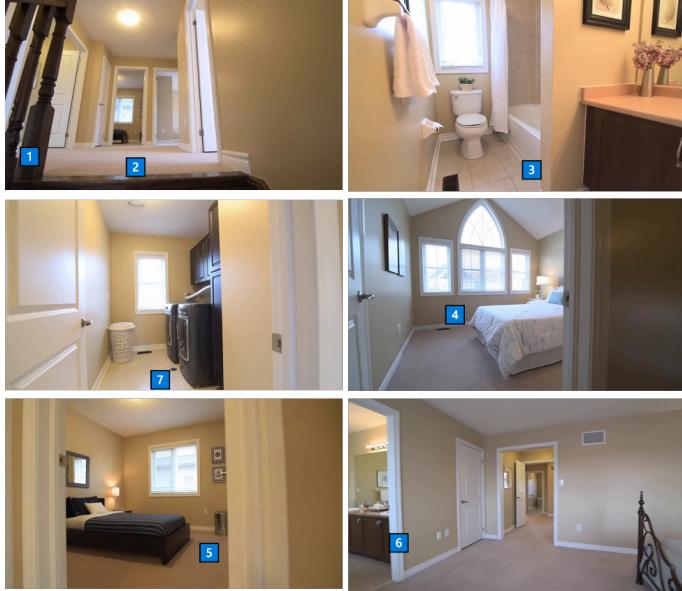


Figure 7. Further qualitative results for scene-level summary generation

step. The model is trained on a single *NVIDIA GeForce RTX 2080 (8GB)* GPU for 30K iterations with a batch size of 1 and gradient accumulation every 8th iteration, using a learning rate of 5×10^{-6} . To improve generalization, we randomly vary the number of sparse observations per training step, ensuring at least one observation every 20 frames.

Qwen2-VL-3D. We adopt a two-step training strategy for Qwen2-VL-3D. In the first phase, we LoRA-finetune Qwen2-VL on a single *NVIDIA A100 (80GB)* GPU for 20 epochs, with early stopping after the 8th epoch. We use the AdamW optimizer with a learning rate of 2×10^{-4} , cosine

scheduling, and a warm-up ratio of 0.03. The effective batch size is 1, with gradient accumulation every 8th iteration. We train the VLM in *bfloat16* precision and apply gradient clipping at a maximum norm of 0.3 to prevent overflow. For the LoRA adapter, we specify both rank and alpha as 64, along with a 0.05 dropout rate. Adapters are added only to the Q and V weights of the attention layers.

Inference Times. All benchmarks were run on a 40 GB NVIDIA A100 GPU. The lightweight Residual Diffuser generates one trajectory in 0.23 ± 0.04 s on average, whereas Qwen2-VL-3D requires 14.9 ± 5.6 s to produce a single tex-

tual response.

11. Out-Of-Distribution Scenes

To evaluate the generalization capabilities of our method, we test it on two out-of-distribution (OOD) scenarios: (1) a single office room from the ScanNet++ dataset and (2) an online drone-view exterior shot of a property (Fig. 9). For these examples, we adjust the decoding temperature to $T = 0.3$ (compared to $T = 1.0$ used on the in-distribution HouseTour dataset). The temperature scaling sharpens the softmax distribution and, in practice, lets the weak image evidence outweigh generic language priors, dampening the language priors learned during training (e.g., persistent mention of bedrooms or kitchens in training samples); hence curtailing hallucinations that do not align with the visual evidence. While the tweak is effective, it also showcases the narrower visual-text diversity of the training data; expanding the dataset should further improve OOD robustness and lessen the reliance on temperature scaling. Note that, in both cases, the generated text is featuring the learned language style. As shown in Figure 8, the trajectory we generate (shown in blue) exhibits smoother and more natural motion compared to linear interpolation (red) and Catmull-Rom splines (green).

12. Evaluation In Metric Scale

In our experiments, we report all evaluation results in metric scale, even though our 3D reconstructions are not inherently metric. We adjust the scale of the reconstructions for evaluation so that errors in both large and small scenes are handled consistently, avoiding the distortion that arises from varying relative scales.

To achieve this, we use the same metric depth model employed by Mast3r [38] for metric alignment of camera pose pairs. Specifically, we uniformly sample 20 pairs of sequential keyframes from each reconstruction and measure the Euclidean distances between their poses as estimated by the Mast3r model. We then compare these distances with the corresponding Euclidean distances in our reconstructions. The ratio of the Mast3r-based distance to the reconstructed distance gives a scale multiplier for each pair, and we average these values at the scene level. The resulting mean scale multiplier for a scene is then applied to align that scene’s trajectory to metric scale.

Figure 10 shows the mean and standard deviation of the scene-wise metric scale multipliers. As indicated by the figure, most of the scale multipliers fall within a reasonable range, with low variance.

13. Bradley-Terry Evaluation

In this section, we explain our evaluation process to generate Bradley-Terry (BT) normalized scores (between 0 and

1) for each of the Multi-Image-to-Text methods.

Algorithm. We begin by creating pairs of generated summaries, comparing outputs from each method to the ground truth for all 130 scenes in our test set. To mitigate positional bias, we shuffle these pairs to randomize the order in which they are presented to the LLM. For evaluation, we use the GPT-4o model [30] with a temperature of 0.5 for text generation. Although we also tested open-source models like Llama-3.1-7b, we found they exhibit strong ordering bias, consistently favoring the first summary presented.

After gathering the binary preferences from the GPT-4o model, we construct a preference matrix $\mathcal{M}(i, j)$, where the entry in the i^{th} row and j^{th} column represents how many times method i is preferred over method j . The score calculation algorithm starts with an initial guess for the parameters: each method is assigned a value of 1 (stored in the array π_i). These parameters represent the latent “strength” or ability of each method. We iteratively update the values of π_i as follows:

$$\pi_i = \frac{\text{wins}_i}{\sum_{j \neq i} \frac{N[i, j]}{\pi_i + \pi_j}},$$

where wins_i is the total number of wins for method i , and $N[i, j]$ is the total number of comparisons between methods i and j . The denominator arises from setting the derivative of the likelihood function to zero (maximizing the likelihood), and it reflects how often method i has been compared with method j relative to their current estimated abilities. After updating all the parameters, the algorithm checks whether the maximum change in any parameter is below a specified tolerance threshold. If so, the iteration stops, indicating convergence.

Transforming Parameters to Scores. Once the parameters π have been estimated, they are converted into scores s_i in the interval $(0, 1)$ using the logistic transformation:

$$s_i = \frac{\pi_i}{\pi_i + 1}.$$

This transformation is useful because it maps the potentially unbounded π_i values to probability-like scores, making them easier to interpret.

Prompt to the Judge. We use the following prompt as input to the GPT-4o judge:

"Given Prediction 0, Prediction 1 and the Ground-Truth texts, select which text is closer to the ground-truth. Evaluate the texts only based on the information available in ground-truth."



“The spacious office features a large window with blinds for privacy and light control, as well as a whiteboard on the opposite wall for brainstorming sessions or team meetings. The open layout includes multiple workstations equipped with ergonomic chairs, computer monitors, and desktops. There is also a large desk area with additional chairs for collaborative workspaces. The office has ample storage options, including a wooden file cabinet, a credenza, and several deskside drawers. A large closet provides additional storage space, while a small closet near the door offers personal belongings storage. This room would be suitable for a variety of professional settings, such as law firms, accounting offices, architectural firms, engineering firms, and more.”

Figure 8. **Inference on ScanNet++.** Generated text and trajectory based on 5 frames from scene: 56a0ec536c.

“The home features a large front porch with a decorative ceiling and columns, as well as a front yard with a landscaped garden bed. The home has been freshly painted and features a new roof, new windows, and new flooring throughout. The home also includes a detached two-car garage with an attached carport, a large backyard with a garden bed, and a covered patio area.”

Figure 9. **Inference on online video.** Generated text based on exterior drone shot, <https://www.youtube.com/watch?v=QZrjzbI-H00>.

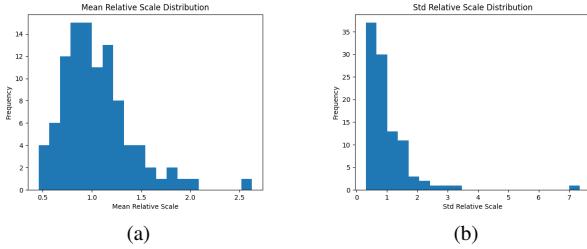


Figure 10. The **Mean** and **Standard Deviation** of the scale multipliers required to achieve metric scale.

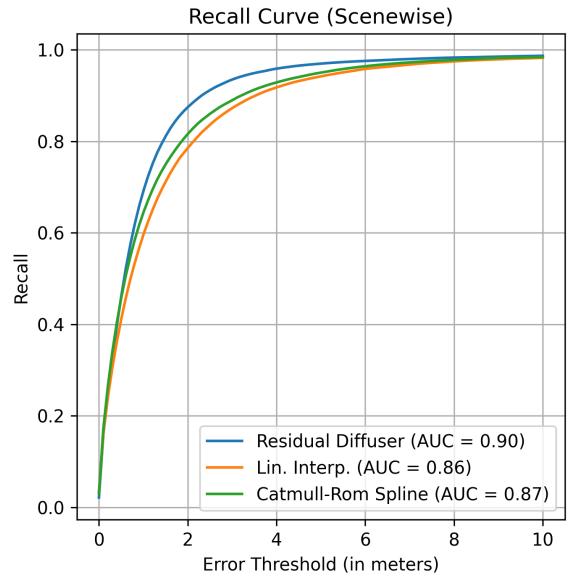


Figure 11. **Recall curve for 3D camera trajectory generation.** The x-axis shows the error threshold (in meters), and the y-axis indicates the ratio of predictions with errors below this threshold.

14. Recall Curves

In our trajectory generation evaluation, we use recall as a way of quantifying the magnitude of errors that the generation methods achieve. In the main paper tables (Tables 1 and 2) we report the results for $R@50cm$, $R@75cm$ and $R@1m$. We provide the complete curves in Figure 11. Our method is shown to have the highest Area Under Curve (AUC) score and consistently shows better performance against the baselines with varying error thresholds.

15. Preliminaries

Generative modeling using denoising diffusion probabilistic models (DDPMs) aims to learn a probability distribution $p_\theta(\mathbf{x})$ that approximates the true data distribution of observed data \mathbf{x} . Unlike other generative methods – such as variational autoencoders or generative adversarial networks – which generate data in a single step, DDPMs gradually transform pure noise into structured data through an iter-

ative denoising process. The discrete stochastic denoising (reverse) process is modeled as a Markov chain, beginning at a predefined time step T where the signal is considered to be pure noise $p(x_T) = \mathcal{N}(x_T; 0, I)$. A neural network ϵ_θ is trained to predict the noise added at each timestep by minimizing the variational bound on the negative log likelihood, $\mathbb{E}[-\log(p_\theta(x_0))]$. In practice, the reverse process is typically parametrized using Gaussian distribution as follows:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (4)$$

Forward diffusion is a process that gradually adds noise to the data via a variance schedule $\beta_t \in (0, 1)$, determining the amount of noise introduced at each timestep t . This formulation enables a closed-form expression for sampling an arbitrary x_t , where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{0 \leq i \leq t} \alpha_i$. The conditional probability distribution $q(x_t|x_0)$ describes how likely x_t is, given the clean signal x_0 :

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (5)$$