CATSplat: Context-Aware Transformer with Spatial Guidance for Generalizable 3D Gaussian Splatting from A Single-View Image

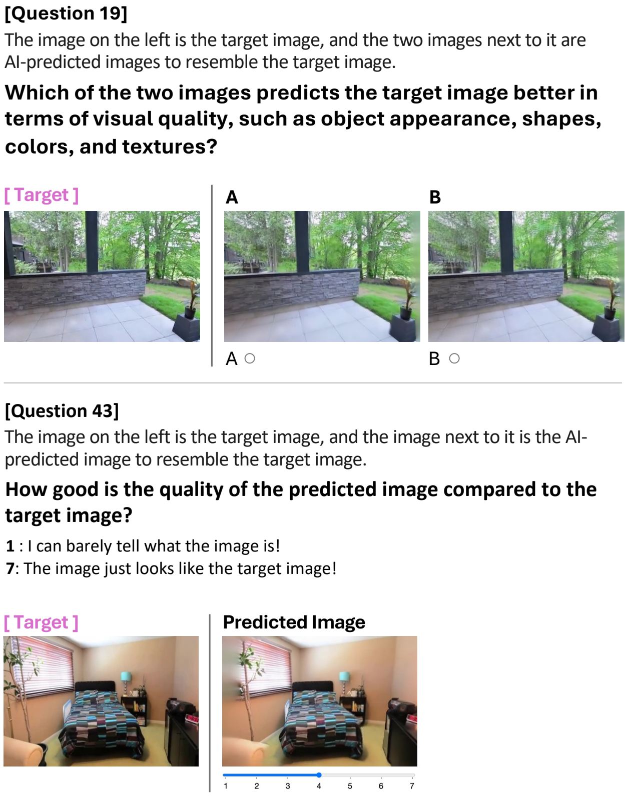
Supplementary Material

Overview

In this supplementary material, we provide further explanations and visualizations of our main paper, *"CATSplat: Context-Aware Transformer with Spatial Guidance for Generalizable 3D Gaussian Splatting from A Single-View Image"*. First, we elaborate on the specifics of our user study (Sec. 1). Then, we present additional technical details on the CATSplat architecture (Sec. 2). Also, we describe the implementation and datasets in more detail (Sec. 3). Moreover, we provide more quantitative and qualitative experimental results to further validate the robustness of CATSplat for 3D reconstruction and novel view synthesis (Sec. 4). We finally discuss the limitations and future directions (Sec. 5).

1. User Study Details

We conduct a user study to validate our method from the perspective of human perception, as described in Sec.4.4 in the main paper. Through Amazon Mechanical Turk (AMT), a widely used platform for user studies, we recruited 100 participants. We randomly sample 60 scenes from the RE10K [19] evaluation set and 20 from the ACID [7] evaluation set. Then, we use rendered images from these sampled scenes for the survey questions. With rendered images and corresponding ground truth target images, we ask two types of questions, as shown in Fig. 1. For the first type of question, we show two rendered images, one from CATSplat and the other from Flash3D [15], along with a target image, and ask, *"Which of the two images predicts the target image better in terms of visual quality, such as object appearance, shapes, colors, and textures?"*. For the second type of question, we request participants to rate the visual quality of the rendered image from either method (CATSplat or Flash3D) on a 7-point Likert scale, with the question, *"How good is the quality of the rendered image compared to the target image?"*. We also include control questions to verify the reliability of responses from each participant by displaying the ground truth image as the rendered image and asking participants to rate it based on the same ground truth image, where the results are expected to be obviously high. Moreover, the method names are anonymized and presented in random order to minimize bias. We finally gathered 9,000 responses on RE10K and 6,000 responses on ACID (*i.e.*, 30 questions for type one and 30 rating questions for each CATSplat and Flash3D on RE10K, as well as 20 questions for type one and 20 rating questions for each on ACID). Given responses from all participants, we report scores with 95% confidence intervals, as shown in Tab.7 of the main paper. Specifically, for the first type of question, which requires participants to choose between two rendered images, we utilize a binomial proportion confidence interval to analyze preferences. In the case of the second type, which queries to rate the visual quality of a single rendered image, we use a normal distribution confidence interval to analyze the average rating score. Ultimately, the results underscore the superiority of our method, as CATSplat is notably preferred and receives higher ratings compared to the latest method.



[Question 19]

The image on the left is the target image, and the two images next to it are AI-predicted images to resemble the target image.

Which of the two images predicts the target image better in terms of visual quality, such as object appearance, shapes, colors, and textures?

A ○ B ○

[Question 43]

The image on the left is the target image, and the image next to it is the AI-predicted image to resemble the target image.

How good is the quality of the predicted image compared to the target image?

1 : I can barely tell what the image is!
7: The image just looks like the target image!

Figure 1. Examples of two types of user study questions. The first type of question (above) asks about preference between ours and Flash3D [15], and the second (below) requires participants to rate the visual quality of the rendered image compared to the target.

2. Architecture Details

2.1. Details on 3D Point Feature Extraction

As described in Sec 3.3 in the main paper, we advocate incorporating 3D priors from 3D point features, which contain more comprehensive 3D domain knowledge than 2D depth maps, to address limited geometric information in-

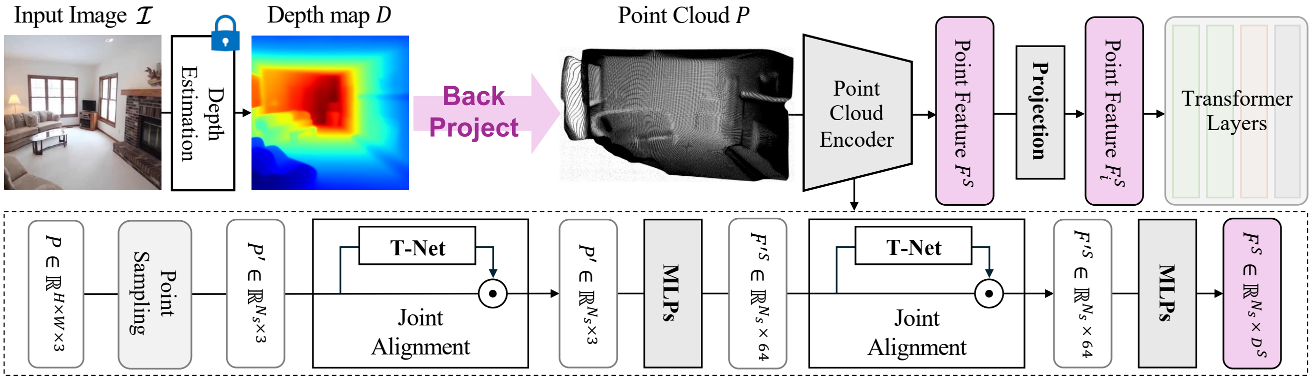

Input Image $\mathcal{I}$
Depth Estimation
Depth map D
Back Project
Point Cloud P
Point Cloud Encoder
Point Feature F^S
Projection
Point Feature F_i^S
Transformer Layers
$P \in \mathbb{R}^{H\times W\times 3}$
Point Sampling
$P' \in \mathbb{R}^{N_s\times 3}$
T-Net
Joint Alignment
$P' \in \mathbb{R}^{N_s\times 3}$
MLPs
$F'^S \in \mathbb{R}^{N_s\times 64}$
T-Net
Joint Alignment
$F'^S \in \mathbb{R}^{N_s\times 64}$
MLPs
$F^S \in \mathbb{R}^{N_s\times D^S}$

Figure 2. Detailed architecture of 3D point feature extraction from a monocular input image $\mathcal{I}$. Our point cloud encoder takes back-projected points P and produces point features F^S based on the PointNet [10] structure. Here, T-Net indicates an affine transform network.

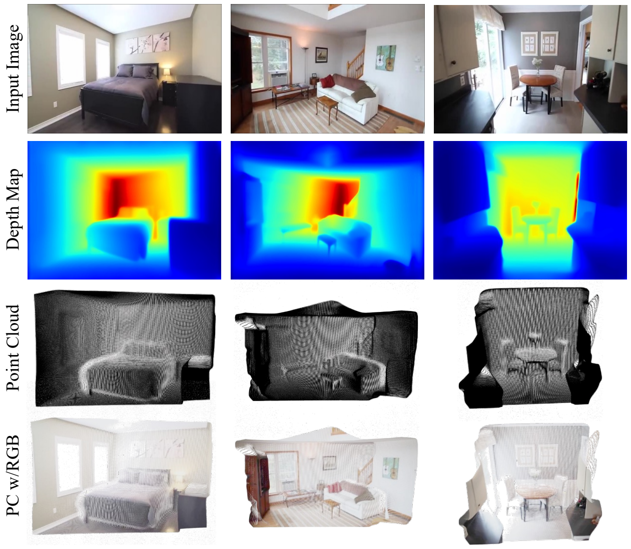


Figure 3. Examples of input images with their corresponding estimated depth maps and back-projected 3D point clouds. For better visualization, we also present 3D point clouds with RGB colors.

herent in single-view settings. In this section, we provide additional explanations on the procedure of producing 3D point features from a single source image. As illustrated in Fig. 2, our approach first extracts a pixel-wise depth map $D \in \mathbb{R}_+^{H\times W\times 1}$ from an input image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ using a pre-trained monocular depth estimation model [9]. Next, we back-project D into a 3D point cloud $P \in \mathbb{R}^{H\times W\times 3}$ with the corresponding camera parameters $K \in \mathbb{R}^{3\times 3}$. Then, a point cloud encoder takes P to yield point features $F^S \in \mathbb{R}^{N_s\times D^S}$. Here, we organize our point cloud encoder based on the prevalent PointNet [10] architecture. Given the points P, we sample N_s points using the Farthest Point Sampling (FPS) [2] algorithm; then, these sampled points $P' \in \mathbb{R}^{N_s\times 3}$ are processed through a series of joint alignment networks and MLP layers. The first alignment network maps the sampled points P' to a canonical space, and the second aligns intermediate features $F'^S \in \mathbb{R}^{N_s\times 64}$ to a joint feature space. Both networks employ an affine transform matrix predicted by the T-Net. Finally, we produce 3D point features $F^S \in \mathbb{R}^{N_s\times D^S}$, where D^S denotes 1,024. In Fig. 3, we present examples of input images $\mathcal{I}$, along with their corresponding depth maps D and back-projected 3D point clouds P (+ w/ RGB), to help understand our process.

2.2. CATSplat Procedure

In Algorithm. 1, we present the overall workflow of our generalizable feed-forward network, incorporating two novel priors, for 3D scene reconstruction from a single image.

Algorithm 1: 3D scene from a single-view image.

```
Input: A monocular image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$
Result: Novel view images $\hat{\mathcal{I}}_t \in \mathbb{R}^{H\times W\times 3}$
Procedure:
Estimate Depth Map $D$ from $\mathcal{I}$.
Concatenate $\mathcal{I}$ and $D$ as $\mathcal{I}'$.
Extract multi-resolution image features $F_i^{\mathcal{I}}$ from $\mathcal{I}'$.
Produce text features $F_i^C$ based on the VLM.
Back project $D$ into 3D points $P$.
Produce 3D point features $F_i^S$ from $P$.
# Multi-resolution Transformer with $N_l$ layers.
for $i = 1$ to $N_l$ do
    # Incorporation of Contextual Cues.
    $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i = W_q \cdot F_i^{\mathcal{I}}, \; W_k \cdot F_i^C, \; W_v \cdot F_i^C$
    $F_i^{\mathcal{I}C} = Attn(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$
    # Incorporation of Spatial Cues.
    $\mathbf{Q}_i', \mathbf{K}_i', \mathbf{V}_i' = W_q' \cdot F_i^{\mathcal{I}C}, \; W_k' \cdot F_i^S, \; W_v' \cdot F_i^S$
    $F_i^{\mathcal{I}CS} = Attn(\mathbf{Q}_i', \mathbf{K}_i', \mathbf{V}_i')$
    # Add and Normalization.
    $\tilde{F}_i^{\mathcal{I}CS} = \text{Norm}(F_i^{\mathcal{I}} + \gamma \, \text{Dropout}(F_i^{\mathcal{I}CS}))$
    # Self Attention.
    $\tilde{\mathbf{Q}}_i, \tilde{\mathbf{K}}_i, \tilde{\mathbf{V}}_i = \tilde{W}_q \cdot \tilde{F}_i^{\mathcal{I}CS}, \tilde{W}_k \cdot \tilde{F}_i^{\mathcal{I}CS}, \tilde{W}_v \cdot \tilde{F}_i^{\mathcal{I}CS}$
    $\tilde{F}_i^{\mathcal{I}} = Attn(\tilde{\mathbf{Q}}_i, \tilde{\mathbf{K}}_i, \tilde{\mathbf{V}}_i)$
end
# 3D Scene Reconstruction and Novel View Synthesis.
Predict $J$ Gaussians $\{(\boldsymbol{\mu}_j, \boldsymbol{\alpha}_j, \boldsymbol{\Sigma}_j, \boldsymbol{c}_j)\}_j^J$ from $\tilde{F}_i^{\mathcal{I}}$.
Render $\hat{\mathcal{I}}_t$ images with rasterization function.
```

3. Experimental Setup

3.1. Datasets

RealEstate10K. The RealEstate10K [19] dataset consists of large-scale home walkthrough videos from YouTube, including approximately 10 million frames from around 80,000 videos. It also provides camera parameters for each frame calibrated using the Structure-from-Motion (SfM) software. We follow the standard training and testing split, with 67,477 scenes for training and 7,289 for evaluation.

NYUv2. The NYUv2 [13] dataset provides video sequences from diverse indoor environments captured using Kinect cameras. In line with [15], we employ 250 source images from 80 scenes for cross-dataset evaluation and randomly sample target frames within a ±30 frame range from the source, following the random protocol of RE10K [19]. For camera trajectories, we use SfM software as RE10K.

ACID. The ACID [7] dataset consists of large-scale natural landscape videos captured by aerial drones. Like the RE10K [19], ACID provides camera parameters for frames, which are calculated via SfM software. For cross-dataset evaluation, we utilize 450 source images from 150 scenes and randomly sample target frames within a ±30 frame range from the source as the random protocol of RE10K. Note that we evaluate and visualize Flash3D [15] on ACID using publicly available code and provided checkpoints.

KITTI. The KITTI [3] is a landmark autonomous driving dataset containing 30 *city* driving sequences. Following the well-established evaluation protocol from Tulsiani et al. [17], we utilize 1,079 source frames and provided corresponding camera parameters for cross-dataset evaluation.

3.2. Scale Handling

For training on the RealEstate10K [19], we employ the SfM [12] strategy of COLMAP to estimate camera poses for images and align them with the estimated depth maps using the scale alignment process from Tucker *et al.* [16]. While this approach is generally reliable, it does not assure perfect alignment between the reconstructed scene scale and novel view camera pose scale due to occasional outliers in the predictions. Hence, when evaluating on RealEstate10K, ACID [7], and NYUv2 [13], we apply RANSAC to address such outliers, ensuring more robust scale handling. For the KITTI [3] dataset, we use the provided poses for evaluation.

3.3. Implementation Details

Our experimental setup is built on the prevalent deep learning framework, PyTorch. For image processing, we utilize the ResNet-50 [4] encoder and the UniDepth [9] pre-trained model for monocular depth estimation, with a single-view image size of 256 × 384. We employ LLaVA [8] 13B for text embeddings and extend the PointNet [10] encoder for extracting point features. Note that we precompute text embeddings to optimize training efficiency by minimizing computational overhead. Our multi-resolution transformer comprises three layers with 8-headed attention, leveraging three different resolution image features to effectively capture both global structures and fine details. We also set the ratio γ as 0.5 to strike a balance, preventing excessive loss of core visual information from image features while integrating our two novel priors. Then, our offset decoder predicts two sets of depth offsets and 3D offsets for vivid scene representation. We train and evaluate on a single A100 GPU and select the best model upon convergence. Specifically, we optimize a combination of $\mathcal{L}_{\ell 1}$, $\mathcal{L}_{\text{ssim}}$, and $\mathcal{L}_{\text{lpips}}$ losses using the Adam optimizer with each coefficient as $\lambda_{\ell 1}$=1, λ_{ssim}=0.85, and λ_{lpips}=0.01, respectively.

3.4. Computational Analysis

In Tab. 1, we compare the training and inference times of ours with Flash3D [15] on a single A100 GPU. For precise examination, we synchronize CUDA events across all operations, as they are typically asynchronous per process. Although CATSplat requires additional six hours (29.1 hrs) of training compared to Flash3D (23.2 hrs), the impact on inference time remains relatively minimal, increasing by only 0.066 secs. In terms of model parameters, CATSplat (72M) is 30% larger than Flash3D (54M) due to the transformer architecture, which introduces additional weight matrices and trainable layers. Despite the slightly higher computational costs, CATSplat consistently outperforms Flash3D across multiple benchmarks by employing two valuable guidance.

Method	Training (hrs)	Inference (secs)	n = *Random* (frames)		
			PSNR ↑	SSIM ↑	LPIPS ↓
Flash3D [15]	23.2	0.327	24.93	0.833	0.160
CATSplat (Ours)	29.1	0.393	**25.45**	**0.841**	**0.151**

Table 1. Comparison of computational costs with the state-of-the-art single-view 3DGS method, Flash3D [15], on the RE10K [19].

4. Additional Experiments

4.1. Ablation Studies in Cross-dataset Settings

In this section, we validate the effectiveness of our two innovative priors through ablative experiments across cross-dataset settings. In Tab. 2 and Tab. 3, we evaluate variants of our method, with/ and w/o Contextual and Spatial priors, on the NYUv2 [13] and ACID [7] datasets, respectively. The Baseline denotes our basic transformer architecture, excluding cross-attention with any of our proposed priors.

First, incorporating contextual cues leads to significant improvements, both for indoor scenes (NYUv2) and outdoor nature scenes (ACID). With text embeddings from a well-trained visual-langualge model (VLM) [8], our network learns not just basic object types or scene semantics but also deeper context, such as how objects relate to each other or the overall scene structure. In other words,

Method			n = *Random* (frames)		
Baseline	Contextual	Spatial	PSNR ↑	SSIM ↑	LPIPS ↓
✓	-	-	25.11	0.775	0.178
✓	✓	-	25.51	0.779	0.163
✓	-	✓	25.48	0.778	0.165
✓	✓	✓	**25.57**	**0.781**	**0.157**

Table 2. Ablation study to investigate the effect of our two intelligent priors on the NYUv2 [13] dataset in cross-dataset settings.

Method			n = *Random* (frames)		
Baseline	Contextual	Spatial	PSNR ↑	SSIM ↑	LPIPS ↓
✓	-	-	24.26	0.732	0.261
✓	✓	-	24.57	0.735	0.253
✓	-	✓	24.62	0.737	0.254
✓	✓	✓	**24.73**	**0.739**	**0.250**

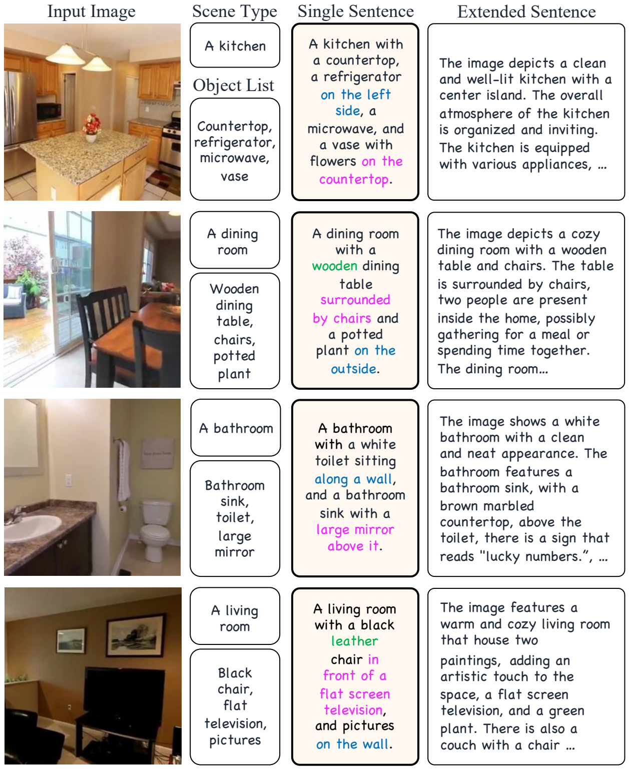
Table 3. Ablation study to investigate the effect of our two intelligent priors on the ACID [19] dataset in cross-dataset settings.

we take advantage of text embeddings to provide comprehensive general knowledge of dynamic real-world environments as well as scene-specific details. As a result, even in unfamiliar scenarios without familiar scene types or objects, text-embedded general cues serve as guiding anchors, enabling our network to better apply trained knowledge.

Additionally, by incorporating spatial guidance, our approach boosts generalization performance on both datasets. Beyond the geometric cues from 2D depth maps, we guide our network to be aware of three-dimensional domains, more associated with 3D Gaussians, through 3D point features. Based on deep spatial understandings, our network effectively reconstructs 3D scenes with accurate Gaussians, even in complex, unfamiliar environments. Finally, combining all priors together achieves further advances, seamlessly complementing limited knowledge from single-view image features. In addition to Tab.4 in our main paper, these results demonstrate the significance of our two novel priors.

4.2. Discussion on Text Descriptions

For rich contextual cues, we leverage text embeddings from a well-trained VLM [8]. Specifically, we prompt the VLM to generate text descriptions for the input image; then, we utilize intermediate text embeddings before they are processed into linguistic descriptions. To discover the optimal text embeddings, we investigate the impact of contextual information within different types of text embeddings on generalizability in Tab.5 (main). For comparison, we conduct experiments with four different styles of prompts: identifying the scene type, listing objects, describing the scene with a detailed single sentence, and two or more extended sentences. We provide examples of text description outputs using these prompts in Fig. 4. Usually, a single sentence captures comprehensive details for the scene, including textures (*e.g.*, *"wooden"*, *"leather"*), object relationships (*e.g.*, *"on the countertop"*, *"surrounded by chairs"*, *"large mirror above it"*), and overall composition (*e.g.*, *"on the left side"*, *"on the outside"*), surpassing simple cues like scene type or object list. However, extended sentences often introduce exaggerated or fabricated elements, such as overly interpretive moods, atmospheric descriptions with excessive adjectives (*e.g.*, *"organized and inviting"*, *"adding an artistic touch"*), or entirely false specifics (*e.g.*, *"two people are present inside the home.."*, *"lucky numbers.."*). These noisy overstatements hinder the network from learning meaningful context information of the text embeddings, resulting in relatively lower performance than using a single sentence. Ultimately, in this work, we employ a single sentence to enhance image features with practical contextual information.



Scene Type	Object List	Single Sentence	Extended Sentence
A kitchen	Countertop, refrigerator, microwave, vase	A kitchen with a countertop, a refrigerator on the left side, a microwave, and a vase with flowers on the countertop.	The image depicts a clean and well-lit kitchen with a center island. The overall atmosphere of the kitchen is organized and inviting. The kitchen is equipped with various appliances, ...
A dining room	Wooden dining table, chairs, potted plant	A dining room with a wooden dining table surrounded by chairs and a potted plant on the outside.	The image depicts a cozy dining room with a wooden table and chairs. The table is surrounded by chairs, two people are present inside the home, possibly gathering for a meal or spending time together. The dining room...
A bathroom	Bathroom sink, toilet, large mirror	A bathroom with a white toilet sitting along a wall, and a bathroom sink with a large mirror above it.	The image shows a white bathroom with a clean and neat appearance. The bathroom features a bathroom sink, with a brown marbled countertop, above the toilet, there is a sign that reads "lucky numbers.", ...
A living room	Black chair, flat television, pictures	A living room with a black leather chair in front of a flat screen television, and pictures on the wall.	The image features a warm and cozy living room that house two paintings, adding an artistic touch to the space, a flat screen television, and a green plant. There is also a couch with a chair ...

Figure 4. Examples of four different formats of text descriptions from the VLM [8], as described in Tab.5 in the main paper.

4.3. Text Embeddings from Various VLMs

Contextual cues from text embeddings are one of our core methods to break through the inherent constraints in monocular settings. Thus, identifying the most effective text embeddings is crucial for achieving high-quality single-view 3D scene reconstruction. In Tab. 4, we explore how text embeddings from various pre-trained VLMs, including OpenFlamingo [1], BLIP2 T5 [6], LLaVA 7B [8], and LLaVA 13B, influence performance on the RE10K [19]. For a fair comparison, we prompt all VLMs to produce a single sentence scene description. Then, we use intermediate text embeddings from each VLM. Even with similar prompts, each model generates distinct structures of text descriptions. For example, OpenFlamingo tends to produce relatively unsta-

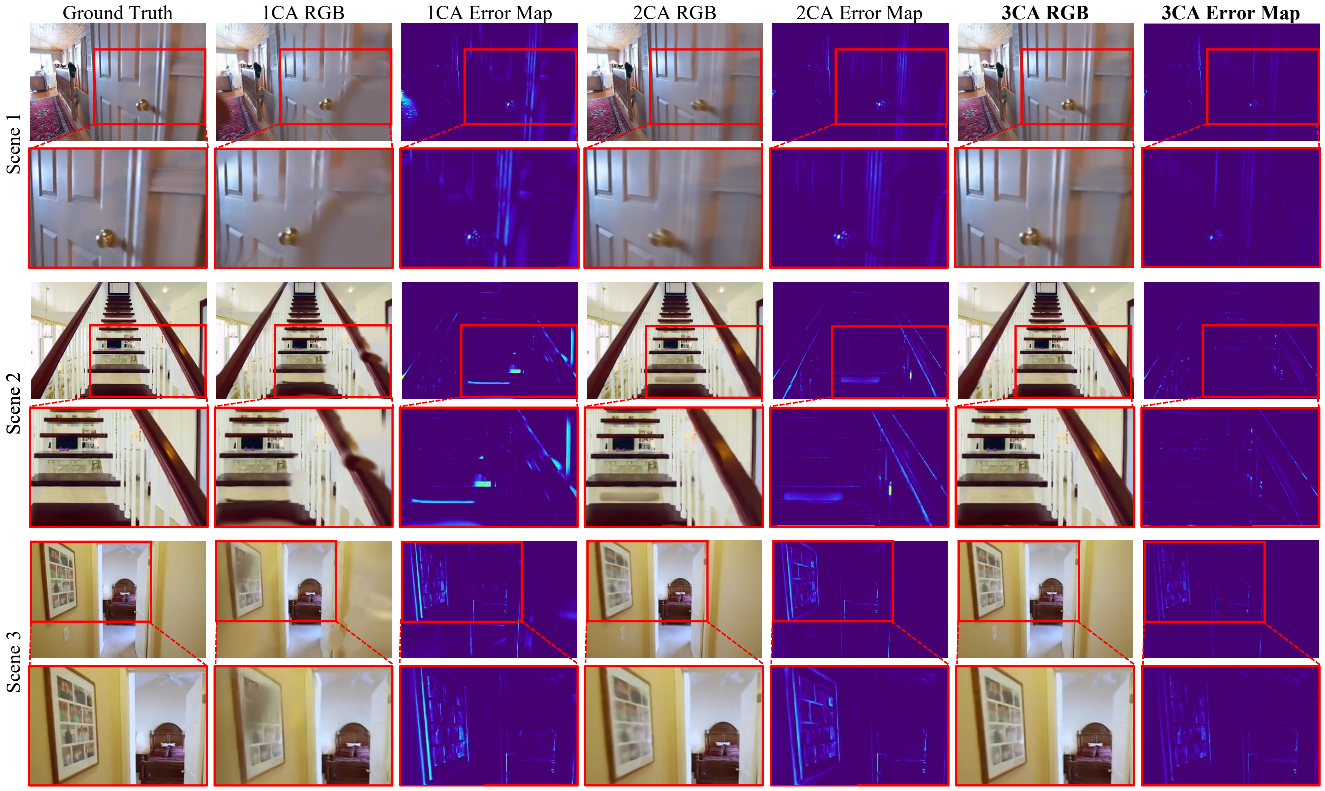


Figure 5. Ablation study to see the effect of iteratively incorporating our novel priors on the RE10K [19] (n=*Random*). For clear ablations, we keep the number of entire transformer layers consistent across the experiments and adjust only the number of cross-attentions (CA).

	n = 10 (frames)			n = *Random* (frames)		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
OpenFlamingo	26.08	0.858	0.131	25.06	0.832	0.158
BLIP2 T5	26.29	0.860	0.129	25.27	0.833	0.156
LLaVA 7B	26.19	0.861	0.129	25.23	0.834	0.156
LLaVA 13B	**26.40**	**0.864**	**0.127**	**25.40**	**0.838**	**0.153**

Table 4. Ablation study to see the impact of text embeddings from different VLMs: OpenFlamingo [1], BLIP2 [6], and LLaVA [8].

ble text descriptions with redundant or exaggerated information, providing minimal benefit for scene reconstruction. Meanwhile, BLIP2 and LLaVA 7B generate monotonous text descriptions that primarily focus on object and scene types. On the other hand, LLaVA 13B yields more informative text descriptions with useful details, such as textures (*e.g.*, *"wooden"*, *"leather"*), object relations (*e.g.*, *"on the countertop"*, *"surrounded by chairs"*, *"large mirror above it"*), and scene composition (*e.g.*, *"on the left side"*, *"on the outside"*), as illustrated in Fig. 4. Finally, we leverage text embeddings from the well-aligned multimodal space of LLaVA 13B for context-aware 3D scene reconstruction.

4.4. Iteratively Incorporating Priors

We present additional ablative experimental results to highlight the benefits of iteratively incorporating our priors in Fig. 5. Consistent with the settings in Fig.5 (main), we randomly sample the target frame within a ±30 range; also, fix the total number of transformer layers at three and apply cross-attention either in the first layer only, across two layers, or throughout all three layers. Through iterative cross-attention between image features and our priors, blurry artifacts gradually fade, sharpening the object contours and enhancing clarity in images. Simultaneously, errors between rendered images and target images also steadily decrease. In essence, iterative incorporations of our novel priors lead to noticeable improvements in overall visual quality. These results emphasize both the importance of our priors and the structural robustness of our transformer-based architecture.

4.5. Comparisons with Various Approaches

In addition to comparing our method with monocular novel view synthesis (NVS) approaches, we evaluate ours against various types of NVS methods to further highlight its effectiveness. For a fair comparison, we assess all methods on unseen scenes in cross-dataset settings across ACID [7] and NYUv2 [13] datasets. First, we compare ours with MASt3R [5] and Splatt3R [14], both optimized for processing stereo image pairs in 3D scene reconstruction. MASt3R relies on feature-based matching between image pairs, and Splatt3R extends MASt3R by incorporating additional prediction heads to directly estimate Gaussian primitives. While both methods excel in 3D reconstruction with two images, they tend to produce relatively blurry results in monocular settings due to their inherent dependence on dual inputs. MoGe [18] extracts high-level features from a single

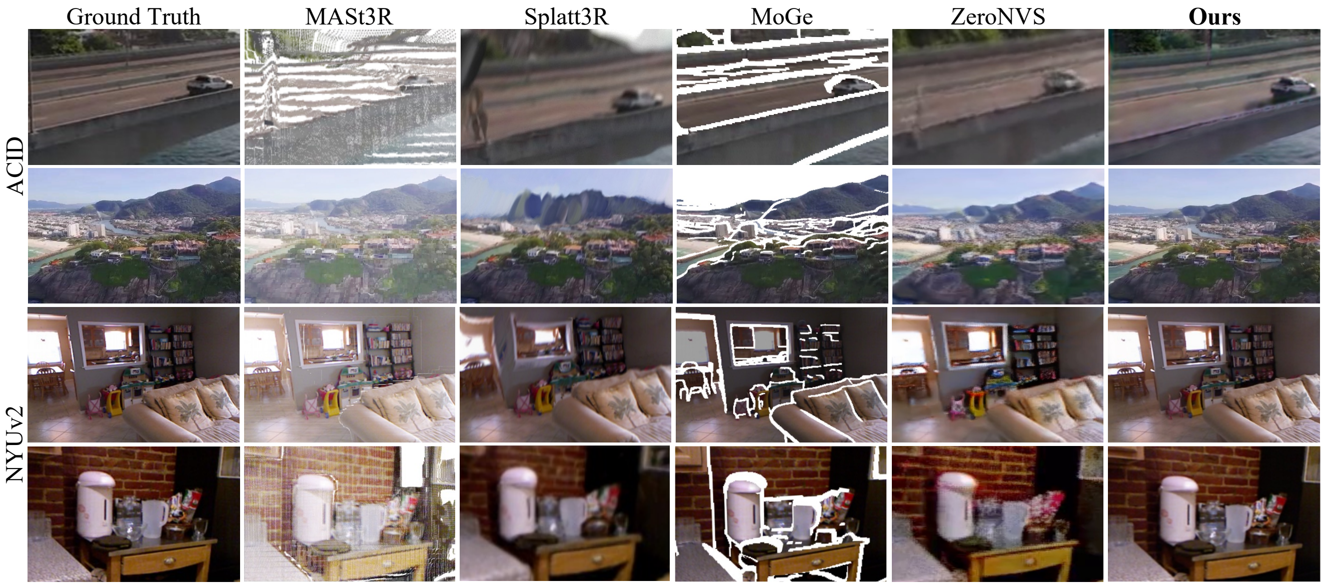


Figure 6. Qualitative comparisons with various novel view synthesis approaches, including MASt3R [5], Splatt3R [14], MoGe [18], and ZeroNVS [11]. All methods are fairly evaluated on unseen scenes in cross-dataset settings across ACID [7] and NYUv2 [13] datasets.

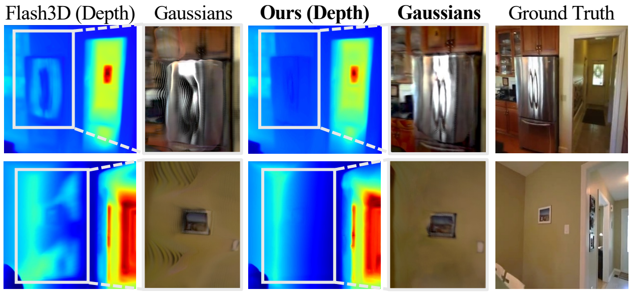


Figure 7. Qualitative comparisons of 3D reconstruction between Flash3D [15] and ours with Ground Truth. We visualize zoom-in views of 3D Gaussian distributions and depth maps from them.

image and regresses these features into 3D point maps for 3D scene reconstruction. However, as shown in the fourth column, unlike ours, which represents the 3D scene using continuous 3D Gaussians, the use of 3D points often results in large gaps in the outlines of objects from novel viewpoints. Moreover, ZeroNVS incorporates [11] diffusion processes based on NeRF architecture to construct detailed 3D scenes from a single image. Although it achieves high-quality novel view images, combining diffusion with NeRF-based volume rendering demands significantly higher computational costs ($\approx$ 3 hrs). On the other hand, CATSplat efficiently reconstructs solid scenes for novel viewpoints using explicit 3D Gaussian distributions in a single forward pass.

4.6. Additional Comparisons with Flash3D

In addition to comparing rendered RGBs, we qualitatively assess the quality of 3D Gaussians for scene representation. In Fig. 7, ours predicts clearer Gaussians than Flash3D [15], which exhibits messy artifacts. Our excellence is also evident in the depth maps produced by these 3D Gaussians.

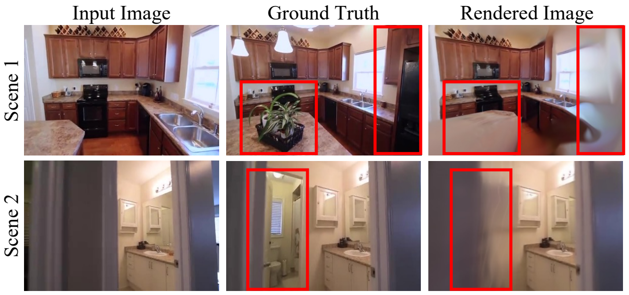


Figure 8. Failure cases of CATSplat. When invisible areas in the input become visible in the target, ours might be less productive.

We present more qualitative comparisons with Flash3D on the RE10K [19] (Fig. 9 and Fig. 10) as well as ACID [7] (Fig. 11) and KITTI [3] (Fig. 12) in cross-dataset settings.

5. Limitations and Future Work

Although CATSplat shines in monocular 3D scene reconstruction with two additional priors, it does not ensure perfect novel view synthesis across all real-world scenarios. Depending on dynamic camera movements, when regions that are occluded, truncated, or even entirely missing in the input image appear in the target view, ours might be less effective. For example, in Fig. 8, when previously unseen elements, like green plants absent in the input, emerge in the target view (Scene1) or when areas of the bathroom, once hidden behind a door, become visible (Scene2), our model struggles to reconstruct these newly revealed parts. In the future, we plan to explore involving generative knowledge to better handle these unseen regions in monocular 3D scene reconstruction. Moreover, we believe that training the model on a broader range of datasets will strengthen its general understanding of challenging natural environments.

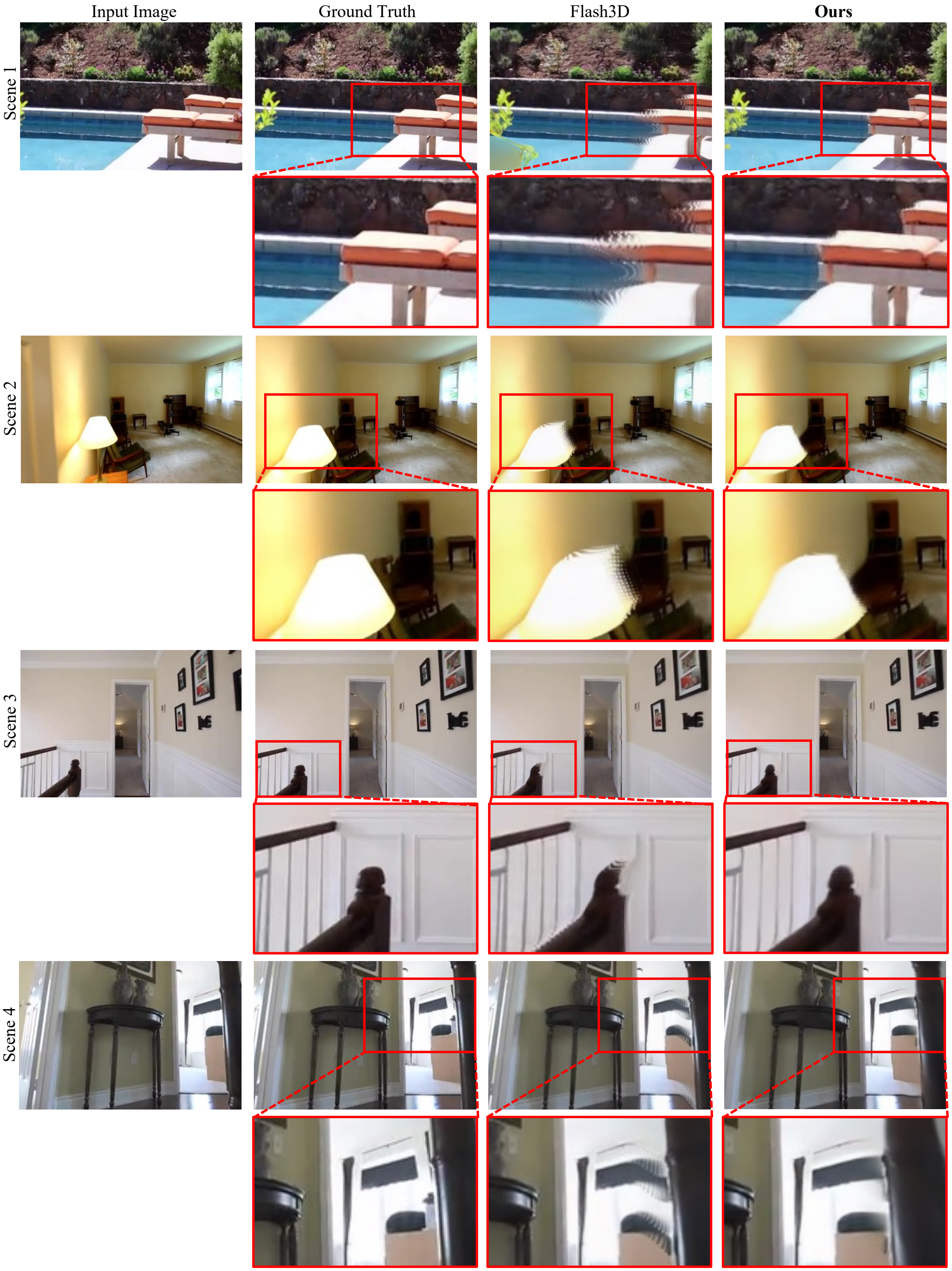


Figure 9. Qualitative comparisons between Flash3D [15] and Ours with Input Image and Ground Truth on the RealEstate10K [19] dataset.

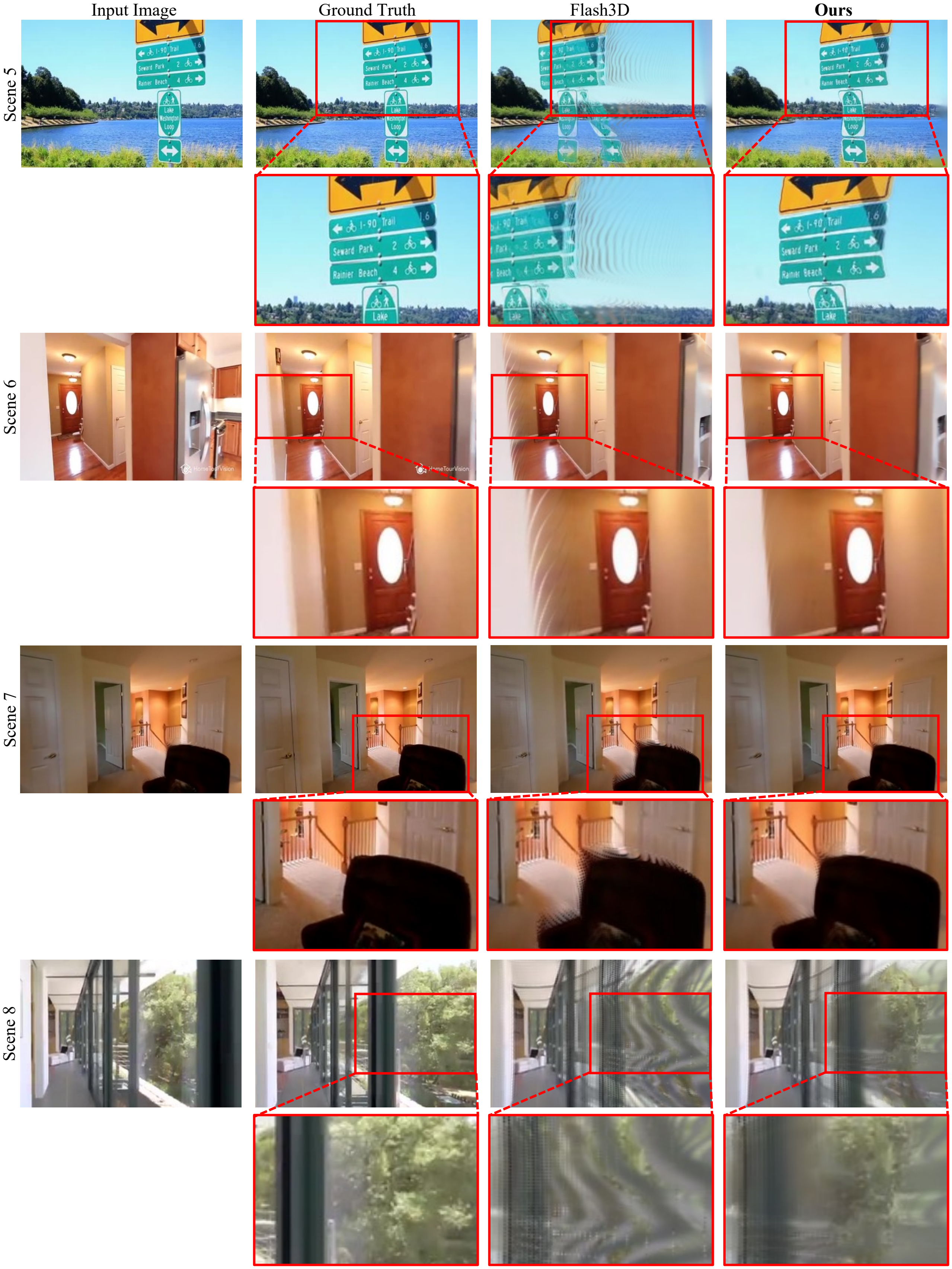


Figure 10. Qualitative comparisons between Flash3D [15] and Ours with Input Image and Ground Truth on the RealEstate10K [19] dataset.

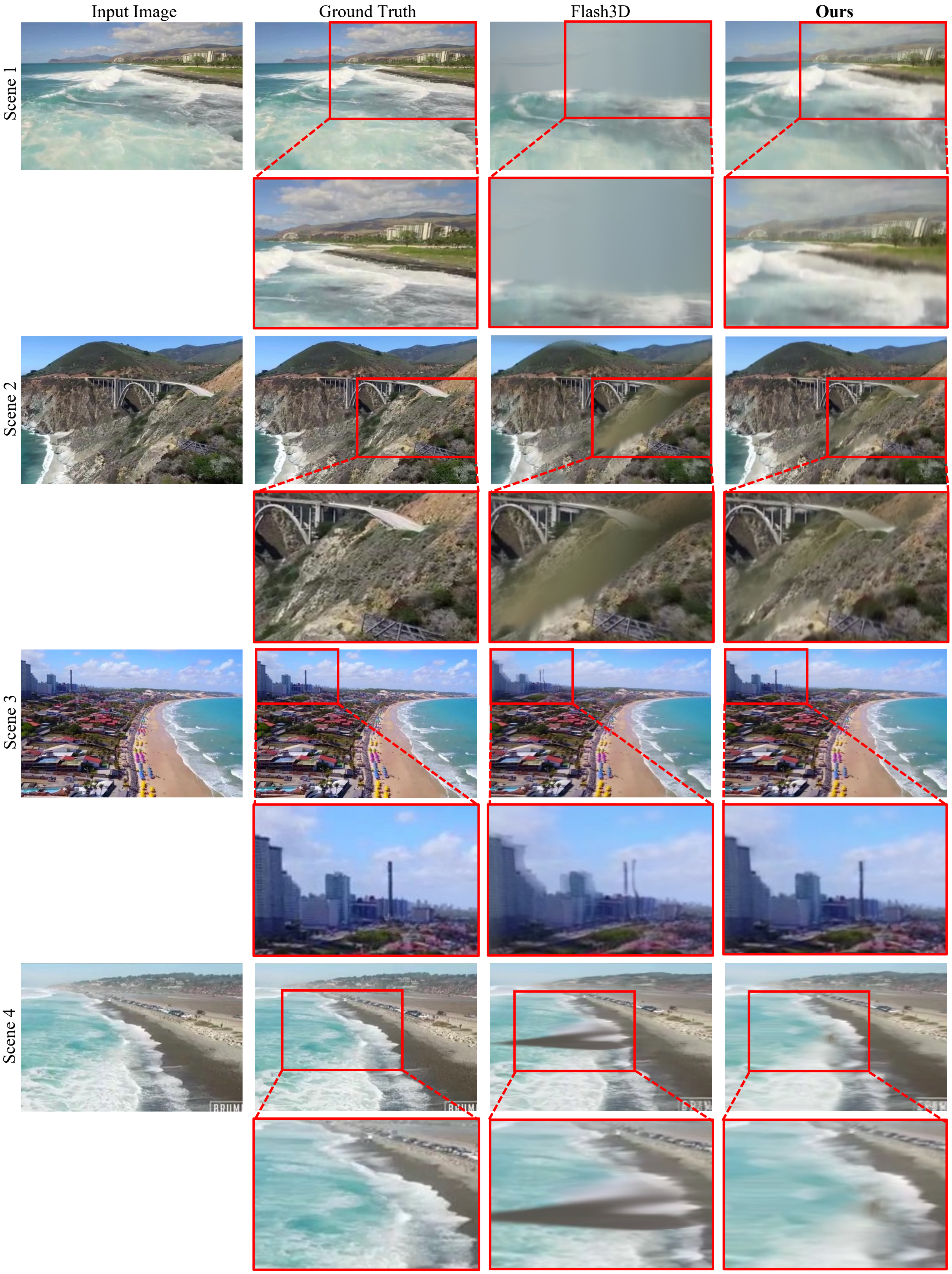


Figure 11. Qualitative comparisons between Flash3D [15] and Ours with Input Image and Ground Truth on the ACID [7] dataset.

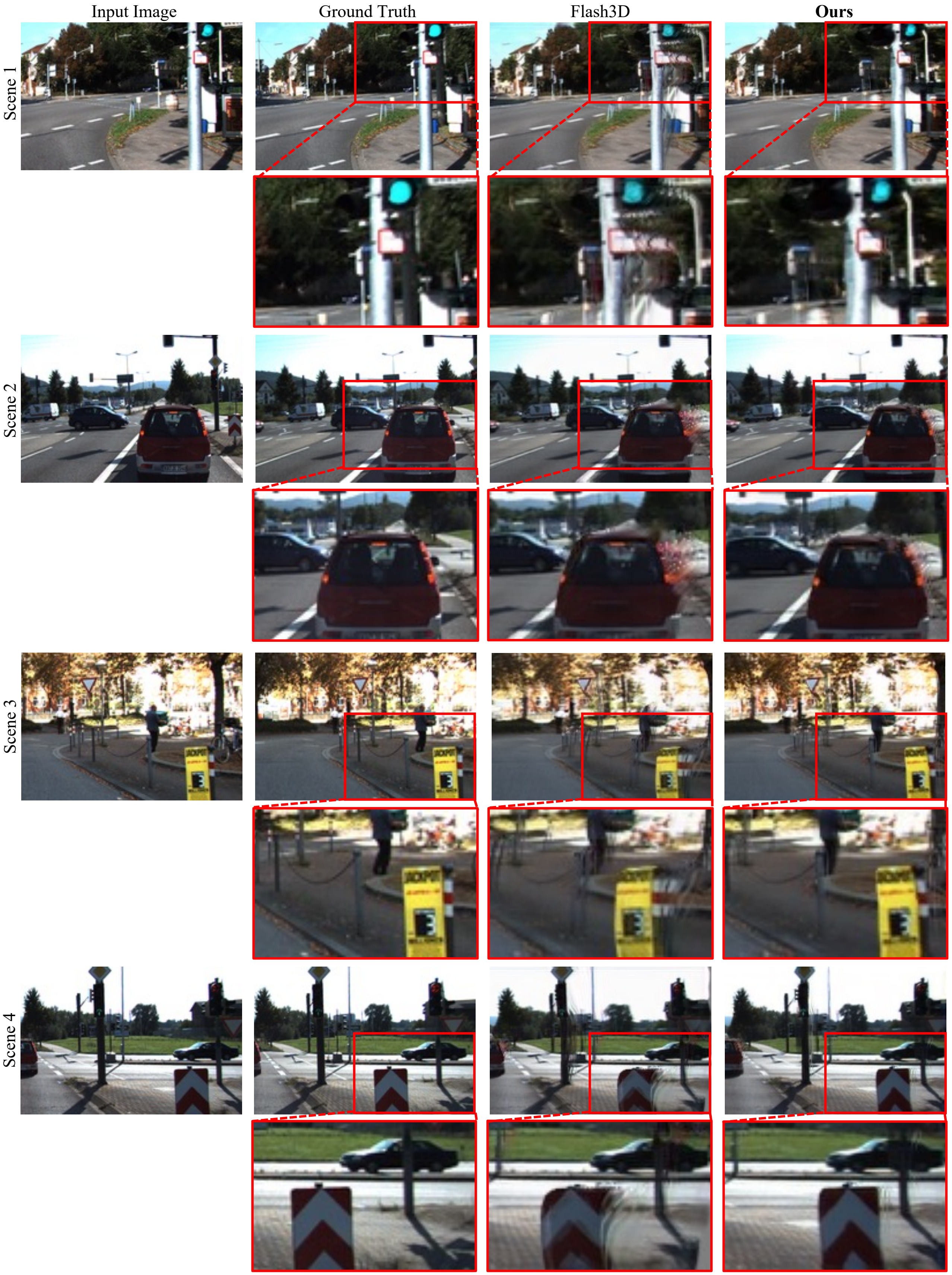


Figure 12. Qualitative comparisons between Flash3D [15] and Ours with Input Image and Ground Truth on the KITTI [3] dataset.

References

[1] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 4, 5

[2] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9):1305–1315, 1997. 2

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 3, 6, 10

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] 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. 5, 6

[6] 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. 4, 5

[7] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 1, 3, 5, 6, 9

[8] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3, 4, 5

[9] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 2, 3

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3

[11] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9420–9429, 2024. 6

[12] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3

[13] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 3, 4, 5, 6

[14] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 5, 6

[15] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 1, 3, 6, 7, 8, 9, 10

[16] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 3

[17] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 3

[18] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024. 5, 6

[19] 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. 1, 3, 4, 5, 6, 7, 8