

# Inpainting 3D Gaussians via Learning Depth Completion from Diffusion Prior

## Supplementary Material

The supplementary materials are structured as follows: We begin with a detailed description of our implementation, including the specifics of our training configurations and the outlier removal process for the point cloud. Subsequently, we undertake an in-depth examination of various issues discussed within the paper. To conclude, we provide a broader set of results, encompassing a wider array of scenes and viewpoints. Additionally, we compare our approach with two possible depth inpainting methods on standard depth estimation benchmarks: (1) performing monocular depth estimation followed by alignment with known depth, and (2) depth completion-related methods. We sincerely invite you to review the HTML files in our supplementary materials, which contain visualizations of the relevant results.

### Contents

<b>A Implementation Details</b>	<b>1</b>
A.1. Training details . . . . .	1
A.2. Outliers removal . . . . .	1
<b>B Analysis</b>	<b>1</b>
B.1. Progressive Inpainting . . . . .	1
B.2. Initialization Details . . . . .	3
B.3. Analysis of pre-trained weights . . . . .	3
B.4. Analysis of depth inpainting . . . . .	3
B.5. Elaboration on point cloud merging . . . . .	3
B.6. Analysis of optimization iterations . . . . .	3
<b>C More Qualitative Results</b>	<b>3</b>
C.1. More 3D Gaussians inpainting results . . . . .	3
C.2. Results under complex masks . . . . .	3
<b>D More Quantitative Results</b>	<b>4</b>
D.1. Comparison with monocular depth estimation methods . . . . .	4
D.2. Comparison with depth completion methods	5

### A. Implementation Details

#### A.1. Training details

The Depth inpainting model is initialized with the Marigold [9] weights. The architecture of the neural network is consistent with that of Stable Diffusion v1.5 [13], with the exception of the first convolutional layer. Moreover, during both training and inference phases, the input to the text encoder is persistently an empty string. The UNet has 9 additional input channels (4 for the encoded masked-depth, 4 for the guided encoded image and 1 for the

mask itself) whose weights were zero-initialized. During training, we generate synthetic masks. In the context of data processing, we maintain the original aspect ratio of the images during both the training and inference stages, resizing them to a maximum resolution of 768 pixels on the longest side.

#### A.2. Outliers removal

we unproject depth map and reference image from image space to 3D coordinates to form a colored point cloud. Before this point cloud is merged into original 3D Gaussian point cloud, we need process outliers to improve rendered image quality. To eliminate Gaussian outliers along the edges of the mask, we initially construct a KDTree from the unprojected point cloud. Subsequently, this KDTree is employed to locate the nearest points within the original point cloud, returning points from the original cloud that are within a specified distance threshold. Subsequently, we utilize the *'remove\_radius\_outlier'* method from the point cloud data (pcd) library to identify points in the original point cloud that have an insufficient number of neighbors within a specified radius. An intersection of these points and the similar points previously determined using a KDTree is performed, thereby efficiently removing Gaussian outliers at the edges of the mask. Additionally, there are various Gaussian segmentation [2, 5, 7, 10, 19] techniques that can be employed for outlier removal, taking advantage of the explicit properties of Gaussian models. Nevertheless, these are not the focal point of the present study and will not be deliberated here.

### B. Analysis

#### B.1. Progressive Inpainting

For occlusion-rich, complex scenes, multiple reference views ( $r > 1$ ) are imperative. To solve these challenges, we implement a progressive inpainting approach. Commencing with the initial reference view  $s(i_1)$  from the selected views  $\mathcal{S} = \{s(i_1), s(i_2), \dots, s(i_r)\}$ , we apply Gaussian inpainting. Subsequent to this, we render the color image, depth map, from the next reference view  $s(i_2)$ . For the mask, we follow Gaussian grouping [19] and utilize the Tracking-Anything-with-DEVA [3] to detect the invisible regions. This process is iterated, employing inpainting for each successive reference view until the view  $s(i_r)$  is addressed.

In our current implementation, we define a circular camera path starting from the first reference view and place reference views procedurally along it. At each step, we use the Gaussians reconstructed from the previous step, perform

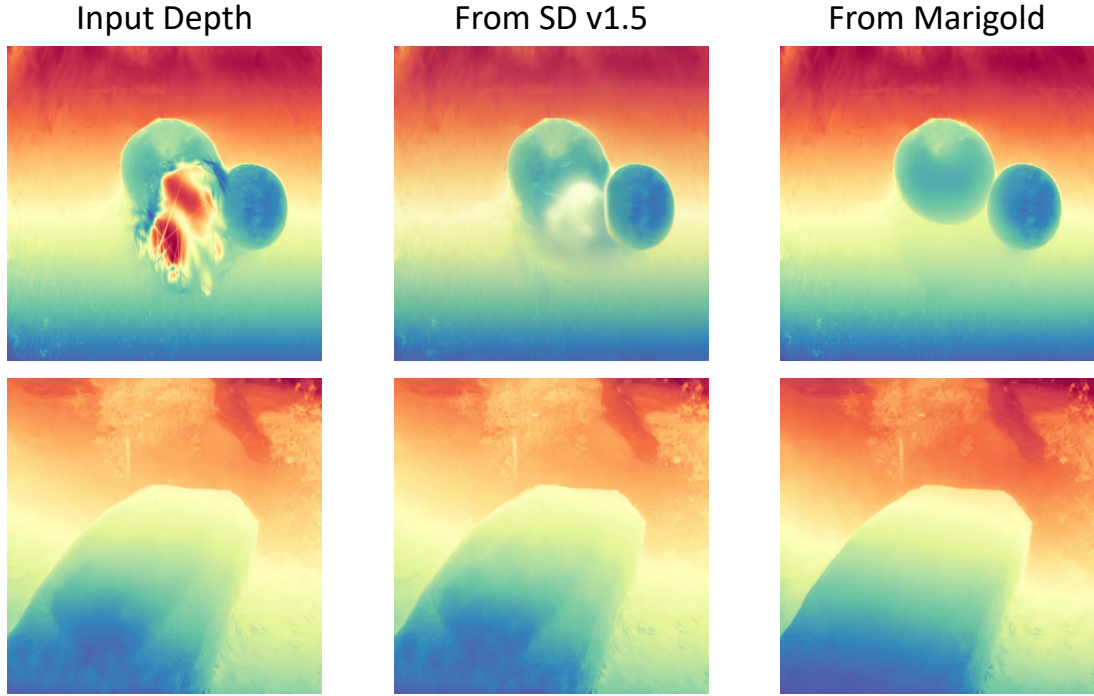


Figure S1. Analysis of Pre-trained Weights.

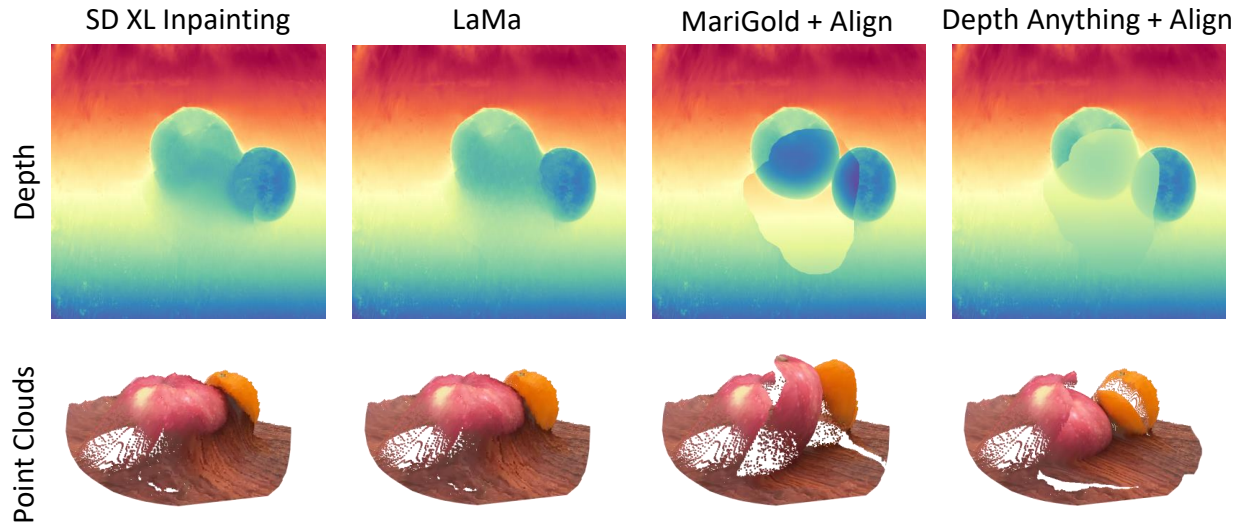


Figure S2. **Analysis of Depth Inpainting.** It is evident that the image-based inpainting models, lacking proper guidance, fail to adequately complete the geometric details. Regarding the monocular estimation methods, while a depth alignment method is implemented, they often lead to discontinuities within the inpainted regions

depth inpainting in the new reference view, and optimize the new Gaussians. The main challenge is ensuring stable progressive inpainting. The pipeline still involves trial and error to determine the optimal degree change for the next reference.

As evidenced in our results, increasing the number of views enhances the handling of occlusions (Fig. S3).

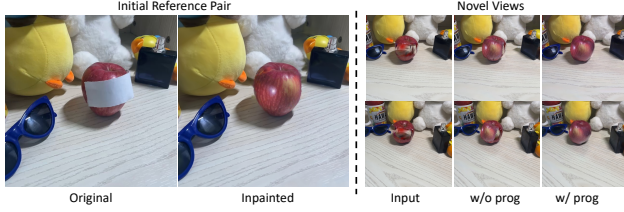


Figure S3. **Ablation study on progressive inpainting.** InFusion can adeptly handle inpainting tasks for views that substantially deviate from the initial reference frames.

## B.2. Initialization Details

The parameters are set as follows: opacity at 0.1, first-degree SH coefficients matching the pixel RGB values of the unprojected points, higher-degree coefficients set to 0, and Gaussian radius determined by the nearest point's distance.

## B.3. Analysis of pre-trained weights

For the task of depth completion, we employ two distinct sets of initial weights: one derived from Marigold and the other based on Stable Diffusion v1.5. As demonstrated in Fig. S1, we display the results under conditions of equivalent data volume and identical training epochs. It is discerned that models initialized with weights from Stable Diffusion v1.5 encountered greater challenges in mastering the depth completion task, a difficulty that was particularly pronounced in complex scenes. In contrast, models that began with Marigold weights exhibited superior proficiency in completing depth, due to prior training on depth maps that reduced the gap between the RGB and depth domains. Following the same training regimen, these models demonstrated an enhanced ability for depth completion and achieved better alignment with the input images.

## B.4. Analysis of depth inpainting

We include feature additional results, comparing our method with various cutting-edge baselines, such as SD XL inpainting [12] and DepthAnything [17], with a focus on alignment accuracy. As shown in Fig. S2, while SD XL inpainting yields visually appealing results in the RGB domain, a closer inspection of the reprojected point clouds reveals noticeable inaccuracies, akin to those observed in LaMa. Similarly, DepthAnything struggles with discontinuities, leading to a pronounced gap between inpainted areas and their adjacent regions, much like the issues seen with MariGold. Consequently, our learned depth inpainting is critical in securing high-fidelity results.

## B.5. Elaboration on point cloud merging

For a clearer understanding of the merging process, as shown in Fig. S4, we illustrate the original Gaussians,

the unprojected inpainted Gaussians, and the result after merging and fine-tuning. Specifically, using the camera intrinsics and extrinsics of the reference view, we project the inpainted depth into 3D space, thereby obtaining the 3D coordinates of the inpainted points. We then initialize Gaussians for optimization with these inpainted points.



Figure S4. **Elaboration on point cloud merging.**

## B.6. Analysis of optimization iterations

We test performance across different optimization iterations. As shown in Fig. S5, thanks to the accurate initial positions, with 50 to 200 iterations, the initial colored Gaussians quickly achieved good visual quality. Floating points appeared after 1000 iterations from a single view, with noticeable disordered Gaussians emerging after 3000 iterations.



Figure S5. **Analysis of optimization iterations.** Please zoom in for details.

Method	NYUv2		ETH3D		DIODE	
	AbsRel↓	$\delta 1 \uparrow$	AbsRel↓	$\delta 1 \uparrow$	AbsRel↓	$\delta 1 \uparrow$
Marigold [9]	5.7	96.3	6.5	95.7	30.3	77.3
GeoWizard [6]	5.3	96.5	6.3	96.2	29.4	79.4
w/ Guidance	5.1	96.3	6.0	96.0	29.7	79.2
w/ Blend	5.4	96.0	6.1	95.9	29.2	79.5
DepthAnything [16]	4.5	97.4	12.4	89.0	27.2	76.4
DepthAnythingV2 [18]	4.3	<b>98.1</b>	12.9	86.5	27.0	76.9
Ours	<b>3.7</b>	97.9	<b>4.9</b>	<b>97.0</b>	<b>22.1</b>	<b>80.3</b>

Table S1. **Comparison with depth estimation methods.**

## C. More Qualitative Results

### C.1. More 3D Gaussians inpainting results

As shown in Fig. S6, we present the single reference images for several scenes, along with multiple novel views, to validate the robust 3D consistency achieved by InFusion. Additionally, we have consolidated all scenes into a web-page included in our supplementary materials and extend an invitation for you to view them.

### C.2. Results under complex masks

As shown in Fig. S7, We also include the testing results of more difficult scenarios, which show cases of drawing



Table S2. **Comparison with depth completion methods.** "Ours\*" represents the zero-shot capability of our model, while "Ours" represents its performance after fine-tuning.

Method	NLSPN [11]	DSN [4]	ACMNet [21]	Struct-MDC [8]	LRRU [15]	CFormer [20]	BP-Net [14]	Ours*	Ours
RMSE	0.092	0.102	0.105	0.245	0.091	0.090	<b>0.089</b>	0.108	0.090

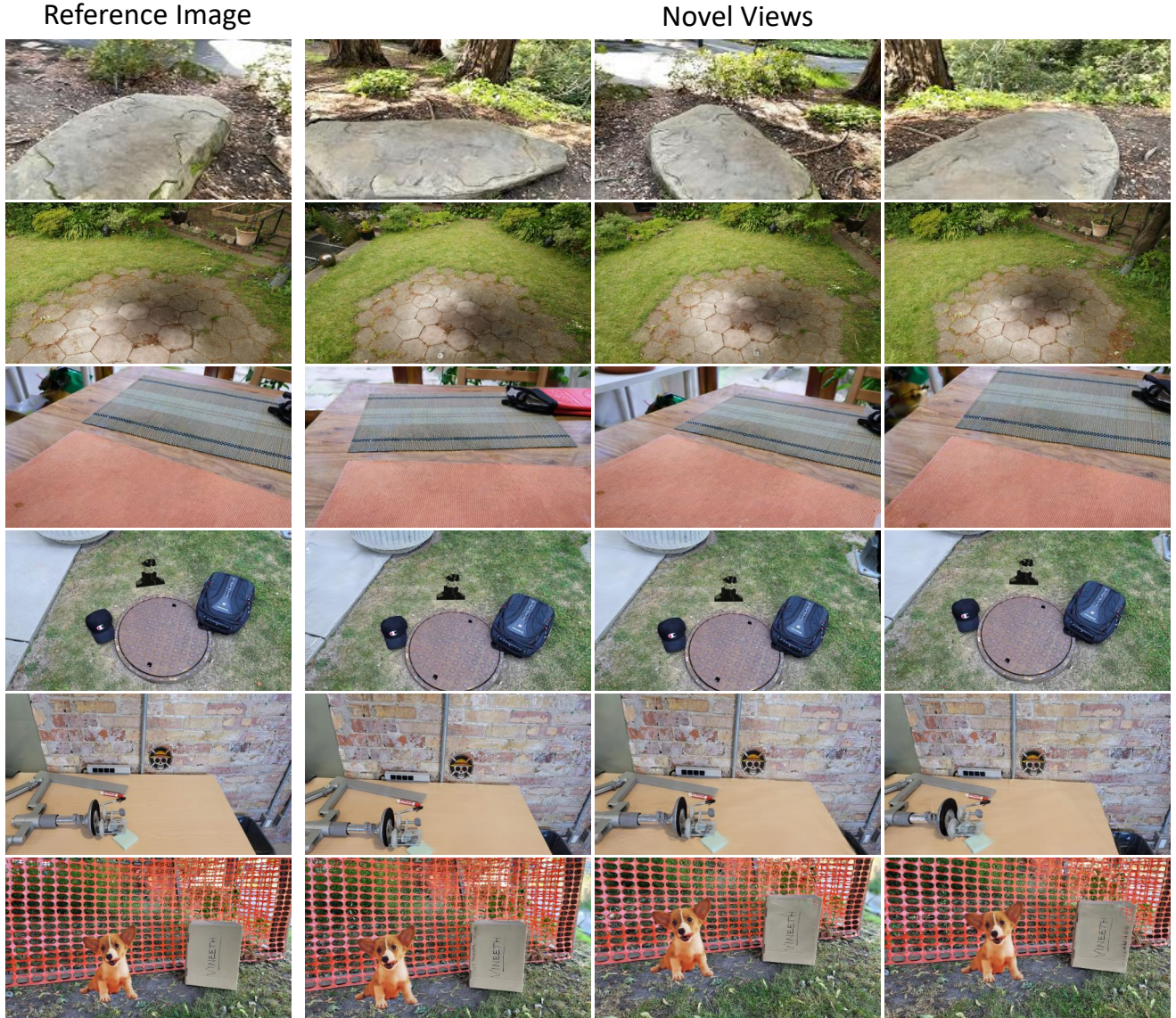


Figure S6. **Qualitative Results.** Zoom in for details. Our method exhibits sharp textures that maintain 3D coherence. We respectfully invite you to view the video featured on the webpage within our supplementary materials

random inpainting masks that go cross multiple object boundaries.

## D. More Quantitative Results

### D.1. Comparison with monocular depth estimation methods

To further evaluate our method, we compare it with existing depth estimation methods. For a fair comparison, we

retrain our model using the same 74k dataset as Marigold. Specifically, given an image with ground-truth depth, we first mask certain regions of the ground-truth depth to simulate missing data and then perform depth inpainting using our model. During testing, the proportion of known regions is randomly varied between 40% and 60%, and the masks include strokes, ellipses, rectangles, or random combinations of these shapes. For the other methods [6, 9,



Figure S7. **Random masks on multiple objects..** Please zoom in for details.

[16, 18], we use their pretrained models to estimate the depth map from the corresponding RGB image and align it to the known regions using least squares. Metrics are computed only in the masked regions to assess inpainting performance while excluding the known regions from evaluation.

Since some depth estimation methods [6, 9] employ generative approaches, we can apply training-free methods to guide the sampling based on the known depth. We select two baselines: First, at each inference step during the denoising process, we predict  $\hat{z}_0$  from  $z_t$  using DDIM. We then compute and backpropagate the loss between  $\hat{z}_0$  and the ground truth over the known areas, updating  $z_t$ . Second, we use Blend Diffusion[1] to ensure that the results match the ground truth in the known areas.

We show the quantitative results in Tab. S1. Our method incorporates known depth information during depth generation, achieving optimal performance across nearly all metrics. Furthermore, guiding generative depth estimation methods during inference with training-free approaches can sometimes improve results, but the improvements are not robust.

## D.2. Comparison with depth completion methods

LiDAR depth completion has significant potential in autonomous driving applications. However, most current depth completion methods suffer from two main drawbacks: (1) they are trained and tested on a specific dataset, lacking generalizability, and (2) they rely heavily on global information derived from sparse points, making them ineffective for complex masks required by various 3D vision downstream tasks, such as those explored in our work. We compare our method with several depth completion approaches. For a fair comparison, all methods are tested with only 0.7% of the global sparse points set as known. As shown in Tab. S2, our model achieves comparable performance without additional training and reaches state-of-the-art levels after fine-tuning for downstream dataset. Notably, our model was not specifically designed for this type of task. There is potential for further improvement by using better pre-trained models and VAEs capable of encoding sparse information more effectively. Therefore, our model also offers a new possibility for advancing depth completion tasks.

## References

- [1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, pages 18208–18218, 2022. 5
- [2] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 1
- [3] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *ICCV*, 2023. 1
- [4] Raul de Queiroz Mendes, Eduardo Godinho Ribeiro, Nicolas dos Santos Rosa, and Valdir Grassi Jr. On deep learning techniques to boost monocular depth estimation for autonomous navigation. *Robotics and Autonomous Systems*, 136:103701, 2021. 4
- [5] Bin Dou, Tianyu Zhang, Yongjia Ma, Zhaohui Wang, and Zejian Yuan. Cosseggaussians: Compact and swift scene segmenting 3d gaussians. *arXiv preprint arXiv:2401.05925*, 2024. 1
- [6] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *arXiv preprint arXiv:2403.12013*, 2024. 3, 4, 5
- [7] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024. 1
- [8] Jinwoo Jeon, Hyunjun Lim, Dong-Uk Seo, and Hyun Myung. Struct-mdc: Mesh-refined unsupervised depth completion leveraging structural regularities from visual slam. *IEEE Robotics and Automation Letters*, 7(3):6391–6398, 2022. 4
- [9] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. *CoRR*, abs/2312.02145, 2023. 1, 3, 4, 5
- [10] Kun Lan, Haoran Li, Haolin Shi, Wenjun Wu, Yong Liao, Lin Wang, and Pengyuan Zhou. 2d-guided 3d gaussian segmentation. *arXiv preprint arXiv:2312.16047*, 2023. 1
- [11] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. Non-local spatial propagation network for depth completion. In *ECCV*, pages 120–136. Springer, 2020. 4
- [12] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, abs/2307.01952, 2023. 3



- [13] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1
- [14] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. Bilateral propagation network for depth completion. In *CVPR*, pages 9763–9772, 2024. 4
- [15] Yufei Wang, Bo Li, Ge Zhang, Qi Liu, Tao Gao, and Yuchao Dai. Lrru: Long-short range recurrent updating networks for depth completion. In *ICCV*, pages 9422–9432, 2023. 4
- [16] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 3, 5
- [17] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *CoRR*, abs/2401.10891, 2024. 3
- [18] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 3, 5
- [19] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*, 2023. 1
- [20] Youmin Zhang, Xianda Guo, Matteo Poggi, Zheng Zhu, Guan Huang, and Stefano Mattoccia. Completionformer: Depth completion with convolutions and vision transformers. In *CVPR*, pages 18527–18536, 2023. 4
- [21] Shanshan Zhao, Mingming Gong, Huan Fu, and Dacheng Tao. Adaptive context-aware multi-modal network for depth completion. *IEEE TIP*, 30:5264–5276, 2021. 4