3DRealCar: An In-the-wild RGB-D Car Dataset with 360-degree Views

Xiaobiao Du [1,2,3] **Haiyang Sun** [3] **Shuyun Wang** [1] **Zhuojie Wu** [1] **Hongwei Sheng** [1]

Jiaying Ying [1] **Ming Lu** [5] **Tianqing Zhu** [4] **Kun Zhan** [3] **Xin Yu**[1]*

[1] The University of Queensland [2] University of Technology Sydney [3] Li Auto
[4] City University of Macau [5] Peking University

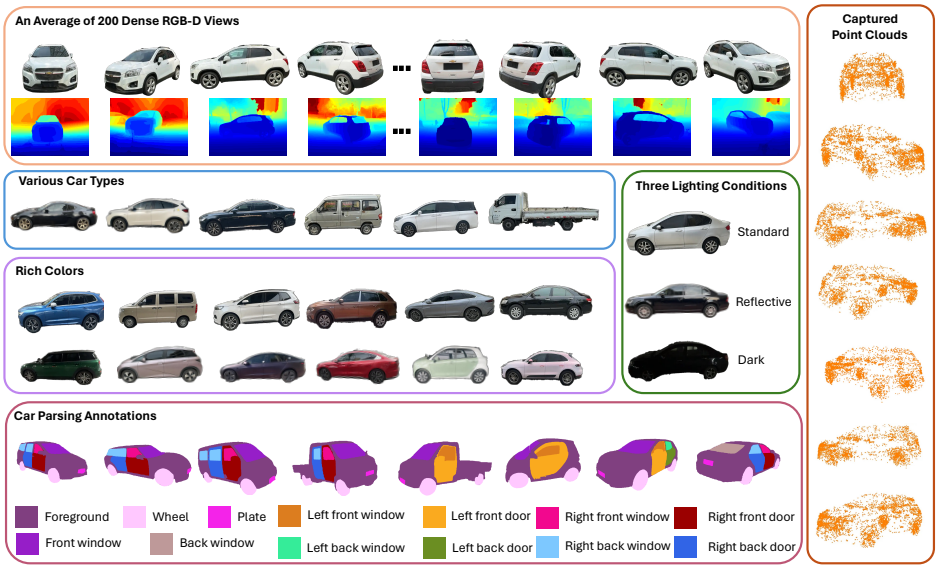

Figure 1: **Characteristics of our curated high-quality 3DRealCar dataset.** 3DRealCar contains detailed annotations for various colors, car types, brands, and even car parsing maps. In particular, our dataset contains three lighting conditions on car surfaces, bringing challenges to existing methods.

Abstract

3D cars are commonly used in self-driving systems, virtual/augmented reality, and games. However, existing 3D car datasets are either synthetic or low-quality, presenting a significant gap toward the high-quality real-world 3D car datasets and limiting their applications in practical scenarios. In this paper, we propose the first large-scale 3D real car dataset, termed 3DRealCar, offering three distinctive features. (1) **High-Volume**: 2,500 cars are meticulously scanned by 3D scanners, obtaining car images and point clouds with real-world dimensions; (2) **High-Quality**: Each car is captured in an average of 200 dense, high-resolution 360-degree RGB-D views, enabling high-fidelity 3D reconstruction; (3) **High-Diversity**: The dataset contains various cars from over 100 brands, collected under three distinct lighting conditions, including reflective, standard, and dark. Additionally, we offer detailed car parsing maps for each instance to promote research in car

*Corresponding author.

Preprint. Under review.

Table 1: **The comparison of 3D car datasets.** Lighting means the lighting conditions of the surfaces of cars. Point Cloud represents the point clouds with actual sizes in real-world scenes.

Dataset	Instances	Type	Views	Resolution	Brand	Lighting	Car Parsing	Depth	Point Cloud
SRN-Car	2151	Synthetic	250	128×128	✗	✗	✗	✗	✗
Objaverse-car	511	Synthetic	-	-	✗	✗	✗	✗	✗
MVMC	576	Real	∼10	600×450	∼40	✗	✗	✗	✗
3DRealCar (Ours)	**2500**	**Real**	**∼200**	**1920×1440**	**100+**	**3**	**13**	✓	✓

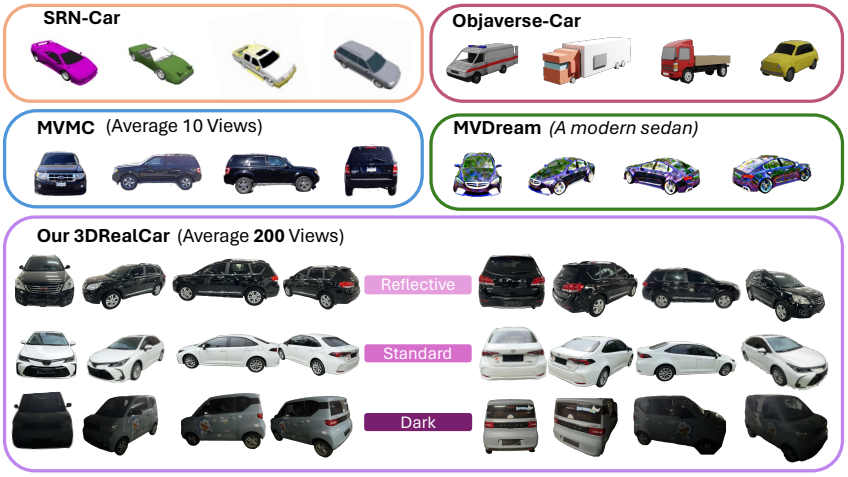

Figure 2: **Visual comparisons of 3D car datasets and the results of a 3D generative method.** Our 3DRealCar is captured in real-world scenes and contains more densely captured views. In addition, our dataset has annotations for three different lighting conditions on the car surface. We also compare a recent state-of-the-art text-to-3D model, MVDream [57] with a prompt "*a modern sedan*", demonstrating its failure to generate high-quality 3D car models.

parsing tasks. Moreover, we remove background point clouds and standardize the car orientation to a unified axis for the reconstruction only on cars without background and controllable rendering. We benchmark 3D reconstruction results with state-of-the-art methods across each lighting condition in 3DRealCar. Extensive experiments demonstrate that the standard lighting condition part of 3DRealCar can be used to produce a large number of high-quality 3D cars, improving various 2D and 3D tasks related to cars. Notably, our dataset brings insight into the fact that recent 3D reconstruction methods face challenges in reconstructing high-quality 3D cars under reflective and dark lighting conditions. Our dataset is available here.

1 Introduction

Cars, as both daily objects and vehicles, are of significant interest to researchers, especially in the field of autonomous driving. Autonomous perception systems are typically trained on daily scene datasets that are collected frequently. However, these datasets often exhibit long-tailed distributions, with far fewer instances of corner-case scenarios, like car accidents. Consequently, this imbalance leads to the autonomous perception system generalizing well in the most frequently occurring scenes. This means that the system is likely to perform poorly in rare situations, posing significant safety risks to drivers. To build a reliable system, it is essential to have a simulator that can simulate photorealistic hazardous scenes. Moreover, high-quality 3D cars are necessary for a realistic simulator.

Recent 3D car reconstruction methods [67, 77, 72] mainly reconstruct cars from self-driving datasets [61, 7, 21]. To apply reconstructed cars to real-world scenes, the reconstructed 3D car should be high-quality. However, it is very challenging to obtain such high-quality 3D cars for the following reasons: (1) Previous 3D car reconstruction methods produce low-quality 3D cars, primarily because they train on self-driving datasets with low-resolution car images and a limited number of trainable views. (2) Manually crafting a high-quality 3D car model requires specialized

artists, which is time-consuming. (3) There is no large-scale 3D real car dataset that can be utilized to produce a bulk of 3D cars.

Moreover, existing 3D car datasets are either synthetic or only contain a few posed images, as shown in Figure 2. SRN-Car [8] and Objaverse-Car [16] collect 3D car computer-aided design (CAD) models from the Internet, but these models are synthetic and contain non-photorealistic texture. Although MVMC [74] is a real car dataset, it collects only ten views on average for each car. On the contrary, our collected 3DRealCar dataset provides an average of 200 dense RGB-D views per car for high-quality 3D car reconstruction.

We also show that the recent state-of-the-art 3D generative method, MVDream [57], as depicted in Figure 2, fails to generate high-quality cars due to the multi-view inconsistency introduced by generative models [54, 2, 39, 60]. Thus, the existing 3D generation methods cannot be employed to generate high-quality 3D real car assets.

In this work, we collect a large-scale 3D real car dataset in the wild, termed 3DRealCar, which contains dense high-quality views and rich diversity. During data collection, we employ 3D scanners to scan cars parked on roadsides or parking lots, obtaining posed RGB-D images and point clouds of cars. In particular, we scan around the cars in three loops to obtain dense views. Note that we collect car data with the consent of owners. In Table 1 and Figure 1, we show our dataset possesses striking characteristics compared with previous 3D car datasets. We capture dense RGB-D images in high resolution, which promotes the reconstruction of high-quality 3D cars. Furthermore, we scan cars under three different lighting conditions, resulting in the surfaces of cars having different lighting effects, such as reflective, standard, and dark, where we denote the standard as the smooth lighting condition without obvious specular highlight. Figure 2 shows some examples of three lighting conditions in our dataset. Note that the number of instances in our dataset is the largest in existing datasets. Therefore, our collected 3DRealCar dataset has a rich diversity in terms of car types, colors, brands, and lighting conditions. We also provide car parsing map annotations with thirteen classes for each instance, which enable our dataset to be applied in car component understanding tasks.

To construct a high-quality dataset, we filter out the images that are out of focus, occluded, or blurred. To facilitate the 3D reconstruction solely on cars, we remove the point clouds of the background. We also adjust the orientation of the car facing along the x-axis before the reconstruction for controllable rendering. Based on the high-quality posed RGB-D images, point clouds, and multi-grained annotations, we can apply the dataset to various tasks related to cars. Figure 3 shows our dataset supports over 10 tasks, including several popular 2D and 3D tasks to promote the advancement of car-related research.

We leverage existing state-of-the-art methods to benchmark 3D reconstruction and car parsing tasks of our 3DRealCar dataset. We also conduct extensive experiments to demonstrate that the reflective and dark lighting conditions in our dataset are challenging to existing methods, which brings a new challenge for 3D reconstruction in awful lighting conditions. Furthermore, we demonstrate that our 3DealCar dataset can bring real-car prior and enhance existing 3D generation and downstream methods. Overall, the contributions of this work can be summarized below:

• We propose the first large-scale 3D real car dataset, named 3DRealCar, which contains 2,500 car instances and their point clouds with actual sizes in real-world scenes.

• 3DRealCar contains RGB-D images and point clouds with detailed annotations, supporting researchers to investigate various tasks in both 2D and 3D.

• We conduct 3D reconstruction and car parsing benchmarks to advance car-related tasks. Notably, we observe that existing methods face challenges under the extreme lighting conditions of 3DRealCar.

• Extensive experiments demonstrate our 3DRealCar dataset can enhance real-car prior and improve the performance of existing 3D generation and novel view synthesis methods.

2 Related Work

3D Car Datasets. There are several well-known large-scale autonomous driving datasets so far, such as Nuscenes [7], KITTI [21], Waymo [61], Pandaset [70], ApolloScape [25], and Cityscape [14]. These datasets are captured by multi-view cameras and lidars mounted on ego cars. Various works [67, 20, 41, 72] attempt to reconstruct 3D cars in these datasets. However, these methods fall short

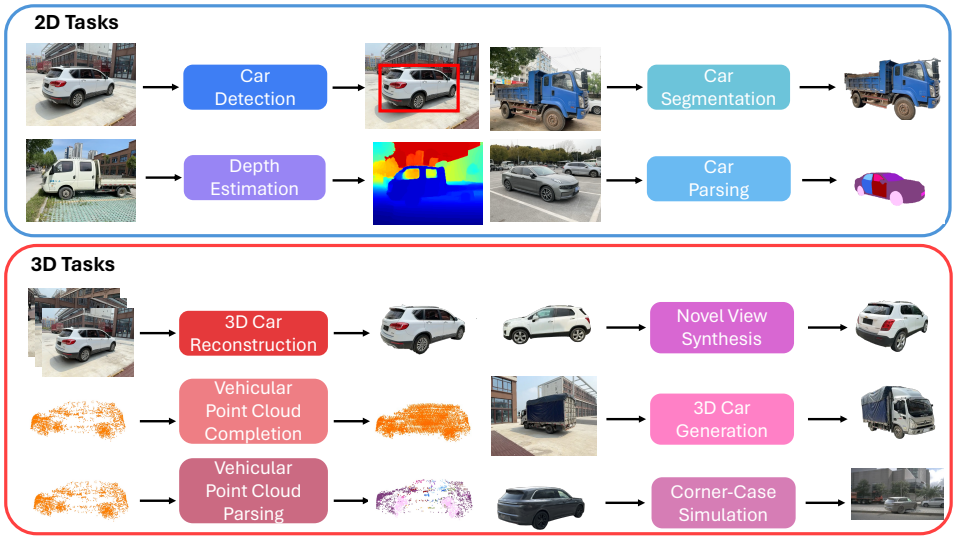

Figure 3: **The applicable tasks of our dataset.** Our proposed 3DRealCar dataset containing RGB-D images and point clouds can be applied to many popular 2D and 3D tasks.

of reconstructing high-quality 3D cars due to the lack of sufficient and dense training views. SRN-Car [8] and Objaverse [16] collect 3D car models from existing repositories and Internet sources. However, these datasets only contain synthetic cars, which cannot produce realistic textures and geometry. MVMC [74] is collected from car advertising websites, which contain a series of car images, especially multi-view images of each car. However, the views of images per car in MVMC are unposed and sparse, which is adverse to reconstructing high-quality 3D car models. In this paper, we collect a high-quality 3D real car dataset to fill the above gaps.

3D Reconstruction with Neural Field. 3D reconstruction aims to create a 3D structure digital representation of an object or a scene from its multi-view images, which is a long-standing task in computer vision. One of the most representative works in 3D reconstruction is Neural Radiance Fields (NeRFs) [46], which demonstrates promising performance for novel view synthesis. Afterward, this method inspires a new wave of 3D reconstruction methods using the volume rendering method, with subsequent works focusing on improving its quality [65, 3, 4, 22, 58, 12, 68, 6], efficiency [18, 47, 53, 59, 30, 9, 19], applying artistic effects [17, 66, 26, 75], and generalizing to unseen scenes [73, 10, 69, 28, 62, 13]. Particularly, Kilonerf [53] accelerates the training process of NeRF by dividing a large MLP into thousands of tiny MLPs. Furthermore, Mip-NeRF [3] proposes a conical frustum rather than a single ray to ameliorate aliasing. Mip-NeRF 360 [5] further improves the application scenes of NeRF to the unbounded scenes. Although these NeRF-based methods demonstrate powerful performance on various datasets, the training time always requires several hours even one day more. Instant-NGP [47] uses a multi-resolution hash encoding method, which reduces the training time by a large margin. 3DGS [30] proposes a new representation based on 3D Gaussian Splatting, which reaches real-time rendering for objects or unbounded scenes. 2DGS [24] proposes a perspective-accurate 2D splatting process that leverages ray-splat intersection and rasterization to further enhance the quality of the reconstructions. Scaffold-GS [43] proposes an anchor growing and pruning strategy to accelerate the scene coverage, which effectively reduces redundant Gaussians and improves rendering quality. However, there is not yet a large-scale 3D real car dataset so far. Therefore, we present a 3D real car dataset, named 3DRealCar in this work.

3D Generation with Diffusion Prior. Some current works [29, 49] leverage a 3D diffusion model to learn the representation of 3D structure. However, these methods lack generalization ability due to the scarcity of 3D data. To facilitate 3D generation without direct supervision of 3D data, image or multi-view diffusion models are often used to guide the 3D creation process. Notable approaches like DreamFusion [50] and subsequent works [45, 36] use an existing image diffusion model as a scoring function, applying Score Distillation Sampling SDS loss to generate 3D objects from textual descriptions. These methods, however, suffer from issues such as the Janus problem [50, 45] and overly saturated textures. Inspired by Zero123 [39], several recent works [2, 56, 42, 34, 76, 44, 38, 37, 52] refine image or video diffusion models to better guide the 3D generation by producing more

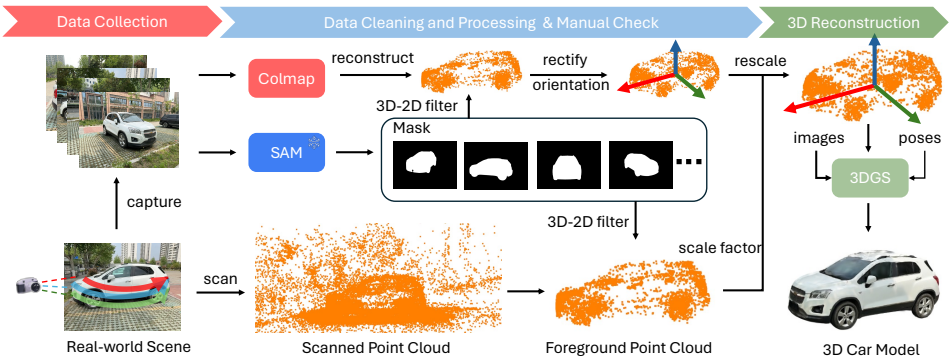

Figure 4: **Illustration of our data collection and preprocessing.** We first circle a car three times while scanning the car with a 3D scanner for the attainment of RGB-D images and its point clouds. Then we use Colmap [55] and SAM [32] to obtain poses and remove the background point clouds. Finally, we use the 3DGS [31] trained on the processed data to obtain 3D car model.

reliable multi-view images. However, these generative methods fail to generate high-quality cars due to the lack of the prior of real cars.

3 Proposed 3DRealCar Dataset

3.1 Data Collection and Annotation

As shown in Figure 4, our dataset is collected using smartphones, specifically iPhone 14 models, equipped with a 3D scanner application to scan cars for their point clouds and RGB-D images. The data collection process is conducted under three distinct lighting conditions, such as standard, reflective, and dark. These lighting conditions represent the lighting states of vehicle surfaces. It is important to note that all data collection is performed with the consent of owners. During the scanning process, the car should be stationary while we meticulously circle the car three times to capture as many views as possible. For each loop, we adjust the height of the smartphone to obtain images from different angles. Furthermore, we try our best to make sure captured images contain the entire car body without truncation. To preserve the privacy of owners, we make license plates and other private information obfuscated. To construct a high-quality dataset, we filter out some instances with blurred, out-of-focus, and occluded images. We also provide detailed annotations for car brands, types, and colors. Particularly, we provide the car parsing maps for each car with thirteen classes in our dataset as shown in Figure 1 for the advancement of car component understanding tasks.

3.2 Data Preprocessing

Background Removal. Since we only reconstruct cars for the 3D car reconstruction task, the background should be removed. Recent Segment Anything Model (SAM) [32] demonstrates powerful context recognition and segmentation performance. However, SAM needs a bounding box, text, or point as a driving factor for accurate segmentation. Therefore, we employ Grounding DINO [40] as a text-driven detector with a detection prompt with "car" for the attainment of car bounding boxes. With these bounding boxes, we use SAM to obtain the masks from captured images. The point cloud initialization is demonstrated useful for the convergence of 3D Gaussian Splatting [31]. Except for the removal of the background in 2D images, we still need to remove the background point clouds. Therefore, we first project the 3D point clouds into 2D space with camera parameters. Then, we can eliminate background point clouds with masks and save them for further processing.

Orientation Rectification. As shown in Figure 4, we utilize Colmap [55] to reconstruct more dense point clouds and obtain accurate camera poses and intrinsics because we find that the estimated poses by the 3D scanner are not accurate. However, after the removal of the background point clouds, we find that the car orientation of the point cloud is random, which leads to the subsequent render task being uncontrollable. Given camera poses $P = \{p_i\}_1^{\mathcal{N}}$, where $\mathcal{N}$ is the number of poses, we use Principal Component Analysis (PCA) [1] to obtain a PCA component $\mathcal{T} \in \mathbb{R}^{3\times3}$. The PCA component is the principal axis of the data in 3D space, which represents rotation angles to each axis.

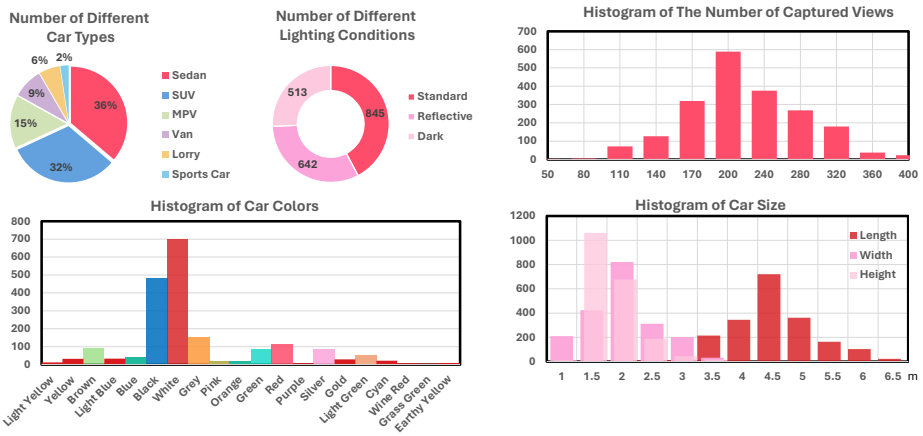

Figure 5: **The distributions of our 3DRealCar dataset.** We show distributions of car types, lighting conditions, captured views, car colors, and car size. We try our best to capture cars with various colors and types for the diversity of our dataset.

Therefore, we leverage it to rectify the postures of cars parallel to the x-axis. However, this process cannot guarantee cars facing along the x-axis. Therefore, in some failure cases, we manually interfere and adjust the orientation along the x-axis. With the fixed car orientation, we can control rendered poses for the subsequent tasks.

Point Cloud Rescaling. The size of the point clouds reconstructed by Colmap [55] does not match the real-world size, which inhibits the reconstruction of a practically sized 3D car. To address this, we calculate the bounding box of the scanned foreground point clouds to obtain its actual size in the real-world scene. Then, we rescale the rectified point clouds into the real size. In addition to the rescaling of the point clouds, we also need to adjust the camera poses. We rescale translations of camera poses using a scale factor calculated by the ratio of scanned point cloud size and Colmap point cloud size. After these rescaling processes, we use rescaled point clouds to reconstruct a 3D car model through recent state-of-the-art methods, like 3DGS [31].

3.3 Data Statistics

In our 3DRealCar, we provide detailed annotations for researchers to leverage our dataset for different tasks. During the data annotating, we discard the data with the number of views less than fifty. As we can observe in Figure 1 and 2, we collect our dataset under real-world scenes and meticulously scan dense views. Therefore, cars in our dataset possess dense views and realistic texture, which is necessary for the application in a real-world setting.

As shown in Figure 5, we conduct detailed statistical analysis to show the features of our dataset. Our dataset mainly contains six different car types, such as Sedan, SUV, MPV, Van, Lorry, and Sports Car. Among them, sedans and SUVs are common to collect in real life, so their volume dominates in our dataset. We also count the number of different lighting conditions on cars. The standard condition means the car is well-lighting and without strong specular highlights. The reflective condition means the car has strong specular highlights. Glossy materials bring huge challenges to recent 3D reconstruction methods. The dark condition means the car is captured in an underground parking so not well-lighting. To promote high-quality reconstruction, we save the captured images in high resolution (1920×1440) and also capture as many views as possible. The number of captured images per car is an average of 200. The number of views ranges from 50 to 400. To enrich the diversity of our dataset, we try our best to collect as many different colors as possible. Therefore, our dataset contains more than twenty colors, but the white and black colors still take up most of our dataset. In addition, we also show the distribution of car size, in terms of their length, width, and height. We obtain their sizes by computing the bounding boxes of the scanned point clouds. Thanks to different car types, the sizes of cars are also diverse.

Figure 6: **The simulated corner-case scenes.** We use a red rectangle to highlight the simulated vehicles. These corner-case scenes show some vehicles have potential risks to traffic safety.

Table 2: **Benchmark results on 2D car parsing of our 3DRealCar dataset.** We use recent advanced image segmentation methods [11, 23, 71, 33] to benchmark our dataset.

Method	DeepLabV3	PointRend	DDRNet	SegFormer
mIOU ↑	0.556	0.562	0.603	**0.613**
mAcc ↑	0.616	0.619	0.659	**0.663**

Table 3: **Quantitative comparisons of Dreamcraft3D [60] and its improved version by trained on our dataset.** CD denotes Chamfer Distance.

Method	**CLIP-I** ↑	**Hausdorff** ↓	**CD** ↓
Dreamcraft3D	0.812	1.572	0.587
+our dataset	**0.847**	**1.364**	**0.371**

4 Overview of 3DRealCar Tasks

4.1 2D tasks

Corner-case scene 2D Detection [64]: Given a serial of images $I = \{I_i\}_1^{\mathcal{N}}$, this task aims to detect vehicles as accurately as possible. However, in some corner cases, like car accidents, detectors sometimes fail to detect target vehicles since this kind of scene is rare or not in the training set.

2D Car Parsing [11, 23, 71, 33]: Given a serial of images $I = \{I_i\}_1^{\mathcal{N}}$ and their car parsing maps $S = \{S_i\}_1^{\mathcal{N}}$, this task aims to segment car components. With annotated parsing maps, we can train a model to understand and segment each component of cars, which is very important in self-driving.

4.2 3D Tasks

3D Reconstruction [48, 31, 24]: Given a serial of images $I = \{I_i\}_1^{\mathcal{N}}$ and matched poses $P = \{p_i\}_1^{\mathcal{N}}$, where $\mathcal{N}$ is the number of images and poses, the task of 3D reconstruction aims to reconstruct 3D model of a object or a scene. The reconstructed 3D model is usually used to render 2D images with different views for the evaluation of the performance of the 3D model.

Novel View Synthesis [39, 42, 2]: Given a serial of reference images $I^{ref} = \{I_i^{ref}\}_1^{\mathcal{N}}$, reference poses $P^{ref} = \{p_i^{ref}\}_1^{\mathcal{N}}$, target images $I^{target} = \{I_i^{target}\}_1^{\mathcal{N}}$, and target poses $P^{target} = \{p_i^{target}\}_1^{\mathcal{N}}$, recent 3D generative models, such as Zero123 [39], Syncdreamer [42], and Stable-Zero123 [2], take relative poses and reference images as inputs and generate target images. However, these models cannot generalize well to real car objects since they are trained on large-scale synthetic datasets [16, 15]. In this work, we will demonstrate that our dataset can improve the robustness of these generative models to real cars.

3D Generation [51, 60, 63]: Given a text prompt or single image, recent 3D generation methods generate 3D objects with Score Distillation Sampling (SDS) [51] and diffusion generative models [54, 39, 2]. However, these methods cannot generate high-quality 3D cars due to the lack of the prior of real cars in 3D-based diffusion models. Therefore, we would demonstrate the value of our dataset by improving recent 3D generation for real cars.

5 Experiments

5.1 Experimental Setup

Corner-case 2D Detection. In this task, we leverage the reconstructed cars to simulate rare and corner-case scenes. To be specific, we use Nuscenes [7] as background to simulate corner-case scenes with reconstructed cars and leverage recent popular detectors, like YOLOv8 [64], as detectors for evaluation. To evaluate the robustness of detectors in corner-case scenes, we use the test part of the corner-case dataset, CODA [35] as a testing set. Since we focus on the corner-case scenes of cars, so we only evaluate a car class.

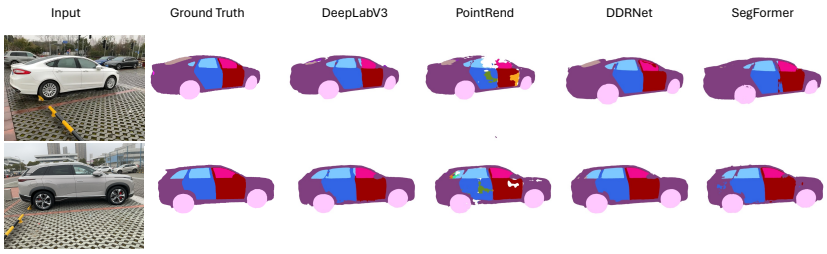

Figure 7: **Qualitative comparisons among recent advanced image segmentation methods.** We select the inputs from the testing set of our images and evaluate the capacity of car component understanding for each method.

2D Car Parsing. In this task, we utilize DeepLabV3[11], DDRNet [23], SegFormer [71], and PointRend [33] to benchmark our dataset. To be specific, we split 80% of our car parsing maps in 3DRealCar as the training set and the rest of 20% as the testing set.

3D Reconstruction. In this task, we randomly choose 100 instances from each lighting condition in our dataset and split 80% of the views per instance as the training set and the rest of 20% as the testing set. Specifically, we employ Instant-NGP [48], 3DGS [31], GaussianShader [27], and 2DGS [24] to benchmark our dataset.

Novel View Synthesis. We finetune Zero123-XL [39] on our 3DRealCar dataset to enhance its generalization to real cars. Note that since the training of diffusion-based models needs entire objects centered on images, we use the images rendered by our trained 3D models as training images.

3D Generation. In this task, we exploit Dreamcraft3D [60] as our baseline. Dreamcract3D exploits Stable-Zero123 [2] as a prior source for providing 3D generative prior. By fine-tuning Stable-Zero123 on our dataset, we enable it to obtain car-specific prior so it generalizes well to real cars.

5.2 2D Tasks

Corner-case 2D Detection. As shown in Table 4, we employ four variants of YOLOv8 serial models as our detectors for training. To evaluate the performance of models in corner-case scenes, we leverage the test part of the CODA dataset [35] as our testing set. In particular, when we increase the training simulated data from 500 to 5,000, the performance of detectors is also improved by a large margin. This phenomenon demonstrates that our simulated data is effective in improving a detector robust to corner-case scenes. We provide the visualizations of simulated corner-case scenes in Figure 6. The detailed simulation process and more visualizations can be seen in the supplementary.

Table 4: **Detection improvements by simulated data for corner-case scenes.** We report the metric by calculating mAP@0.5 on CODA dataset [35].

Method	Simulated Data					
	500	1000	2000	3000	4000	5000
YOLOv8n	0.261	0.299	0.312	0.357	0.386	**0.386**
YOLOv8s	0.336	0.371	0.366	0.403	0.413	**0.435**
YOLOv8m	0.347	0.354	0.362	0.416	0.419	**0.441**
YOLOv8l	0.358	0.375	0.381	0.411	0.413	**0.458**

2D Car Parsing. We conduct benchmarks for car parsing maps of our dataset using recent image segmentation methods, such as DeepLabV3[11], PointRend[33], DDRNet[23], and SegFormer[71]. The quantitative performance for these methods on our dataset is summarized in Table 2. Visual comparisons are provided in Figure 7. Our high-quality dataset enables these methods to achieve promising performance, highlighting its potential for application in self-driving systems.

5.3 3D Tasks

3D Reconstruction. As depicted in Table 5, we show benchmark results of recent state-of-the-art 3D reconstruction methods, such as Instant-NGP [48], 3DGS [31], GaussianShader [27], and 2DGS [24] on our dataset. To the standard lighting condition, we can find that recent methods are capable of achieving PSNR more than 27 dB, which means these methods can reconstruct relatively high-quality 3D cars from our dataset. However, the reflective and dark condition results are lower than the standard. These two parts of our 3DRealCar bring two challenges to recent 3D methods. The

Table 5: **Benchmark results on 3D reconstruction of our 3DRealCar dataset.** We present the 3D reconstruction performance of recent state-of-the-art methods in three lighting conditions, standard, reflective, and dark, respectively. The best results are highlighted.

Method	Standard			Reflective			Dark		
	PSNR ↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Instant-NGP [48]	27.31	0.9315	0.1264	24.37	0.8613	0.1962	23.17	0.9152	0.1642
3DGS [31]	27.47	**0.9367**	**0.1001**	24.58	0.8647	0.1852	**23.51**	**0.9181**	**0.1613**
GaussianShader [27]	27.53	0.9311	0.1109	**25.41**	**0.8684**	**0.1423**	23.39	0.9172	0.1631
2DGS [24]	**27.58**	0.9361	0.1084	23.19	0.8509	0.2041	22.63	0.9148	0.1681

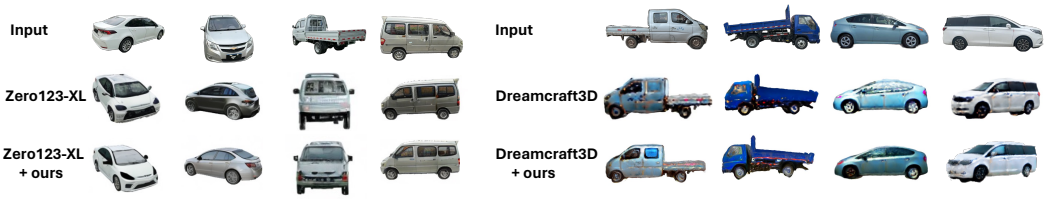

Figure 8: **Visualizations of novel view synthesis (left) and image-to-3D generation (right).** we compare the results of the recent state-of-the-art method, Zero123-XL [39], Dreamcraft3D [60], and their improvement by training on our dataset.

first challenge is the reconstruction of specular highlights. Due to the particular property of cars, materials of car surfaces are generally glossy, which means it would produce plenty of specular highlights if cars are exposed to the sun or strong light. The second challenge is the reconstruction in a dark environment. The training images captured in the dark environment lose plenty of details for reconstruction. Therefore, how to achieve high-quality reconstruction results from these two extremely lighting conditions is a challenge to recent methods. 3D visualizations can be found on our project page.

Novel View Synthesis. As illustrated in the left part of Figure 8, we show visual comparisons of Zero123-XL [39] and our improved version by training on our dataset. As we can see, given input images, we use Zero123-XL and our improved version to synthesize novel views. In this figure, we can find that Zero123-XL prefers to generate synthetic results with unrealistic texture and geometry, due to the lack of prior for real objects. In contrast, our improved version of Zero123-XL can generate photorealistic geometry and texture, which demonstrates the effectiveness of our dataset.

3D Generation. As depicted in the right part of Figure 8, we visualize 3D generation results of the recent state-of-the-art single-image-to-3D method, Dreamcraft3D [60], along with its improved version by our dataset. This figure shows that Dreamcraft3D sometimes fails to generate complete geometry or realistic texture, due to the scarcity of the real car prior. As shown in Table 3, we also show quantitative comparisons of Dreamcraft3D and its improved version. CLIP-I means the similarity of rendered images with the original input. The quantitative and qualitative results indicate our dataset significantly improves 3D generation performance.

6 Conclusion

In this paper, we propose the first large-scale high-quality 3D real car dataset, named 3DRealCar. The collected dense and high-resolution 360-degree views for each car can be used to reconstruct a high-quality 3D car. Extensive experiments demonstrate the efficacy and challenges of our 3DRealCar in 3D reconstruction. Thanks to the reconstructed high-quality 3D cars from our dataset and car-part level annotations, our dataset can be utilized to support various tasks related to cars. In addition, the benchmarking results can serve as baselines for prospective research. Although 3DRealCar currently only has car exterior views, we intend to provide both exterior and interior views in the future to further promote the reconstruction of more intact 3D cars.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] S. AI. Stable Zero123: Quality 3d object generation from single images. `https://stability.ai/news/stable-zero123-3d-generation`, 2023.

[3] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[5] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2022.

[6] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[10] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021.

[11] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[12] T. Chen, P. Wang, Z. Fan, and Z. Wang. Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15191–15202, 2022.

[13] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll. Stereo radiance fields (srf): Learning view synthesis from sparse views of novel scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021.

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[15] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[17] Z. Fan, Y. Jiang, P. Wang, X. Gong, D. Xu, and Z. Wang. Unified implicit neural stylization. In *European Conference on Computer Vision*, pages 636–654. Springer, 2022.

[18] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[19] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021.

[20] C. J. García Orellana, R. Gallardo Caballero, H. M. González Velasco, and F. J. López Aligué. Neusim: a modular neural networks simulator for beowulf clusters. In *Bio-Inspired Applications of Connectionism: 6th International Work-Conference on Artificial and Natural Neural Networks, IWANN 2001 Granada, Spain, June 13–15, 2001 Proceedings, Part II 6*, pages 72–79. Springer, 2001.

[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[22] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022.

[23] Y. Hong, H. Pan, W. Sun, and Y. Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021.

[24] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao. 2d gaussian splatting for geometrically accurate radiance fields. *arXiv preprint arXiv:2403.17888*, 2024.

[25] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 954–960, 2018.

[26] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.

[27] Y. Jiang, J. Tu, Y. Liu, X. Gao, X. Long, W. Wang, and Y. Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.

[28] M. M. Johari, Y. Lepoittevin, and F. Fleuret. Geonerf: Generalizing nerf with geometry priors. *Proceedings of the IEEE international conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] H. Jun and A. Nichol. Shap-E: Generating conditional 3D implicit functions, 2023.

[30] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

[31] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

[32] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[33] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[34] X. Kong, S. Liu, X. Lyu, M. Taher, X. Qi, and A. J. Davison. Eschernet: A generative model for scalable view synthesis. *arXiv preprint arXiv:2402.03908*, 2024.

[35] K. Li, K. Chen, H. Wang, L. Hong, C. Ye, J. Han, Y. Chen, W. Zhang, C. Xu, D.-Y. Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. *arXiv preprint arXiv:2203.07724*, 2022.

[36] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023.

[37] M. Liu, R. Shi, L. Chen, Z. Zhang, C. Xu, X. Wei, H. Chen, C. Zeng, J. Gu, and H. Su. One-2-3-45++: Fast single image to 3D objects with consistent multi-view generation and 3D diffusion. *arXiv preprint arXiv:2311.07885*, 2023.

[38] M. Liu, C. Xu, H. Jin, L. Chen, Z. Xu, H. Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023.

[39] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[40] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[41] T. Liu, H. Zhao, Y. Yu, G. Zhou, and M. Liu. Car-studio: Learning car radiance fields from single-view and unlimited in-the-wild images. *IEEE Robotics and Automation Letters*, 2024.

[42] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023.

[43] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *arXiv preprint arXiv:2312.00109*, 2023.

[44] L. Melas-Kyriazi, I. Laina, C. Rupprecht, N. Neverova, A. Vedaldi, O. Gafni, and F. Kokkinos. IM-3D: Iterative multiview diffusion and reconstruction for high-quality 3D generation. *arXiv preprint arXiv:2402.08682*, 2024.

[45] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023.

[46] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing Scenes As Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[47] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[48] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, Jan. 2022.

[49] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-E: A System for Generating 3D Point Clouds from Complex Prompts, 2022.

[50] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[51] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[52] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

[53] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021.

[54] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[55] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[56] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023.

[57] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

[58] M. Suhail, C. Esteves, L. Sigal, and A. Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022.

[59] C. Sun, M. Sun, and H.-T. Chen. Direct Voxel Grid Optimization: Super-Fast Convergence for Radiance Fields Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[60] J. Sun, B. Zhang, R. Shao, L. Wang, W. Liu, Z. Xie, and Y. Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023.

[61] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[62] M. V. T, P. Wang, X. Chen, T. Chen, S. Venugopalan, and Z. Wang. Is attention all that neRF needs? In *The Eleventh International Conference on Learning Representations*, 2023.

[63] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[64] Ultralytics. YOLOv8: A cutting-edge and state-of-the-art (sota) model that builds upon the success of previous yolo versions. `https://github.com/ultralytics/ultralytics?tab=readme-ov-file`, 2023.

[65] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *arXiv preprint arXiv:2112.03907*, 2021.

[66] C. Wang, M. Chai, M. He, D. Chen, and J. Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022.

[67] J. Wang, S. Manivasagam, Y. Chen, Z. Yang, I. A. Bârsan, A. J. Yang, W.-C. Ma, and R. Urtasun. Cadsim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation. *arXiv preprint arXiv:2311.01447*, 2023.

[68] P. Wang, Y. Liu, Z. Chen, L. Liu, Z. Liu, T. Komura, C. Theobalt, and W. Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. *CVPR*, 2023.

[69] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021.

[70] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021.

[71] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[72] Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023.

[73] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[74] J. Zhang, G. Yang, S. Tulsiani, and D. Ramanan. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. *Advances in Neural Information Processing Systems*, 34:29835–29847, 2021.

[75] K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely. Arf: Artistic radiance fields, 2022.

[76] C. Zheng and A. Vedaldi. Free3D: Consistent novel view synthesis without 3D representation. *arXiv preprint arXiv:2312.04551*, 2023.

[77] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*, 2023.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We ensure our dataset would not bring negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No]
   (b) Did you include complete proofs of all theoretical results? [No]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide a URL in the abstract for researchers to download our dataset with detailed instructions.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Training details are described in Section 4. More details can be found in supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [No] They are public and we only use them for academic study.

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Our dataset is collected under the consent of owners.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] Our dataset does not contain any potential participant risks.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] We collect our dataset by ourselves. We do not need to pay for participants.