

HUGS: Holistic Urban 3D Scene Understanding via Gaussian Splatting

Hongyu Zhou¹, Jiahao Shao¹, Lu Xu¹, Dongfeng Bai², Weichao Qiu², Bingbing Liu²
Yue Wang¹, Andreas Geiger^{3,4}, Yiyi Liao¹✉

¹ Zhejiang University ² Huawei Noah's Ark Lab ³ University of Tübingen ⁴ Tübingen AI Center

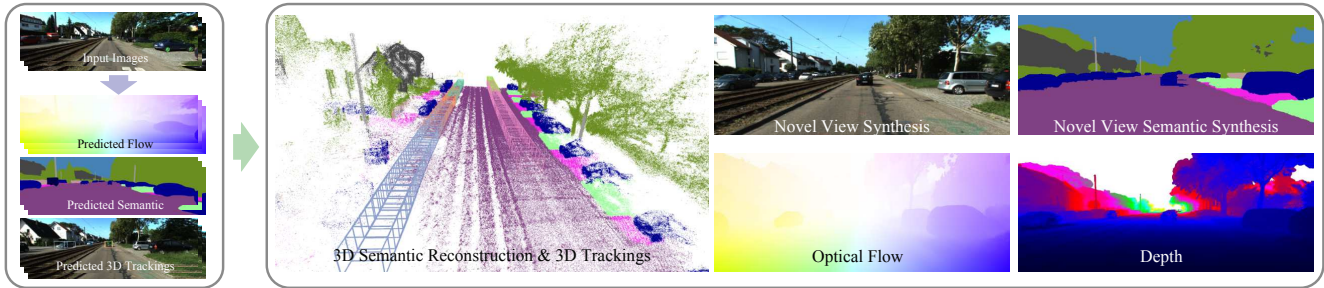


Figure 1. **Illustration.** Given posed RGB images as input, our method lifts noisy 2D & 3D predictions to the 3D space via decomposed 3D Gaussians, and enables holistic scene understanding in 2D and 3D space.

Abstract

Holistic understanding of urban scenes based on RGB images is a challenging yet important problem. It encompasses understanding both the geometry and appearance to enable novel view synthesis, parsing semantic labels, and tracking moving objects. Despite considerable progress, existing approaches often focus on specific aspects of this task and require additional inputs such as LiDAR scans or manually annotated 3D bounding boxes. In this paper, we introduce a novel pipeline that utilizes 3D Gaussian Splatting for holistic urban scene understanding. Our main idea involves the joint optimization of geometry, appearance, semantics, and motion using a combination of static and dynamic 3D Gaussians, where moving object poses are regularized via physical constraints. Our approach offers the ability to render new viewpoints in real-time, yielding 2D and 3D semantic information with high accuracy, and reconstruct dynamic scenes, even in scenarios where 3D bounding box detection are highly noisy. Experimental results on KITTI, KITTI-360, and Virtual KITTI 2 demonstrate the effectiveness of our approach. Our project page is at https://xdimlab.github.io/hugs_website.

1. Introduction

Reconstructing urban scenes is an important task in computer vision with numerous applications. Consider the creation of a photorealistic simulator for autonomous driving,

in this context, it becomes crucial to holistically represent all aspects of the scene relevant to driving. This entails tasks like synthesizing images at interpolated and extrapolated viewpoints in real-time, reconstructing 2D and 3D semantics, generating depth information, and tracking dynamic objects. To minimize sensor cost, achieving such a holistic understanding exclusively from posed RGB images holds significant value.

With the rise of neural rendering, many approaches have emerged to lift 2D information to 3D space, enabling scene understanding based solely on RGB images. Several previous works focus on reconstructing static urban scenes, achieving high-quality novel view appearance and semantic synthesis [11, 30, 51]. Another line of work addresses dynamic scenes [19, 27, 40, 46], but most of them require ground truth 3D bounding boxes of dynamic objects as input, which are costly to acquire. PNF [19] is the only method that utilizes noisy bounding boxes obtained through monocular 3D detection and tracking, where the transformations of the bounding boxes are jointly optimized during training. However, naïve joint optimization of per-frame pose transformations is prone to local minima and sensitive to the initialization. Furthermore, while existing methods are capable of rendering accurate 2D semantic labels, it is non-trivial to extract accurate semantics in 3D due to the inaccurate (inferred) 3D geometry. In addition, most of these methods are unable to achieve real-time rendering.

In this paper, We leverage predicted 2D semantic labels, optical flow, and 3D tracks, despite their inherent noise and imperfections, to achieve a holistic understanding of the dy-

✉ Corresponding author.

dynamic scenes based on RGB images (see Fig. 1). Towards this goal, we infer geometry, appearance, semantics, and motion in 3D space using a decomposed scene representation. We leverage 3D Gaussians as the scene representation, which have recently demonstrated superior novel view synthesis performance on static scenes with real-time rendering capability [17]. Specifically, we propose to decompose the scene into static regions and rigidly moving dynamic objects. We model the poses of these moving objects while adhering to the physical constraints of a unicycle model, effectively reducing the impact of noise during tracking and leading to superior performance compared to optimizing object poses individually. This allows us to reconstruct dynamic scenes even when 3D bounding box predictions are highly noisy. Further, we extend 3D Gaussian Splatting to model camera exposure and explore initialization on dynamic scenes, enabling state-of-the-art novel view synthesis performance on urban scenes. Additionally, we incorporate semantic information into 3D Gaussians, enabling the rendering of semantic maps and the extracting of 3D semantic point clouds. Finally, we integrate the RGB, semantics and optical flow to jointly supervise the model training, and investigate the interaction between these image cues to improve the performance of the scene understanding tasks.

Our main contributions are as follows: 1) Our method addresses the task of dynamic 3D urban scene understanding by extending Gaussian Splatting to model additional modalities, including semantic, flow, and camera exposure, as well as dynamic objects. 2) We achieve the decomposition of static and multiple dynamic objects from sparse urban images and noisy labels by incorporating physical constraints, omitting the requirement of ground truth 3D bounding boxes for reconstructing dynamic scenes. 3) Our method achieves state-of-the-art performance on various benchmarks, including novel view appearance and semantic synthesis, as well as 3D semantic reconstruction.

2. Related Work

3D Scene Understanding: Understanding urban scenes from various aspects has been considered essential for autonomous driving. Numerous techniques have focused on predicting semantic labels [5, 9, 35], depth maps [10, 28], and optical flows [42] solely from 2D input images. While these methods have demonstrated impressive accuracy within the confines of the 2D space, they often fall short of grasping a profound understanding of the underlying 3D environment. Consequently, this limitation can hinder the multi-view consistency of their predictions. Another line of approach suggests conducting semantic scene understanding solely based on 3D input [29, 31]. This approach heavily relies on LiDAR input, which is known to be costly and resource-intensive to collect.

More recently, a particular approach has emerged, aiming to elevate 2D information to the 3D space to facilitate scene understanding within the 2D image domain. This advancement is made possible through the utilization of differential neural rendering techniques, such as NeRF (Neural Radiance Fields) [25]. Numerous NeRF-based approaches [2–4, 14, 26, 34, 38] have made significant advancements in terms of both quality and efficiency. Furthermore, some other techniques have empowered NeRF with improved scene understanding capabilities. Semantic NeRF [52] first proposes the lifting of noisy 2D annotations to the 3D space based on NeRF. Significant progress has been achieved through the efforts of the following works [37, 44, 49]. While these methods have shown promising results, they are currently limited to dense input viewpoints within indoor scenes and are only applicable to static environments. In this study, our focus lies in dynamic 3D scene understanding specifically tailored to urban settings, achieved by lifting 2D information to the 3D space.

Urban Scene Reconstruction: Numerous studies have been conducted to reconstruct urban scenes using various methods. These methods can be categorized into three classes: point-based [1, 32], mesh-based [12, 20] and NeRF-based [15, 22, 24, 30, 33, 39, 51]. While point-based and mesh-based methods demonstrate faithful reconstructions, they struggle to recover all aspects of the scene, especially when it comes to high-quality appearance modeling. In contrast, NeRF-based models allow for reconstructing scene appearance and enable high-quality rendering of novel viewpoints. However, these approaches are primarily designed for static scenes, lacking the ability to handle dynamic urban environments. In this study, our focus lies in addressing the challenges of dynamic urban scenes.

Several methods have also been developed to address the reconstruction of dynamic urban scenes. Many of these approaches rely on the availability of accurate 3D bounding boxes for moving objects in order to separate the dynamic elements from the static components, as seen in NSG [27], MARS [40] and UniSim [46]. PNF [19] takes a different approach by leveraging monocular-based 3D bounding box predictions and proposes a joint optimization of object poses during the reconstruction process. However, our experimental observations indicate that the straightforward optimization of object poses yields unsatisfactory results due to the absence of physical constraints. Another method, SUDS [36], avoids the use of 3D bounding boxes by grouping the scene based on learned feature fields. However, the accuracy of this approach lags behind. In parallel, the concurrent work EmerNeRF [45] follows a similar idea to SUDS by decomposing the scene purely into static and dynamic components. In our research, we possess the capability to further decompose individual dynamic objects within the scene and estimate their motion.

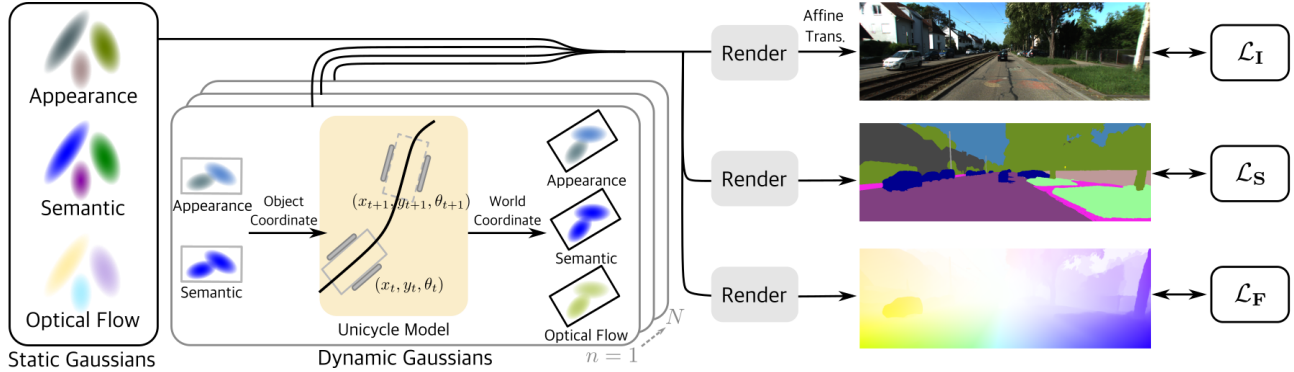


Figure 2. **Method Overview.** We decompose the scene into static regions and N rigidly moving dynamic objects. Each dynamic object is represented using 3D Gaussians in its canonical space and then transformed to the world coordinates based on transformations constrained by a unicycle model. We use N unicycle models of different parameters to individually represent the motion of N dynamic objects. Each 3D Gaussian encompasses information about appearance and semantics, whereas the optical flow can be obtained by calculating the Gaussian center’s motion, enabling the rendering of RGB images, semantic maps, and optical flow within a unified model. Our method is supervised using RGB images, noisy 2D semantic labels, and noisy optical flow, denoted as \mathcal{L}_I , \mathcal{L}_S , and \mathcal{L}_F , respectively.

Gaussian Splatting: 3D Gaussians are demonstrated as a powerful scene representation for novel view synthesis. While the original 3D Gaussian Splatting [17] primarily focuses on static scenes, subsequent research has extended this approach to handle dynamic scenes. Dynamic 3D Gaussians [23] necessitates a substantial number of training views accompanied by ground truth masks. Other studies [43, 47, 48, 53] have also attempted to decompose 3D Gaussians into static and dynamic components, without further decomposing multiple dynamic objects. In our work, we strive to achieve the decomposition of each individual dynamic object while being capable of learning such decomposition from sparse urban images and noisy labels.

3. Method

Fig. 2 illustrates our proposed method, HUGS. Our algorithm takes as input posed images of a dynamic urban scene. We decompose the scene into static and dynamic 3D Gaussians, with the motion of dynamic vehicles being modeled via a unicycle model. The 3D Gaussians represent not only appearance but also semantic and flow information, allowing for rendering the RGB images, semantic labels, as well as optical flow through volume rendering.

3.1. Decomposed Scene Representation

We assume that the scene is composed of static regions and a total of N dynamic vehicles exhibiting rigid motions. Static regions are represented using static Gaussians in the world coordinate system. Each of the N dynamic vehicles is modeled using dynamic Gaussians in a canonical coordinate system along with a set of rigid transformations $\{(\mathbf{R}_t^n, \mathbf{t}_t^n)\}_{t=1}^T$ with t denoting the timestamp.

Static and Dynamic 3D Gaussians: Following Gaussian

Splatting [17], we model both static and dynamic regions using 3D Gaussians. Each Gaussian is defined by a 3D covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$ and a 3D position $\mu \in \mathbb{R}^3$, as well as an opacity $\alpha \in \mathbb{R}^+$:

$$G(\mathbf{x}) = \alpha \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right) \quad (1)$$

In addition, each Gaussian represents a color vector $\mathbf{c} \in \mathbb{R}^3$ parameterized as SH coefficients. In this work, we propose to additionally model semantic logits $\mathbf{s} \in \mathbb{R}^5$ of each 3D Gaussian, allowing for rendering 2D semantic labels. Furthermore, we can naturally obtain a rendered optical flow $\mathbf{f}_{t_1 \rightarrow t_2} \in \mathbb{R}^2$ for each 3D Gaussian by projecting the 3D position μ to the image space at two different timestamps, t_1 and t_2 , and calculating the motion.

Unicycle Model: We parameterize the transformations $(\mathbf{R}_t, \mathbf{t}_t)$ following the unicycle model¹. The state of a unicycle model is parameterized by three elements: (x_t, y_t, θ_t) , where x_t and y_t represent the first two axes of \mathbf{t} with $\mathbf{t}_t = [x_t, y_t, z_t]$, and θ_t is the yaw angle of \mathbf{R}_t . To adapt the continuous unicycle model to discrete frames, we derive the calculus of the unicycle model for the vehicle transition from timestamp t to $t + 1$ as follows:

$$\begin{aligned} x_{t+1} &= x_t + \frac{v_t}{\omega_t}(\sin \theta_{t+1} - \sin \theta_t) \\ y_{t+1} &= y_t - \frac{v_t}{\omega_t}(\cos \theta_{t+1} - \cos \theta_t) \\ \theta_{t+1} &= \theta_t + \omega_t \end{aligned} \quad (2)$$

Here, v_t represents the forward velocity, and ω_t is the angular velocity. This model integrates physical constraints

¹While it is more accurate to model vehicles using a bicycle model, we observe that using the simpler unicycle model is sufficient for our task.

when compared to directly optimizing the transformations of dynamic vehicles at every frame independently, thus enabling smoother motion modeling of moving objects and making them less prone to local minima.

While it is possible to define an initial state (x_1, y_1, θ_1) and derive the following states recursively based on velocities, v_t and ω_t , such a recursive parameterization is challenging to optimize. In practice, we define a set of trainable states $\{(x_t, y_t, \theta_t)\}_{t=1}^T$ along with trainable velocities $\{v_t, \omega_t\}_{t=1}^{T-1}$, and add a regularization term to ensure that the vehicle’s states adhere to the characteristics of a unicycle model in Eq. 2. The regularization terms will be described in Section 3.3. Additionally, we model the vertical locations of the vehicle, $\{z_t\}_{t=1}^T$, as optimizable parameters.

3.2. Holistic Urban Gaussian Splatting

Given the HUGS representation specified above, we are able to render images, semantic maps and optical flow to supervise the model or make predictions at inference time. We now elaborate on the rendering of each modality.

Novel View Synthesis: The combination of static and dynamic Gaussians can be sorted and projected onto the image plane via α -blending:

$$\pi : \quad \mathbf{C} = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (3)$$

Here, α'_j is determined by the projected 2D Gaussian and the 3D opacity α , see supplement for details.

In contrast to single-object scenes, urban scenes typically involve more complex lighting conditions and the images are usually captured with auto white balance and auto exposure. NeRF-based methods [24] typically feed a per-frame appearance embedding along with the 3D positions into a neural network to compute the color, thereby compensating exposure. However, when working with 3D Gaussians, there is no neural network capable of processing appearance embeddings. Inspired by Urban Radiance Field [30], we generate an exposure affine matrix for each camera by mapping the camera’s extrinsic parameters to an affine matrix $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ and vector $\mathbf{b} \in \mathbb{R}^3$ via a small MLP:

$$\tilde{\mathbf{C}} = \mathbf{A} \times \mathbf{C} + \mathbf{b} \quad (4)$$

We demonstrate that modeling the exposure improves rendering quality in the experimental section.

Semantic Reconstruction: Similarly to Eq. 3, we can obtain 2D semantic labels via α -blending based on the 3D semantic logit \mathbf{s} :

$$\pi : \quad \mathbf{S} = \sum_{i \in \mathcal{N}} \text{softmax}(\mathbf{s}_i) \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (5)$$

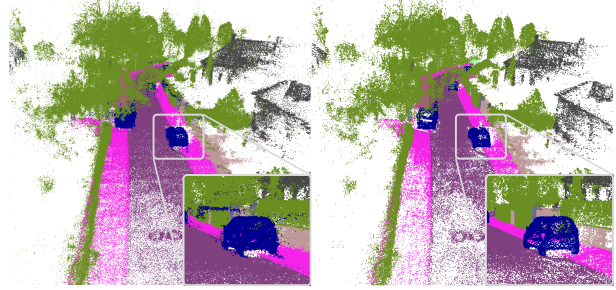


Figure 3. **3D Semantic Reconstruction.** Comparison between applying softmax to accumulated 2D semantic logits (left) and to 3D semantic logits (right). Normalizing semantic logits in 3D space clearly reduces floaters and yields better 3D semantic reconstruction than the 2D normalization counterpart.

Note that we perform the softmax operation on 3D semantic logits \mathbf{s}_i prior to α blending, in contrast to most existing methods that apply softmax to 2D semantic logits $\tilde{\mathbf{S}}$ obtained by accumulating unnormalized 3D semantic logits \mathbf{s}_i [11, 52]. As shown in Fig. 3, applying softmax in 2D space leads to noisy 3D semantic labels. This is due to the fact that 2D space softmax can produce accurate 2D semantics by adjusting the scale of the 3D semantic logits, allowing a single sampled point with a substantial logit value to significantly influence the volume rendering outcome. For example, an undesired floating point labeled with “car” may not be penalized despite the target rendered label is “tree”, as long as there is a 3D Gaussian providing a large logit value of “tree” along this ray. Our solution instead removes such floaters by normalizing logits in 3D space. See supplement for more quantitative and qualitative details.

Optical Flow: The 3D Gaussian representation also enables the rendering of optical flow. Given two timestamps t_1 and t_2 , we first calculate the optical flow of each 3D Gaussian’s center μ as $\mathbf{f}_{t_1 \rightarrow t_2}$. Specifically, we project μ to the 2D image space based on the camera’s intrinsic and extrinsic parameters:

$$\mu'_1 = \mathbf{K}[\mathbf{R}_{t_1}^{\text{cam}}; \mathbf{t}_{t_1}^{\text{cam}}] \mu, \quad \mu'_2 = \mathbf{K}[\mathbf{R}_{t_2}^{\text{cam}}; \mathbf{t}_{t_2}^{\text{cam}}] \mu, \quad (6)$$

and then calculate the motion vector as $\mathbf{f}_{t_1 \rightarrow t_2} = \mu'_2 - \mu'_1$. Next, we render the optical flow via accumulate the optical flows via volume rendering:

$$\pi : \quad \mathbf{F} = \sum_{i \in \mathcal{N}} \mathbf{f}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (7)$$

Note that this rendering process assumes that any pixel of a 2D Gaussian splat shares the same optical flow direction as the corresponding Gaussian center but with scaled magnitude. While this is indeed a simplified approximation, we observe this to work well in practice.

In our experiments, we demonstrate that supervising the rendered optical flow with pseudo ground truth helps to im-

prove the performance of the geometry in terms of rendered depth maps. This is due to the fact that flow provides explicit pixel correspondences, which is inherently supervising the underlying surface location.

3.3. Loss Functions

We leverage pre-trained recognition models to provide noisy 2D semantic and instance predictions, noisy 2D optical flow, as well as noisy 3D tracking results. These easy-to-obtain predictions are critical to enable RGB-only holistic scene understanding in both 2D and 3D space, without relying on LiDAR input or 3D semantic supervision.

Image-based Losses: Our model is supervised with the ground truth images using a combination of L1 and SSIM losses. Let $\tilde{\mathbf{I}}$ denote the rendered image and $\hat{\mathbf{I}}$ the ground truth, our rendering loss is defined as follows:

$$\mathcal{L}_{\mathbf{I}} = (1 - \lambda_{SSIM}) \|\hat{\mathbf{I}} - \tilde{\mathbf{I}}\|_1 + \lambda_{SSIM} \text{SSIM}(\hat{\mathbf{I}}, \tilde{\mathbf{I}}) \quad (8)$$

We additionally apply the cross-entropy loss to the rendered semantic label wrt. pseudo-2D semantic segmentation ground truth $\hat{\mathbf{S}}$:

$$\mathcal{L}_{\mathbf{S}} = - \sum_{k=0}^{S-1} \hat{\mathbf{S}}_k \log(\mathbf{S}_k) \quad (9)$$

Similarly, we leverage pseudo optical flow ground truth $\hat{\mathbf{F}}$ to supervise the rendered optical flow using:

$$\mathcal{L}_{\mathbf{F}} = \|\hat{\mathbf{F}} - \mathbf{F}\|_1 \quad (10)$$

While 3D Gaussians can enable the rendering of optical flow without any supervision, we observe artifacts in the rendered flow without supervision. Further, the optical flow supervision yields an improvement in the depth maps as shown in our ablation study.

Unicycle Model Losses: We use a unicycle model to guide the noisy 3D bounding box predictions:

$$\mathcal{L}_{\mathbf{t}} = \sum_t \|x_t - \hat{x}_t\|_2 + \sum_t \|y_t - \hat{y}_t\|_2 \quad (11)$$

where \hat{x}_t and \hat{y}_t are the x and y locations of a noisy 3D bounding box at timestamp t .

As mentioned earlier, we parameterize the vehicle’s states (x_t, y_t, θ_t) and the velocities v_t, ω_t as learnable parameters. Hence, we add the following regularization to make the states adhere to the unicycle model as follows:

$$\begin{aligned} \mathcal{L}_{uni} = & \sum_t \|x_{t+1} - x_t - \frac{v_t}{\omega_t} (\sin \theta_{t+1} - \sin \theta_t)\| + \\ & \sum_t \|y_{t+1} - y_t + \frac{v_t}{\omega_t} (\cos \theta_{t+1} - \cos \theta_t)\| + \\ & \sum_t \|\theta_{t+1} - \theta_t - \omega_t\| \end{aligned} \quad (12)$$

In addition, we regularize the acceleration of the forward velocity v_t and angular velocity ω_t to be smooth:

$$\begin{aligned} \mathcal{L}_{reg} = & \sum_t \|v_{t+1} + v_{t-1} - 2v_t\|_2 + \\ & \sum_t \|\theta_{t+1} + \theta_{t-1} - 2\theta_t\|_2 \end{aligned} \quad (13)$$

The total loss can be summarized as follows:

$$\mathcal{L} = \mathcal{L}_{\mathbf{I}} + \lambda_{\mathbf{S}} \mathcal{L}_{\mathbf{S}} + \lambda_{\mathbf{F}} \mathcal{L}_{\mathbf{F}} + \lambda_{\mathbf{t}} \mathcal{L}_{\mathbf{t}} + \lambda_{uni} \mathcal{L}_{uni} + \lambda_{reg} \mathcal{L}_{reg} \quad (14)$$

3.4. Implementation Details

Initialization: While 3D Gaussian Splatting is not highly sensitive to the initialization, better initialization can yield better performance. We utilize the dense point cloud obtained from COLMAP for initialization by default. When the ego-vehicle is static, we use random initialization.

Pseudo-GTs: We utilize InverseForm [5] to generate pseudo ground truth for semantic segmentation. For initializing the unicycle model, we employ a monocular-based method, QD-3DT [16], to acquire pseudo ground truth for 3D bounding boxes and tracking IDs at each training view. For optical flow, we use Unimatch [41] to obtain pseudo ground truth.

Training: We train the model for 30,000 iterations on dynamic scenes. For the KITTI-360 leaderboard, we perform early stopping at 15,000 iterations. Following [17], we adopt the approach of setting the weight parameter λ_{SSIM} to 0.2. Furthermore, we assign weights $\lambda_{\mathbf{S}}$ and $\lambda_{\mathbf{F}}$ as 0.01, while $\lambda_{\mathbf{t}}$, λ_{uni} and λ_{reg} are set as 0.1. The learning rate of the unicycle model parameters progressively decreases during training.

Time Consuming: Our approach can converge within 30 minutes and achieve inference at a speed of approximately 93 fps on a single NVIDIA RTX 4090. While NSG and MARS inference at a speed of less than 1 fps. A speed breakdown of our method is provided in the supplement.

4. Experiments

Datasets: We perform a range of experiments to assess the performance of our model across various tasks, such as novel view synthesis, novel semantic synthesis, and 3D semantic reconstruction. These experiments are conducted using the KITTI [13], Virtual KITTI 2 (vKITTI) [7], and KITTI-360 datasets [21]. We apply 50% dropout rate following existing evaluation protocols [21, 40] on all of these datasets.

Baselines: We evaluate the dynamic scene novel view synthesis task by comparing our method with NSG [27] and



Figure 4. **Qualitative Comparison** on KITTI and vKITTI. We use monocular-based 3D bounding box predictions for KITTI, and manually jittered 3D bounding boxes for vKITTI. We zoom in on a patch of a dynamic object for each KITTI scene.

	KITTI Scene02			KITTI Scene06			vKITTI Scene02			vKITTI Scene06		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
NSG [27]	23.00	0.664	0.373	23.78	0.717	0.234	21.40	0.689	0.376	20.60	0.719	0.255
MARS [40]	23.30	0.731	0.139	25.09	0.856	0.083	22.67	0.882	0.128	21.67	0.856	0.134
Ours	25.42	0.821	0.092	28.20	0.919	0.027	26.21	0.911	0.040	26.65	0.921	0.030

Table 1. **Novel View Synthesis on Dynamic Scenes** with predicted or noisy 3D trackings.

MARS [40], which are two open-source methods for dynamic urban scenes. Additionally, we compare the static novel view appearance and semantic synthesis task with mip-NeRF [2], PNF [19], and MARS [40]. Furthermore, we assess the quality of 3D semantic scene reconstruction by comparing it with Semantic Nerfacto [34].

Evaluation Metrics: For *novel view synthesis*, we adopt the default setting for quantitative assessments, including the evaluation of PSNR, SSIM and LPIPS [50]. Regarding *novel view semantic synthesis*, we follow KITTI-360 [21], which reports the mean Intersection over Union on class ($mIoU_{cls}$) and category ($mIoU_{cat}$), respectively. Further, we evaluate our performance on *3D Semantic Segmentation* against a ground truth semantic LiDAR point cloud, measuring both geometric reconstruction quality and semantic accuracy. The geometric quality is evaluated as the chamfer distance between two point clouds, including completeness and accuracy, whereas the semantic accuracy is also measured using $mIoU_{cls}$. In our ablation study, we evaluate *3D tracking* performance by measuring the rotation and translation error e_R and e_t of our optimized 3D bounding boxes wrt. the ground truth.

4.1. Novel View Synthesis

We first evaluate HUGS for novel view synthesis on various datasets including dynamic and static scenes. For dynamic scenes, we leverage noisy 3D bounding box predictions as input, instead of using the ground truth. Despite not being

our main focus, we include a comparison of using ground truth 3D bounding boxes in the supplement.

Dynamic Scene with Noisy 3D Bounding Boxes: Following [27, 40], we evaluate our performance on dynamic scenes of the KITTI and vKITTI datasets. In contrast to these methods that leverage ground truth poses, we investigate a more practical scenario where the bounding boxes are generated by a monocular-based 3D tracking algorithm, QD-3DT [16], in Table 1. Here, the predicted 3D bounding boxes are only provided for training views, as testing views should not be used as inputs for the tracking model. In experiments where the unicycle model is not utilized, the bounding boxes of testing views are obtained through linear interpolation from neighbour training views. Where the unicycle model is used, the bounding boxes of testing views are computed using Eq. 2. For vKITTI, there is no pre-trained monocular tracking algorithm. We hence jitter the ground truth poses to simulate noisy monocular predictions, with an average noise of 0.5 meters in translation and 5 degrees in rotation. Our model’s robustness wrt. various levels of noise will be analyzed in the ablation study.

Table 1 demonstrate that our method consistently outperforms against the baselines. Note, that QD-3DT yields reasonable predictions on the KITTI dataset². Hence, NSG and MARS reconstruct the dynamic objects reasonably well, but with more blurriness and artifacts (see Fig. 4), as they do

²In fact, following the evaluation protocol of MARS, the sequences we evaluate on are used as training sequences for QD-3DT.

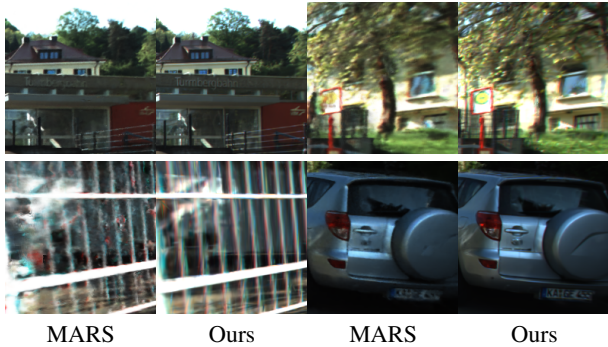


Figure 5. **Details Qualitative Comparison** with MARS on KITTI-360 Leaderboard.

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	mIoU $_{cls}$ \uparrow	mIoU $_{cat}$ \uparrow
mip-NeRF [2]	21.54	0.778	0.365	48.25	67.47
PNF [19]	22.07	0.820	0.221	73.06	84.97
MARS [40]	23.09	0.857	0.174	-	-
Ours	23.38	0.870	0.121	72.65	85.64

Table 2. **Novel View Semantic and Appearance Synthesis** on KITTI-360.

not model the optimization of the object poses. In contrast, our method allows for reconstructing dynamic objects with sharp details, not only in cases of minor pose error on the KITTI dataset but also on the vKITTI dataset with more severe noise.

Static Scene Leaderboard: We further evaluate our performance on the KITTI-360 leaderboard, which contains 5 static sequences. Our method achieves state-of-the-art performance on the leaderboard as in Table 2 (left), demonstrating the effectiveness of the 3D Gaussian representation in modeling complex urban scenes. As we will discuss in the ablation study, incorporating the affine transform to model camera exposure is important for reaching high fidelity. Fig. 5 shows the qualitative comparison of our proposed method to another top-ranking method, MARS, on the leaderboard.

4.2. Semantic and Geometric Scene Understanding

Next, we evaluate our model on various semantic and geometric scene understanding tasks on the KITTI-360 dataset.

Novel View Semantic Synthesis: Our holistic representation also enables novel view semantic synthesis. Hence, we submit our novel view semantic synthesis performance to the KITTI-360 leaderboard for comparison as well, see Table 2 (right). Despite not leveraging category-level prior as done in previous work [19], our approach achieves comparable performance to the SOTA [19] as shown in Fig. 6.

3D Semantic Scene Reconstruction: While existing 2D-to-3D semantic lifting methods solely evaluate their performance in the 2D image space, we further evaluate our per-

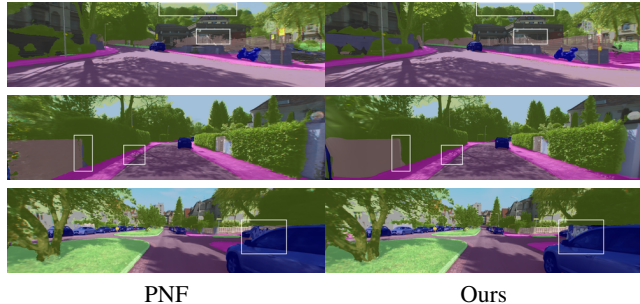


Figure 6. **Qualitative Comparison** with PNF on KITTI-360 Leaderboard.

	acc. \downarrow	comp. \downarrow	mIoU $_{cls}$ \uparrow
Semantic Nerfacto	1.508	24.28	0.055
Ours	0.233	0.214	0.505

Table 3. **3D Semantic Reconstruction** on KITTI-360. Note that all metrics are calculated in 3D space.

formance in the 3D space to examine the underlying 3D geometry. To this goal, we leverage the ground truth LiDAR points provided by the KITTI-360 dataset for evaluation. With each Gaussian possessing semantic information, we can obtain a semantic point cloud by extracting the Gaussian’s center μ and its semantic label. We evaluate the geometric quality and semantic accuracy of this semantic point cloud in Table 3. We compare our method with Semantic Nerfacto [34], a Semantic NeRF implemented using a more advanced backbone, as the state-of-the-art novel view semantic synthesis method, PNF, in Table 2 is not open-source. For this baseline, we extract a semantic point cloud by specifying a threshold to the density field. While Semantic Nerfacto enables rendering faithful 2D semantic labels as shown in the supplement, the underlying 3D semantic point cloud is significantly worse in comparison. The Gaussian based representation instead allows for extracting a much more accurate semantic point cloud in comparison.

4.3. Scene Editing

Our decomposed scene representation enables various downstream applications. Our method allows for decomposing foreground moving objects from the background as shown in Fig. 7. Further, we can edit the scene by swapping dynamic objects, or manipulating their rotation and translations, see Fig. 8.

4.4. Ablation Study

We conduct ablation studies on dynamic and static scenes, respectively.

Dynamic Scene: As KITTI provides accurate 3D bounding box ground truth, we ablate the effectiveness of our unicycle model on KITTI by manually adding noise to the 3D

	KITTI (5% noise)					KITTI (10% noise)					KITTI (20% noise)				
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	$e_R\downarrow$	$e_t\downarrow$	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	$e_R\downarrow$	$e_t\downarrow$	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	$e_R\downarrow$	$e_t\downarrow$
w/o opt., w/o uni.	23.83	0.878	0.062	0.031	0.027	22.16	0.861	0.079	0.063	0.106	20.28	0.835	0.101	0.125	0.425
w/ opt., w/o uni.	24.80	0.897	0.038	0.022	0.051	22.75	0.879	0.056	0.054	0.130	20.56	0.855	0.081	0.135	0.612
w/ opt., w/ uni. (Ours)	28.78	0.928	0.023	0.017	0.022	26.66	0.908	0.032	0.037	0.035	23.59	0.875	0.061	0.081	0.176

Table 4. Ablation Study on Dynamic Scenes of KITTI.

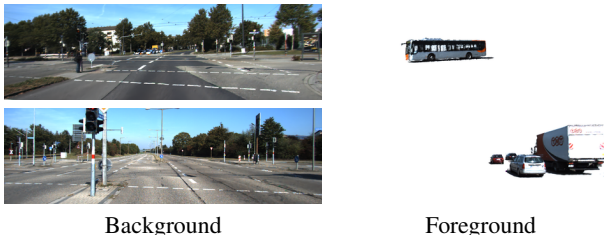


Figure 7. **Scene Decomposition** on KITTI. Our approach enables clear decomposition of foreground and background.



Figure 8. **Scene Editing** on KITTI. Our decomposed scene representation enables replacing dynamic objects (1st row) and moving dynamic objects around (2nd & 3rd rows).

bounding boxes and evaluate both the novel view synthesis results and the tracking performance, see Table 4. In this experiment, we compare our full model to two variants, i.e., using the noises without optimization (w/o opt., w/o uni.), and performing naïve per-frame optimization without using the unicycle model (w/ opt., w/o uni.). The results validate the effectiveness of the unicycle model, which obviously improves the rendering quality and 3D tracking accuracy. Qualitative results in Fig. 9 further verify the effectiveness of our unicycle model in enabling accuracy object reconstruction given noisy 3D bounding boxes.

Static Scene: We further study the effect of different components on three static scenes of KITTI-360 in Table 5. This allows us to ablate design choices without mixing up the impact of dynamic objects. The results indicate the significance of exposure modeling, which is particularly important for scenes with strong exposure variance. The semantic and flow losses have little contribution in improving novel view synthesis. It is rational as imposing a constraint on the semantic or flow does not necessarily contribute to appear-

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	Depth \downarrow
w/o Affine transform	24.18	0.827	0.083	–
w/o \mathcal{L}_S	24.47	0.831	0.081	0.892
w/o \mathcal{L}_F	24.45	0.831	0.080	1.031
Ours	24.52	0.833	0.081	0.872

Table 5. Ablation Study on Static Scenes on KITTI-360.

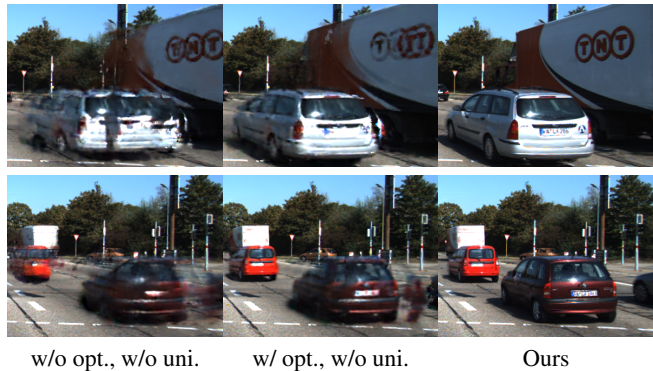


Figure 9. **Detail Qualitative Comparison** on KITTI with Noisy Bounding Boxes.

ance. However, note that incorporating the flow supervision clearly improves the underlying geometry, since optical flow provides explicit correspondence. See supplement for qualitative comparison.

5. Conclusion

In this paper, we present HUGS, a holistic scene representation that jointly optimizes appearance, geometry, and motion for urban scenes. This leads to state-of-the-art performance on various tasks. Our method has several limitations. Firstly, the reconstructed dynamic objects can only rotate to a certain degree. Future work may explore category-level prior, to enable accurate reconstruction of the full object. Further, our model lacks control of more degrees of freedom, e.g., light editing, which could be a promising direction to explore based on the Gaussian representation.

Acknowledgements: This work is supported by NSFC under grant 62202418, U21B2004 and the National Key R&D Program of China under Grant 2021ZD0114501. Yiyi Liao is with the Zhejiang Provincial Key Laboratory of Information Processing, Communication and Networking (IPCAN). Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533) and the DFG EXC number 2064/1 - project number 390727645.

References

- [1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. [2](#)
- [2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields, 2021. arXiv:2103.13415 [cs]. [2](#), [6](#), [7](#)
- [3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields, 2022. arXiv:2111.12077 [cs].
- [4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields, 2023. arXiv:2304.06706 [cs]. [2](#)
- [5] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. InverseForm: A Loss Function for Structured Boundary-Aware Segmentation, 2021. arXiv:2104.02745 [cs]. [2](#), [5](#)
- [6] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. InverseForm: A Loss Function for Structured Boundary-Aware Segmentation, 2021. arXiv:2104.02745 [cs]. [14](#)
- [7] Johann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2, 2020. arXiv:2001.10773 [cs, eess]. [5](#), [12](#)
- [8] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category Level Object Pose Estimation via Neural Analysis-by-Synthesis, 2020. arXiv:2008.08145 [cs]. [12](#)
- [9] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation, 2020. arXiv:1911.10194 [cs]. [2](#)
- [10] Ainaz Eftekhari, Alexander Sax, Roman Bachmann, Jitendra Malik, and Amir Zamir. Omnidata: A Scalable Pipeline for Making Multi-Task Mid-Level Vision Datasets from 3D Scans, 2021. [2](#)
- [11] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic NeRF: 3D-to-2D Label Transfer for Panoptic Urban Scene Segmentation, 2022. arXiv:2203.15224 [cs]. [1](#), [4](#), [12](#)
- [12] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1418–1425, San Francisco, CA, USA, 2010. IEEE. [2](#)
- [13] A. Geiger, P. Lenz, and R. 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, Providence, RI, 2012. IEEE. [5](#), [12](#)
- [14] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes’ Rays: Uncertainty Quantification for Neural Radiance Fields, 2023. arXiv:2309.03185 [cs]. [2](#)
- [15] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. StreetSurf: Extending Multi-view Implicit Surface Reconstruction to Street Views, 2023. arXiv:2306.04988 [cs]. [2](#)
- [16] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular Quasi-Dense 3D Object Tracking, 2021. arXiv:2103.07351 [cs]. [5](#), [6](#), [15](#)
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *42(4)*. [2](#), [3](#), [5](#)
- [18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. [11](#)
- [19] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation, 2022. arXiv:2205.04334 [cs]. [1](#), [2](#), [6](#), [7](#), [13](#)
- [20] Florent Lafarge, Renaud Keriven, Mathieu Bredif, and Hoang-Hiep Vu. A Hybrid Multiview Stereo Algorithm for Modeling Urban Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):5–17, 2013. [2](#)
- [21] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. [5](#), [6](#), [12](#)
- [22] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban Radiance Field Representation with Deformable Neural Mesh Primitives, 2023. arXiv:2307.10776 [cs]. [2](#)
- [23] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis, 2023. arXiv:2308.09713 [cs]. [3](#)
- [24] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections, 2021. arXiv:2008.02268 [cs]. [2](#), [4](#)
- [25] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, 2020. arXiv:2003.08934 [cs]. [2](#)
- [26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. [2](#)
- [27] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural Scene Graphs for Dynamic Scenes, 2021. arXiv:2011.10379 [cs]. [1](#), [2](#), [5](#), [6](#), [13](#), [14](#)
- [28] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. iDisc: Internal Discretization for Monocular Depth Estimation, 2023. arXiv:2304.06334 [cs]. [2](#)

- [29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. [2](#)
- [30] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban Radiance Fields, 2021. arXiv:2111.14643 [cs]. [1](#), [2](#), [4](#)
- [31] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning Multi-View Aggregation In the Wild for Large-Scale 3D Semantic Segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5565–5574, New Orleans, LA, USA, 2022. IEEE. [2](#)
- [32] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, Las Vegas, NV, USA, 2016. IEEE. [2](#)
- [33] Matthew Tancik, Vincent Casser, Xinchun Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable Large Scene Neural View Synthesis, 2022. arXiv:2202.05263 [cs]. [2](#)
- [34] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A Modular Framework for Neural Radiance Field Development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, pages 1–12, 2023. arXiv:2302.04264 [cs]. [2](#), [6](#), [7](#), [13](#), [15](#)
- [35] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical Multi-Scale Attention for Semantic Segmentation, 2020. arXiv:2005.10821 [cs]. [2](#)
- [36] Haihem Turki, Jason Y. Zhang, Francesco Ferroni, and Deva Ramanan. SUDS: Scalable Urban Dynamic Scenes, 2023. arXiv:2303.14536 [cs]. [2](#)
- [37] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi S. M. Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. NeSF: Neural Semantic Fields for Generalizable Semantic Segmentation of 3D Scenes, 2021. arXiv:2111.13260 [cs]. [2](#)
- [38] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F²-NeRF: Fast Neural Radiance Field Training with Free Camera Trajectories, 2023. arXiv:2303.15951 [cs]. [2](#)
- [39] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the Scenes: Density Fields for Single View Reconstruction, 2023. arXiv:2301.07668 [cs]. [2](#)
- [40] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. MARS: An Instance-aware, Modular and Realistic Simulator for Autonomous Driving, 2023. arXiv:2307.15058 [cs]. [1](#), [2](#), [5](#), [6](#), [7](#), [13](#), [14](#)
- [41] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofghi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying Flow, Stereo and Depth Estimation, 2023. arXiv:2211.05783 [cs]. [5](#)
- [42] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofghi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying Flow, Stereo and Depth Estimation, 2023. arXiv:2211.05783 [cs]. [2](#)
- [43] Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4K4D: Real-Time 4D View Synthesis at 4K Resolution, 2023. arXiv:2310.11448 [cs]. [3](#)
- [44] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning Object-Compositional Neural Radiance Field for Editable Scene Rendering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13759–13768, Montreal, QC, Canada, 2021. IEEE. [2](#)
- [45] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. EmerNeRF: Emergent Spatial-Temporal Scene Decomposition via Self-Supervision, 2023. arXiv:2311.02077 [cs]. [2](#)
- [46] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A Neural Closed-Loop Sensor Simulator. [1](#), [2](#)
- [47] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction, 2023. arXiv:2309.13101 [cs]. [3](#)
- [48] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting, 2023. arXiv:2310.10642 [cs]. [3](#)
- [49] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction, 2022. arXiv:2206.00665 [cs]. [2](#)
- [50] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, Salt Lake City, UT, 2018. IEEE. [6](#), [12](#)
- [51] Xiaoshuai Zhang, Abhijit Kundu, Thomas Funkhouser, Leonidas Guibas, Hao Su, and Kyle Genova. Nerflets: Local Radiance Fields for Efficient Structure-Aware 3D Scene Representation from 2D Supervision, 2023. arXiv:2303.03361 [cs]. [1](#), [2](#)
- [52] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-Place Scene Labelling and Understanding with Implicit Scene Representation, 2021. arXiv:2103.15875 [cs]. [2](#), [4](#), [12](#)
- [53] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3D Gaussian Avatars, 2023. arXiv:2311.08581 [cs]. [3](#)
- [54] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. EWA splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. [11](#)

Appendix

In this appendix, we begin by discussing implementation details in Appendix A, which includes information about our 3D Gaussian, metrics, and the training and inference processes. We then describe the datasets used in our experiments in Appendix B. Appendix C provides information about the baselines we compare with. Finally, Appendix D contains additional experiment results.

A. Implementation

In this section, we begin by discussing our 3D Gaussian details, encompassing semantic, opacity and depth implementation (Appendix A.1). Subsequently, we discuss the difference between 3D softmax and 2D softmax in 3D Semantic Scene Reconstruction (Appendix A.2). Finally, we elucidate the evaluation metrics we utilize (Appendix A.3). Our source code will be released.

A.1. 3D Gaussian Details

Following [18], each Gaussian has the following attributes: rotation ($\mathbf{R}_g \in \mathbb{R}^{3 \times 3}$), scale ($\mathbf{S}_g \in \mathbb{R}^{3 \times 1}$), opacity (α) and spherical harmonics (SH). The corresponding 3D covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$ can be calculated using the following formula:

$$\Sigma = \mathbf{R}_g \mathbf{S}_g \mathbf{S}_g^T \mathbf{R}_g^T \quad (15)$$

When provided with a viewing transformation $\mathbf{W} \in \mathbb{R}^{3 \times 3}$ and the Jacobian of the affine approximation of the projective transformation $\mathbf{J} \in \mathbb{R}^{3 \times 3}$, the covariance matrix $\Sigma' \in \mathbb{R}^{3 \times 3}$ in camera coordinates can be expressed as:

$$\Sigma' = \mathbf{J} \mathbf{W} \Sigma \mathbf{W}^T \mathbf{J}^T \quad (16)$$

Following EWA splatting [54], we can skip the third row and column of Σ' to obtain a 2×2 covariance matrix with the same structure and properties. For brevity, we still use the notation $\Sigma' \in \mathbb{R}^{2 \times 2}$ to denote the 2D covariance matrix.

By considering the projected 3D Gaussian center $\mu \in \mathbb{R}^{2 \times 1}$ and an arbitrary point $\mathbf{x} \in \mathbb{R}^{2 \times 1}$ on camera coordinates, the opacity α' of \mathbf{x} contributed by this 3D Gaussian can be computed as follows:

$$\alpha' = \alpha \exp \left(-\frac{1}{2} (\mathbf{x} - \mu)^T (\Sigma')^{-1} (\mathbf{x} - \mu) \right) \quad (17)$$

The color \mathbf{c} of each Gaussian can be computed based on the view direction and its corresponding spherical harmonics (SH). Given a set of sorted 3D Gaussians \mathcal{N} along the ray, we obtain the accumulated color via volume rendering:

$$\pi : \quad \mathbf{C} = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (18)$$

The same volume rendering technique can be applied to obtain semantic \mathbf{S} , depth \mathbf{D} and optical flow \mathbf{F} . With the given semantic feature \mathbf{s}_i , depth value d_i , and Gaussian motion \mathbf{f}_i relative to the camera pose, we can define the semantic rendering, depth rendering, and flow rendering as follows:

$$\mathbf{S} = \sum_{i \in \mathcal{N}} \text{softmax}(\mathbf{s}_i) \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (19)$$

$$\mathbf{D} = \sum_{i \in \mathcal{N}} d_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (20)$$

$$\mathbf{F} = \sum_{i \in \mathcal{N}} \mathbf{f}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (21)$$

Note that all the projections and volume rendering techniques mentioned are implemented in CUDA. Calculating the projected 2D opacity α' on each pixel and sorting Gaussians based on their distances from the camera takes the majority of computations in the rendering process. These computations need to be performed only once for rendering all modalities, thus maintaining the real-time rendering property of the original 3D Gaussian Splatting.

A.2. 3D Semantic Scene Reconstruction

We utilize Eq. (19), referred to as 3D softmax, to render semantic maps. This is in contrast to most existing NeRF-based semantic reconstruction methods that perform softmax to the accumulated 2D logits [11, 52], described in Eq. (22), referred to as 2D softmax. The fundamental difference between these two rendering techniques lies in the fact that 3D softmax normalizes the logits of each 3D point. This normalization process helps prevent a single point with a significantly high logit value from imposing an overwhelming influence on the overall volume rendering outcome. On the other hand, it also prevents placing 3D points of low logit values in empty space. As a result, 3D softmax is effective in reducing floaters and enhancing the geometry of the reconstruction results. In Appendix D.3, we present a comprehensive analysis of the qualitative and quantitative comparison results between these two rendering methods.

$$S_{2D_norm} = \text{softmax} \left(\sum_{i \in \mathcal{N}} s_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \right) \quad (22)$$

In the following sections, we refer to our default setting obtained by Eq. (19) as S_{3D_norm} .

A.3. Metrics

Novel View Appearance Synthesis: To assess the quality of novel view appearance synthesis, we utilize the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [50] following the common practice.

Novel View Semantic Synthesis: Following KITTI-360 [21], we evaluate the quality of novel view semantic synthesis via the mean Intersection over Union (mIoU) metric.

3D Semantic Reconstruction: We evaluate 3D semantic reconstruction quality by extracting a 3D semantic point cloud and comparing it with the ground truth LiDAR points. We evaluate both geometric and semantic metrics in the 3D space. Specifically, we evaluate geometric reconstruction quality by measuring the accuracy (*acc.*) and completeness (*comp.*). Accuracy measures the average distance from reconstructed points to the nearest LiDAR point, while completeness measures the average distance from LiDAR points to the nearest reconstructed points. In order to measure the semantic quality of the reconstructed point cloud, we map the predicted 3D semantics to the LiDAR points. Concretely, for each point in the LiDAR point cloud, we identify its closest counterpart in the predicted semantic point cloud and allocate a semantic label based on this nearest neighbor. The assigned semantic labels of all LiDAR points are then compared with the 3D semantic segmentation ground truth provided by KITTI-360, evaluated via the mIoU metric. Note that we only use the LiDAR point clouds for evaluation.

3D Tracking: To demonstrate the effectiveness of our model in rectifying noisy 3D tracking results, we evaluate the accuracy of predicted poses compared to ground truth poses in our ablation study. Considering the rotation and translation parameters of a ground truth bounding box denoted as $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$, respectively, and the corresponding parameters of predicted poses, represented as \mathbf{R} and \mathbf{t} , we employ two metrics for this evaluation following [8]: $e_{\mathbf{R}}$ quantifies the rotation accuracy, while $e_{\mathbf{t}}$ assesses the translation accuracy as follows

$$e_{\mathbf{R}} = \arccos \frac{Tr(\hat{\mathbf{R}} \cdot \mathbf{R}^{-1}) - 1}{2} \quad (23)$$

$$e_{\mathbf{t}} = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 \quad (24)$$

where Tr represents the trace of a matrix.

Depth Estimation: In our ablation study, we evaluate the depth estimation quality of our different variants. This is achieved by first projecting the LiDAR points acquired at the same frame to the 2D image space, followed by measuring the L2 distance between the projected LiDAR depth and our method. Considering the projected LiDAR depth is sparse, our assessment focuses solely on pixels with valid LiDAR projections when calculating the L2 distance.

B. Data

In this section, we present details of datasets on which we conducted our experiments, including KITTI [13], Virtual KITTI 2 (vKITTI) [7] and KITTI-360 [21].

	Pre.	+ π RGB	+ Affine	+ π Semantic	+ π Flow
Speed (ms)	6.25	8.13 (+1.88)	8.54 (+0.41)	9.70 (+1.16)	10.17 (+0.47)

Table 6. **Time consumption breakdown** of our method.

KITTI: Following NSG [27] and MARS [40], we select frames 140 to 224 from Scene02 and frames 65 to 120 from Scene06 on KITTI for conducting our experiments.

vKITTI: Virtual KITTI 2 is a synthetic dataset that closely resembles the scenes present in KITTI. In line with the settings outlined in NSG and MARS, we conduct experiments on exactly the same frames from Scene02 and Scene06.

KITTI-360: In addition, we perform experiments on KITTI-360, encompassing both static and dynamic scenes. For the tasks of novel view synthesis and novel semantic synthesis on the leaderboard, we conduct experiments on the sequences provided by the official dataset. Furthermore, we explore dynamic scenes, such as frames 11322 to 11381 from sequence 00, as showcased in our teaser.

C. Baselines

In this section, we discuss the baselines against which we compare our approach, including NSG [27], MARS [40], PNF [19], and Semantic Nerfacto [34].

NSG: NSG is the pioneering method that introduces the decomposition of dynamic scenes into static background and dynamic foreground components. They propose a learned scene graph representation that enables efficient rendering of novel scene arrangements and viewpoints. However, the official source code provided by NSG often encounters issues when training on KITTI Scene02. Therefore, we utilize the version implemented by the authors of MARS, which is more stable and yields slightly improved results compared to the original version.

MARS: We utilize the latest version of the code provided by the official MARS repository. This latest version incorporates bug fixes and includes additional training iterations, resulting in improved performance. In fact, the updated version achieves a notable improvement of 3 to 4 dB on PSNR compared to the numbers reported in the original paper.

PNF: Since PNF is not open-source, we directly compare our method to their submission on the KITTI-360 leaderboard regarding novel view appearance & semantic synthesis. To the best of our knowledge, PNF is the only work that considers the optimization of noisy 3D bounding boxes of dynamic objects. In our ablation study, we conduct a naïve baseline that optimizes the 3D bounding boxes of each frame independently, which can be considered as a re-implementation of PNF’s bounding box optimization in our framework.

Semantic Nerfacto: For the evaluation of 3D semantic point cloud geometry, we compare our results with Semantic Nerfacto [34] as an alternative to PNF [19]. Nerfacto [34] is an integration of several successful methods that demonstrate strong performance on real data. It incorporates camera pose refinement, per-image appearance embedding, proposal sampling, scene contraction, and hash encoding within its pipeline. Additionally, Nerfacto includes a semantic head in its framework, enabling the generation of meaningful semantic maps, as demonstrated in Appendix D.2.

D. Additional Experiment Results

D.1. Time Consumption Breakdown

Tab. 6 shows our detailed runtime breakdown as various components are incrementally enabled. Preparation (Pre.) contains operations like tile partition and Gaussian sorting. π denotes volume rendering, and affine denotes affine transform. Other components like unicycle model, dynamic decomposition, and depth rendering are excluded as they hardly consume any additional time.

D.2. Additional Comparison Experiments

Dynamic Scene with GT 3D Bounding Boxes: Despite not being our primary focus, we additionally provide a comparison with NSG and MARS using ground truth 3D trackings. In this setting, our approach demonstrates superior performance across all test scenes, see Tab. 7.

	KITTI Scene02			KITTI Scene06			vKITTI Scene02			vKITTI Scene06		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
NSG [27]	22.51	0.653	0.397	23.38	0.717	0.243	23.50	0.718	0.352	26.42	0.811	0.170
MARS [40]	22.95	0.728	0.145	27.01	0.883	0.062	29.80	0.950	0.034	32.71	0.959	0.023
Ours	25.89	0.829	0.092	28.90	0.925	0.016	30.73	0.955	0.018	33.31	0.963	0.010

Table 7. **Novel View Appearance on Dynamic Scenes** with ground truth 3D trackings.

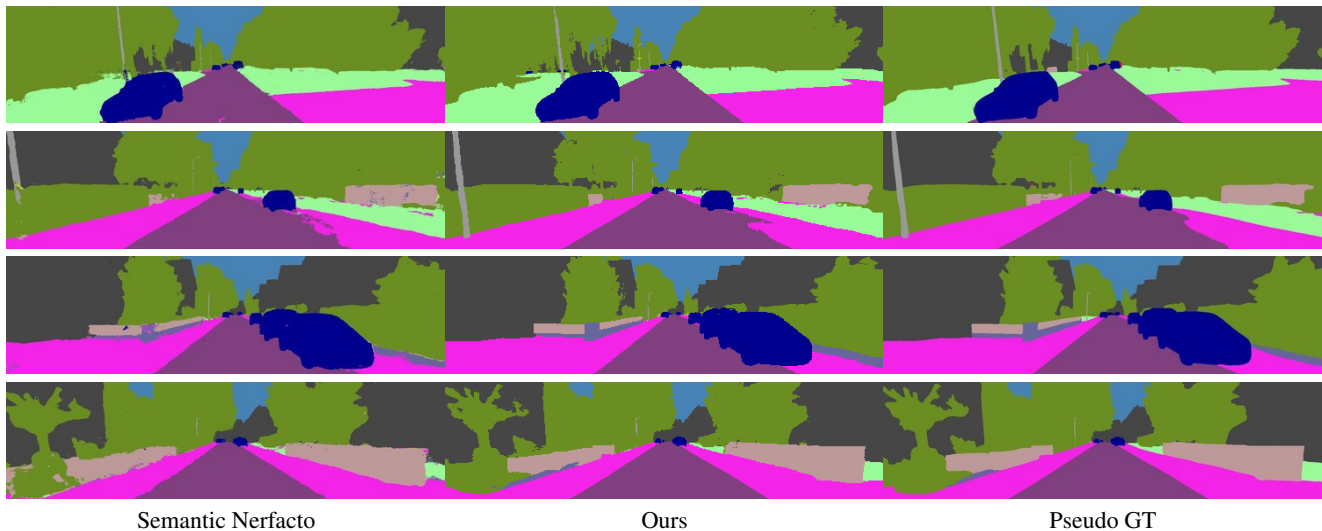


Figure 10. **Qualitative Comparison** with Nerfacto on 2D space. The Pseudo GT column represents the semantic maps that are predicted by [6] on GT RGB images.

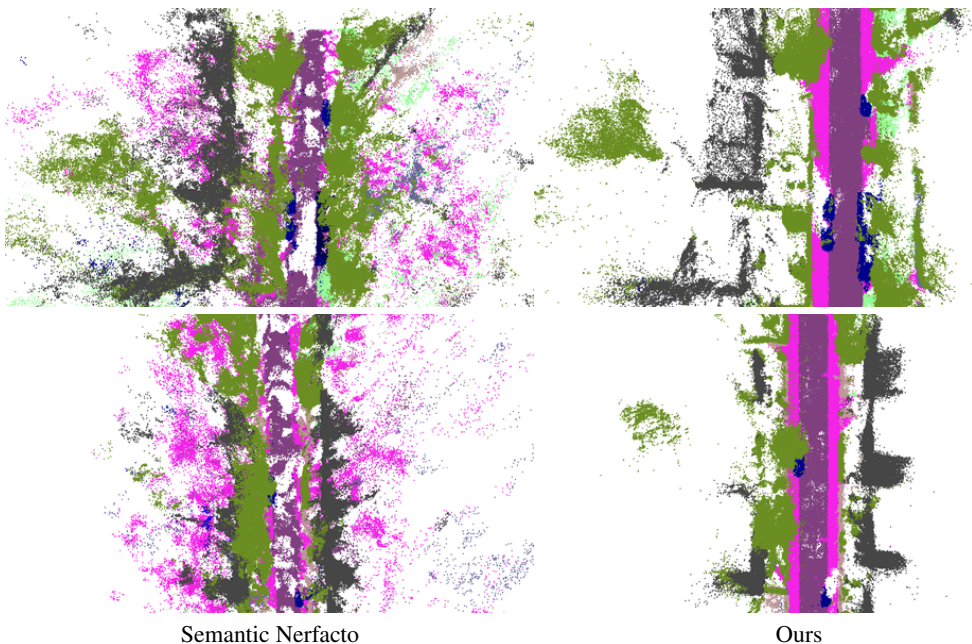


Figure 11. **Qualitative Comparison** with Nerfacto on 3D space. The semantic point cloud extracted from Semantic Nerfacto struggles to faithfully represent the geometry.

		$e_R \downarrow$	$e_t \downarrow$
KITTI 02	QD-3DT	0.027	0.215
	Ours	0.018	0.108
KITTI 06	QD-3DT	0.017	0.046
	Ours	0.012	0.033

Table 8. **Qualitative Comparison** with a tracking method, QD-3DT [16], on two sequences.

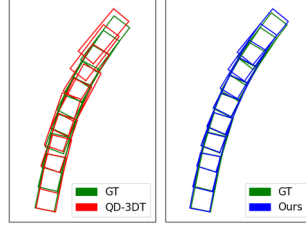


Figure 12. **Pose comparison** with QD-3DT.

	Seq01 mIoU _{cls} \uparrow	Seq02 mIoU _{cls} \uparrow	Seq03 mIoU _{cls} \uparrow	Average mIoU _{cls} \uparrow
Ours w/ $S_{2D, norm}$	0.427	0.363	0.416	0.402
Ours w/ $S_{3D, norm}$	0.544	0.452	0.520	0.505

Table 9. **Comparison on 3D and 2D Semantic Softmax** on KITTI-360.

	KITTI-360 Scene00			KITTI-360 Scene01			KITTI-360 Scene02			Average		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Random	20.84	0.784	0.150	19.40	0.705	0.171	22.55	0.800	0.136	20.93	0.763	0.457
LiDAR	25.64	0.856	0.070	22.88	0.784	0.089	24.04	0.836	0.080	24.19	0.825	0.080
COLMAP	26.23	0.863	0.069	22.94	0.794	0.096	24.38	0.843	0.077	24.52	0.833	0.081

Table 10. **Quantitative Comparison** with different initialization.

Details of Comparison with Semantic Nerfacto: While Semantic Nerfacto excels at rendering meaningful novel view semantic images (as seen in Fig. 10), Fig. 11 shows it struggling to accurately reconstruct correct geometry. Following the common practice of NeRF-based semantic reconstruction methods [34], we apply 2D softmax to Semantic Nerfacto. When we attempted to apply the 3D Softmax technique to Nerfacto, it did not yield better results compared to using 2D softmax. The results can be attributed to the incorrectness of Nerfacto’s 3D geometry. Instead of adjusting 2D logits with large-scale logits in 3D, the use of 3D softmax prevents the “cheating” approach by normalizing logits in 3D space. However, this normalization requirement necessitates sufficiently accurate geometry for satisfactory results.

Comparisons with Tracking Methods: To further compare with off-the-shelf tracking methods, we show the performance of QD-3DT [16] and our optimized pose initialized with [16] in Tab. 8 and qualitatively illustrate the poses of one vehicle in Fig. 12. Our method consistently improves [16] across two KITTI scenes.

D.3. Additional Ablation Experiments

3D and 2D Semantic Softmax: We provide more 3D and 2D semantic logits softmax comparison in Fig. 13 and Tab. 9. As can be seen, normalizing semantic logits in 3D space leads to notable qualitative and quantitative improvement compared to 2D space normalization.

Improvements on Geometry: We now qualitatively examine how the optical flow loss \mathcal{L}_F and the semantic loss \mathcal{L}_S impact the geometry, as shown in Fig. 14 and Fig. 15. Both figures reveal that incorporating either the semantic loss or the optical flow loss improves the underlying geometry. While the impact of the semantic loss on geometry may be less evident, the optical flow clearly enhances geometric accuracy. This improvement is rationalized by the fact that optical flow guides correspondences across neighboring frames. It’s important to note that when the semantic loss \mathcal{L}_S is active, the sky region of the depth maps in Fig. 14 is set to infinite.

Effects of Initialization: We conduct a thorough comparison of the results obtained through different initialization strategies. In particular, we consider random initialization and COLMAP-based initialization. To further investigate whether adopting LiDAR point cloud for initialization is helpful in urban scenes, we further consider LiDAR point clouds as initialization. We report the quantitative and qualitative comparison in Tab. 10 and Fig. 16, respectively. We observe that both LiDAR and COLMAP initialization outperform random initialization. Interestingly, the COLMAP-based initialization even shows a slight advantage over the LiDAR-based one. This could be attributed to the presence of points in the LiDAR

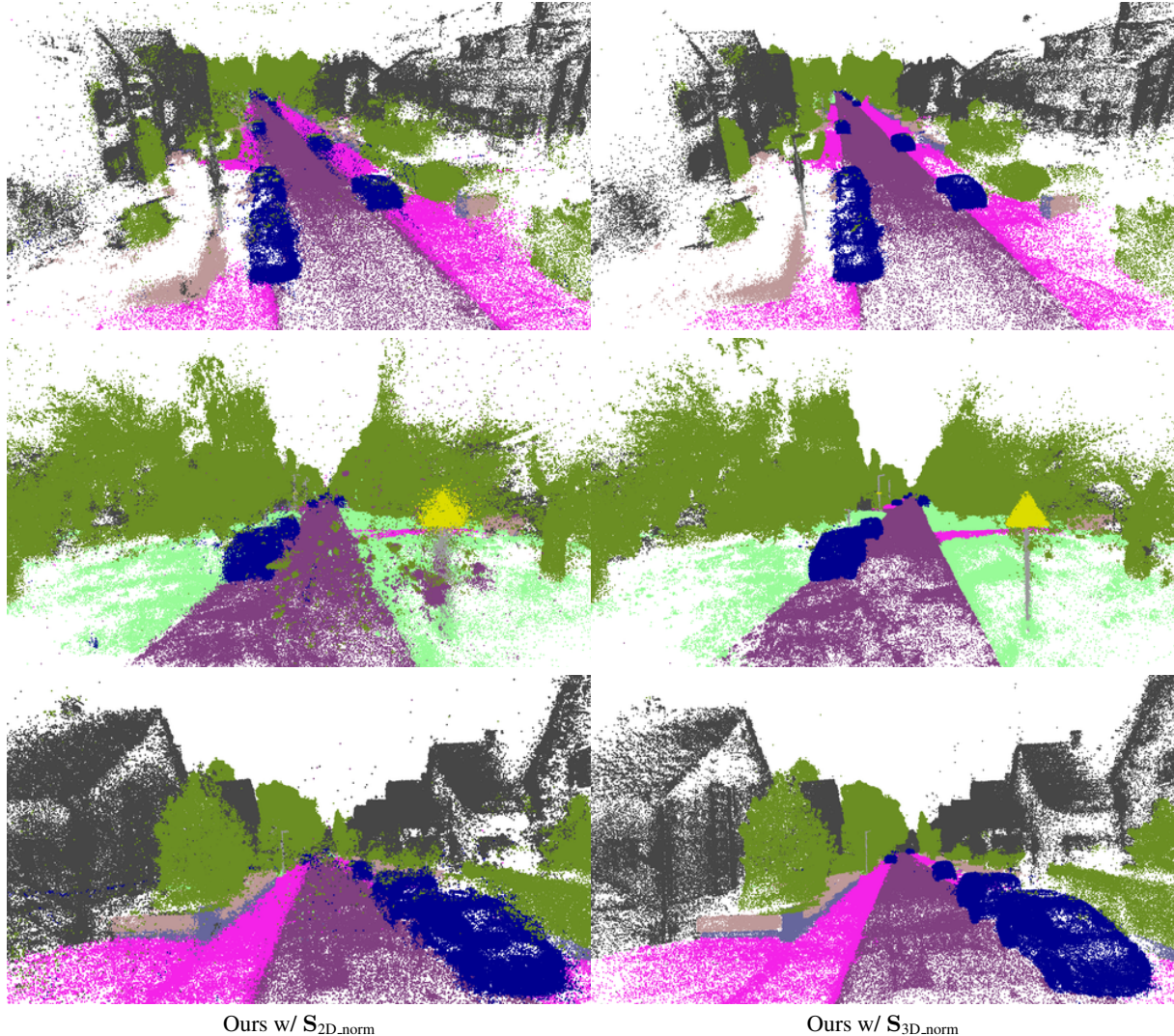


Figure 13. **Qualitative Comparison** of 3D and 2D softmax results. Note that normalizing semantic logits in 3D space (Ours w/ $S_{3D, norm}$) clearly reduces floaters and yields better 3D semantic reconstruction than the 2D normalization counterpart (Ours w/ $S_{2D, norm}$).

point clouds that remain unobserved in any training views, leading to artifacts in test viewpoints. Furthermore, COLMAP improves the quality of objects located at far distances, which cannot be accurately captured by LiDAR. These findings underscore the potential for achieving high-fidelity novel view synthesis in urban scenes based solely on RGB images. In our main experiments, we adopt the COLMAP-based initialization by default.

D.4. Visualization of Optimization Progress

We present the visualization of the optimization progress for both the noisy bounding boxes and the background semantic point cloud in Fig. 17. Using noisy 3D bounding boxes as input, our approach optimizes both the background and the poses of the bounding boxes simultaneously. As evident, the application of physical constraints derived from the unicycle model results in a smooth trajectory for the bounding boxes.

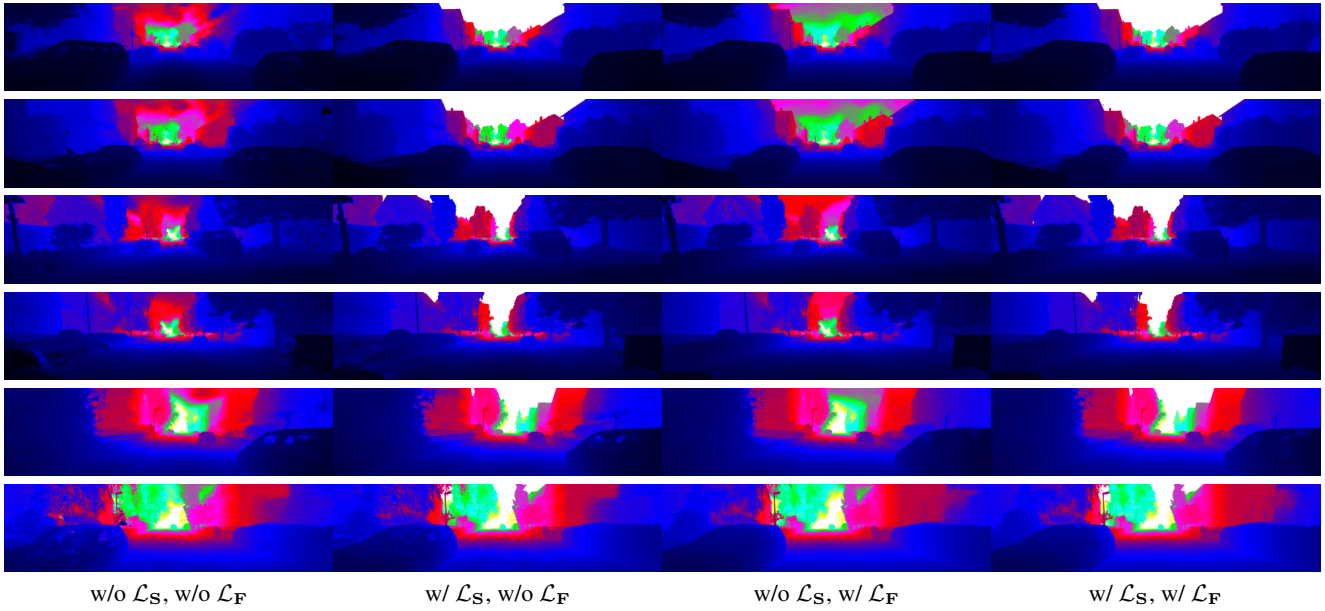


Figure 14. **Qualitative Comparison** on depth. In the presence of the semantic loss \mathcal{L}_S (2nd and 4th columns), we set the sky region’s depth infinite based on its semantic label. Note that the activation of either the semantic loss \mathcal{L}_S (2nd column) or the optical loss \mathcal{L}_F (3rd column) yields enhancements in geometry, e.g., the left car in the bottom row, with the improvement in optical flow loss being more evident.

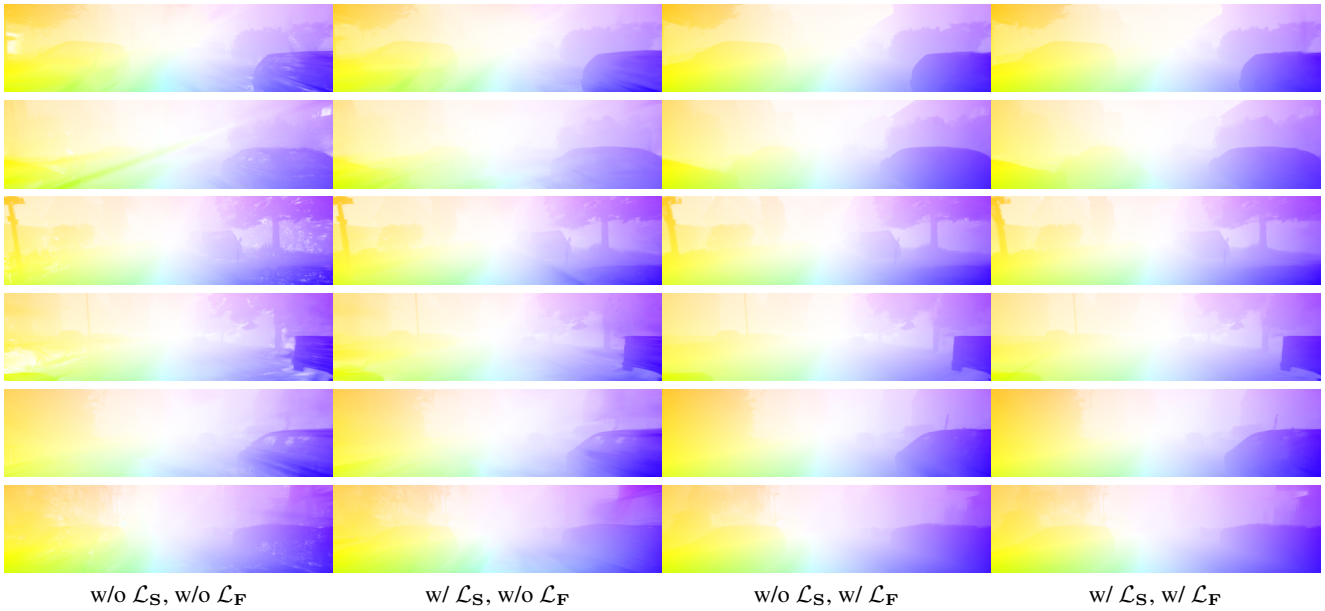


Figure 15. **Qualitative Comparison** on optical flow. While 3D Gaussians can enable the rendering of optical flow without additional supervision on semantic or optical flow, the rendered flow maps exhibit clear artifacts (1st column). These artifacts are particularly noticeable on the cars and the ground. Interestingly, the incorporation of semantic supervision \mathcal{L}_S mitigates the artifacts to some extent (2nd column). Additionally, introducing pseudo-optical flow supervision \mathcal{L}_F contributes to further improvement in the optical flow results (3rd and 4th columns).

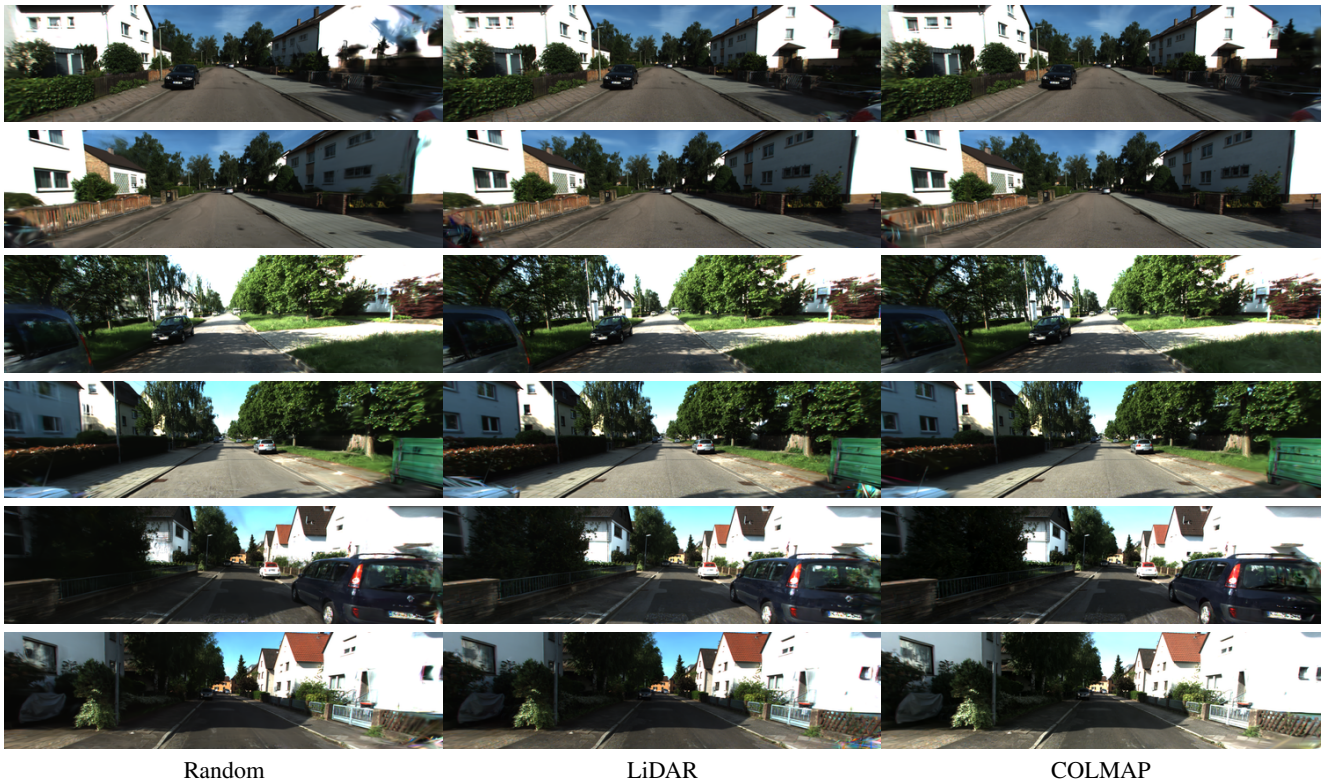


Figure 16. **Qualitative Comparison** with different initialization strategies. The superiority of both LiDAR-based and COLMAP-based initialization over random initialization is evident. Random initialization occasionally results in significant artifacts, as illustrated by the right building in the 1st row. LiDAR-based initialization, while generally effective, introduces artifacts in areas very close to the ego car, such as the bottom right corner of the 4th-6th rows. These regions typically encompass LiDAR points unseen by any training views. The COLMAP-based initialization further demonstrates an improvement over the LiDAR-based approach in distant regions, exemplified by the trees in the 1st row.

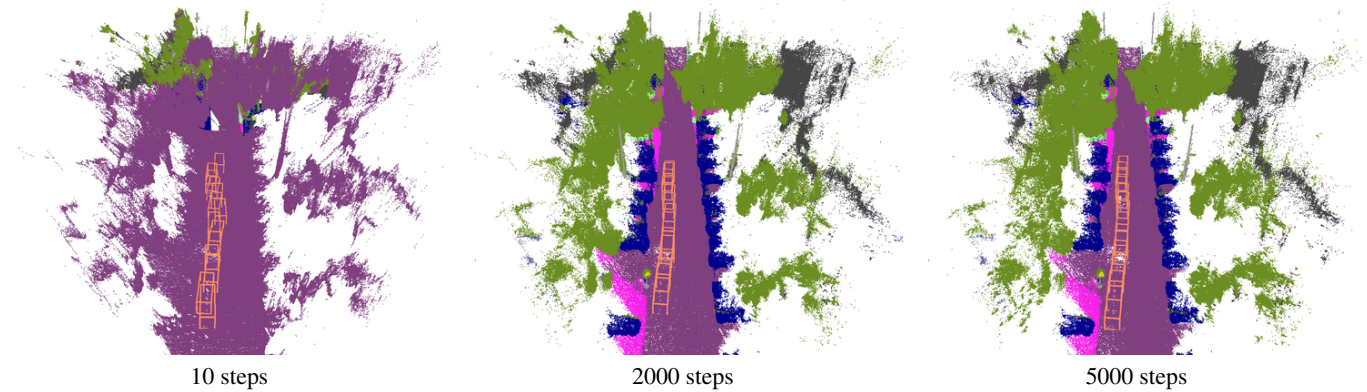


Figure 17. **Visualization of Optimization Progress.** Our method jointly optimizes the static background and the trajectory of the dynamic foreground objects. By integrating physical constraints using the unicycle model, our method allows for recovering a smooth trajectory from noisy 3D bounding boxes. To prevent visual clutter, we exclude point clouds of the dynamic object and only visualize the bounding boxes.