Leveraging 2D Priors and SDF Guidance for Dynamic Urban Scene Rendering

Siddharth Tourani[2,3] Jayaram Reddy[1] Akash Kumbar[1] Satyajit Tourani[1]
Nishant Goyal[5] Madhava Krishna[1] N Dinesh Reddy[4] Muhammad Haris Khan[2]
[1] IIIT Hyderabad, [2]MBZUAI, [3]University of Heidelberg, [4] VLM Run, [5] IIT Kharagpur
https://dynamic-ugsdf.github.io/

Abstract

Dynamic scene rendering and reconstruction play a crucial role in computer vision and augmented reality. Recent methods based on 3D Gaussian Splatting (3DGS), have enabled accurate modeling of dynamic urban scenes, but for urban scenes they require both camera and LiDAR data, ground-truth 3D segmentations and motion data in the form of tracklets or pre-defined object templates such as SMPL. In this work, we explore whether a combination of 2D object agnostic priors in the form of depth and point tracking coupled with a signed distance function (SDF) representation for dynamic objects can be used to relax some of these requirements. We present a novel approach that integrates Signed Distance Functions (SDFs) with 3D Gaussian Splatting (3DGS) to create a more robust object representation by harnessing the strengths of both methods. Our unified optimization framework enhances the geometric accuracy of 3D Gaussian splatting and improves deformation modeling within the SDF, resulting in a more adaptable and precise representation. We demonstrate that our method achieves state-of-the-art performance in rendering metrics even without LiDAR data on urban scenes. When incorporating LiDAR, our approach improved further in reconstructing and generating novel views across diverse object categories, without ground-truth 3D motion annotation. Additionally, our method enables various scene editing tasks, including scene decomposition, and scene composition.

1. Introduction

Representing and modeling large-scale dynamic scenes serves as the foundation of 3D scene understanding, playing a critical role in various autonomous driving tasks such as 3D object detection [6, 11], motion planning [69] and simulation of safety-critical scenarios [15, 81]. As long as driving simulators like CARLA [15] and MARS [90] work with synthetic data, there remains a big domain gap to real-world driving scenarios, which can lead to adverse results when deployed in self-driving cars [65].

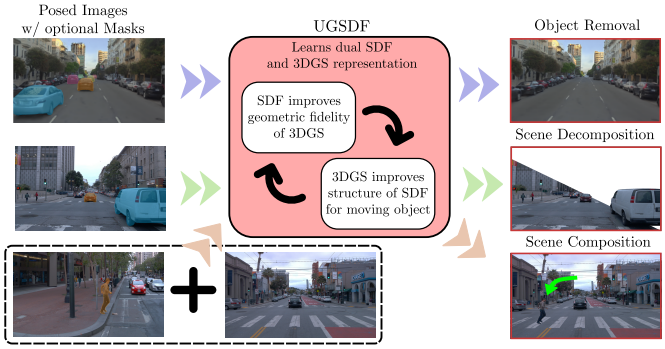

Figure 1. We propose **U**rban **G**aussians via **S**igned **D**istance **F**unctions (UGSDF) for dynamic object modeling and rendering in urban scenes. UGSDF maintains a Signed Distance Function (SDF) and 3D Gaussian Splatting (3DGS) representation to model and render a dynamic object. It can be used for object removal, scene decomposition, scene composition (*inserted object indicated with green arrow*) and other tasks related to simulation in urban scenes.

To mitigate the domain gap, real-world data integration is essential. Neural Radiance Fields (NeRFs) [50] and 3D Gaussian Splatting (3DGS) [33] excel in scene reconstruction and novel view synthesis with high visual fidelity but may compromise geometric accuracy. In contrast, Signed Distance Fields (SDFs) offer precise surface modeling but require dense representations to achieve comparable visual fidelity.

Separate from real-world verisimilitude, reconstructing dynamic driving scenes is challenging due to diverse actors, sensor noise, complex motions, occlusions, and so on. Early methods to tackle these challenges focused on static scenes [26, 49, 64, 68]. Recent methods handle dynamic scenes by either **(i)** decomposing the scene into static and dynamic components via NeRFs [68, 78, 96] and 3DGS [12, 29, 105] or **(ii)** constructing a scene graph [13, 17, 39, 55, 70, 90, 97] in which dynamic actors and static background are represented as nodes with edges encoding relative transformations that represent each actor's motion

over time. *Henceforth, we refer to any dynamic entity in a scene, including pedestrians as objects.*

Also of interest are methods for dynamic scene estimation [40, 46, 47, 69, 72, 73, 84, 85, 93, 98] that work on casually captured videos. These methods are object-agnostic and take as input *point tracking* or optical flow data to capture object motion, depth, and camera poses for 3D information to render monocular sequences for novel viewpoints. Unlike these casually captured videos, traffic datasets typically have both camera and LiDAR data, as well as camera intrinsics, extrinsics, and object motion information in the form of object tracklets and for pedestrians SMPL templates.

We introduce ***U**rban **G**aussians via **S**igned **D**istance **F**unctions* (UGSDF) (Fig. 1), a novel approach that reduces reliance on 3D object-specific motion annotations and LiDAR by leveraging 2D priors from depth networks and point trackers to extract 3D information and motion cues. In addition to standard driving dataset inputs—images, intrinsics, extrinsics, object masks and optionally LiDAR, our method also incorporates object tracks from a point tracker and depth from a depth network. This enables state-of-the-art scene reconstruction and view synthesis without requiring object tracklets, 3D bounding boxes, or SMPL templates. UGSDF models dynamic objects using a combination of 3D Gaussian primitives and a Signed Distance Function (SDF), integrating their complementary strengths for enhanced rendering and geometric accuracy. The 3D Gaussians facilitate motion modeling and high-fidelity rendering, generating depth maps that refine SDF ray sampling. In turn, the SDF smooths surfaces and aids in Gaussian placement, iteratively improving the scene's geometry. This dual representation effectively captures dynamic objects while remaining adaptable to static scenes by simply omitting motion modeling when unnecessary.

Our **contributions** are:

- **2D Prior Based Dynamic Object Modeling In Urban Scenes** Our method derives motion and 3D structure information of dynamic objects from off-the-shelf point-trackers [32] and metric depth networks [58] by learning a dual SDF and 3DGS representation. It does so without using 3D tracklet like motion information.
- **SDF based improvement of Gaussian primitive distribution** We train an SDF deformation network to model the geometry of a dynamic object, which we in-turn query to improve the localization of Gaussians on the dynamic object. Likewise, we use the location of the Gaussians to focus the ray-sampling of the SDF network.
- **State-of-the-art results on Reconstruction and Novel View Synthesis of Dynamic objects** The learned dual representation excels in scene reconstruction and novel view synthesis, surpassing template-free methods on KITTI [21] and Waymo [67], and even outperforming template-based methods in some cases. We further demonstrate its versatility on casually captured datasets.

2. Related Works

Dynamic Modeling of Urban Scenes Neural representations [2, 3, 18, 33, 50, 52] have emerged as a dominant force in novel view synthesis, and have since been extended to dynamic scenes. *Deformation* based methods like [5, 30, 42, 43, 47, 56, 57, 59, 76, 88, 89, 91, 98, 106], warp time-varying observations to a canonical space via a deformation network or input image timestamps (or latent codes) into neural representations. These techniques are typically limited to small scenes, making them less effective for dynamic urban environments.

Dynamic Decomposition based methods like [29, 78, 96] have demonstrated reconstruction abilities for dynamic driving scenes, but are limited in control due to their using a single dynamic field for all scene objects.

Scene Graph-based methods, such as [13, 17, 55, 70, 90, 94, 97], model dynamic objects using separate neural representations within a scene graph. These approaches require ground-truth motion data, along with 3D bounding box tracklets and 2D masks. Additionally, OmniRe [13] incorporates SMPL templates for modeling pedestrians. Our method shows using 2D priors in the form of point trackers and depth networks when combined with our dual representation of SDF networks and 3D Gaussians can also yield high-fidelity novel-view synthesis without requiring any 3D annotations, even without LiDAR data.

Neural Surface Reconstruction Meets 3DGS Neural rendering has advanced neural surface reconstruction [44, 53, 83, 86, 99, 100], using neural networks to represent scene geometry through occupancy fields or SDF values. Recent methods [44, 63, 87] leverage hashed feature grids [52] for enhanced representation power, achieving excellent results. Hybrid techniques combining surface and volume rendering [63, 79, 87] improve both speed and quality. Approaches like [25, 92, 102] align 3D Gaussians with surfaces, while [28] enhances ray-splat intersections. NeuSG [10] and 3DGSR [48] use 3D Gaussian Splatting with SDF fields for static scenes. To our knowledge, we are the first to combine SDFs and 3DGS for dynamic urban scenes for modeling individual dynamic objects.

SAM Meets 3DGS Semantic-NeRF [103] pioneered integrating semantic information into NeRF, enabling 3D segmentation from noisy 2D labels. Later work [4, 19, 39, 66] introduced object-aware representations with instance modeling, relying on ground-truth labels. For open-world segmentation, approaches [22, 35, 38, 77] distilled 2D features from foundation models [7, 41, 61] into radiance fields [60, 104], but struggled with similar objects. SAM's open-world segmentation [8, 9, 27, 31, 36, 37, 95, 101] was adopted for applying 3D segmentation to static and dynamic

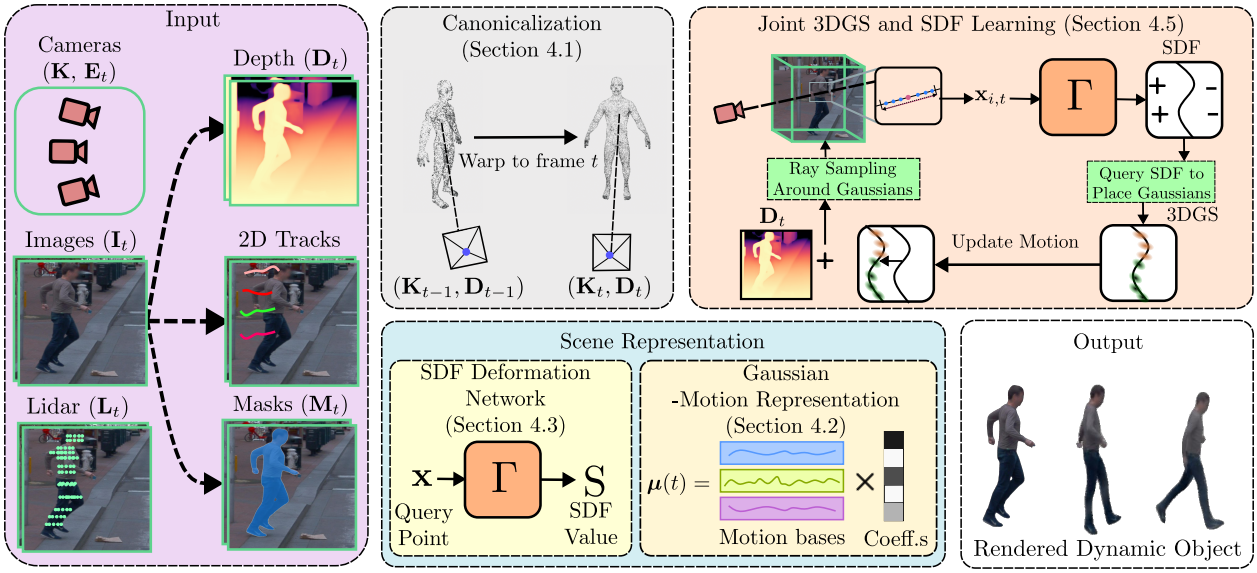

Figure 2. **Overview** UGSDF takes posed images, dynamic object masks, 2D tracking data, and depth maps (optionally LiDAR) as input, and outputs rendered dynamic scene. The initial model is constructed through canonicalization described in Section 4.1. A dynamic 3DGS motion representation described in Section 4.2. The object of interest is represented using a combined representation of Signed Distance Functions (SDFs) and 3D Gaussian Splatting (3DGS) as detailed in Section 4.3 and the joint optimization scheme is described in Section 4.5. These two representations are jointly learned. The coarse geometry of the Gaussians constrains the ray sampling of the SDF, while SDF queries add detail to the Gaussian representation.

but indoor scenes. In contrast, to the best of out knowledge, ours is the first method to use per-object SAM masks to learn 3D consistent segmentations in urban scenes without additional 3D annotations.

3. Preliminaries

3D Gaussian Splatting 3DGS [33] represents the 3D scene as differentiable 3D Gaussian primitives. Each Gaussian primitive is defined by the tuple $(\boldsymbol{\mu}, \mathbf{R}, \mathbf{s}, \mathbf{o}, \mathbf{r})$ where $\boldsymbol{\mu} \in \mathbb{R}^3$ is its mean, $\boldsymbol{R}$ and $\boldsymbol{S}$ are the orientation and scale of the Gaussians. $\mathbf{o} \in \mathbb{R}$ and $\mathbf{r} \in \mathbb{R}^3$ are the opacity and color (represented as the *RGB* value) of the Gaussians. We denote the rendering operation $\hat{\mathbf{I}} = \mathcal{R}(\mathcal{G}, \mathbf{K}, \mathbf{E})$ where $\mathcal{R}$ is the rendering function, $\mathbf{K}$, $\mathbf{E}$ are the camera intrinsics and extrinsics. $\mathcal{G}$ is the set of Gaussians and the output $\hat{\mathbf{I}}$ is the rendered image. We can rasterize normal maps, depth maps and mask images by modifying the rendering equation as in [14, 80].

Neural Implicit SDFs Signed Distance Functions (SDFs) offer an effective way for representing surfaces implicitly as a zero-level set, $\{\boldsymbol{x} \in \mathbb{R}^3 \mid \Gamma(\boldsymbol{x}) = 0\}$, where $\Gamma(\boldsymbol{x})$ is the SDF value from a neural network $\Gamma(\cdot)$. Following NeuS [83], we replace the volume density with SDF and convert the SDF value to the opacity α_i with a logistic function:

$$\alpha_i = \max\left(\frac{\phi_s(\Gamma(\boldsymbol{x}_i)) - \phi_s(\Gamma(\boldsymbol{x}_{i+1}))}{\phi_s(\Gamma(\boldsymbol{x}_i))}, 0\right), \tag{1}$$

where ϕ_s denotes a Sigmoid function. Using the volume rendering methodology in [83], the predicted color of pixel $\boldsymbol{p}$ can be used to render images. Likewise, replacing $\mathbf{c}_i$ with depth $\mathbf{d}_i$ and normals $\mathbf{n}_i$,

4. Method

Our method *Urban Gaussians via Signed Distance Functions* (UGSDF) takes as input a sequence of RGB frames captured by different cameras, represented as $\{\mathbf{I}_t^c \mid t = 1, \ldots, N\}$. Each frame includes a mask of the object of interest, $\{\mathbf{M}_t^c\}$, along with the camera intrinsics $\mathbf{K}^c \in \mathbb{R}^{3\times3}$ and world-to-camera extrinsics $\mathbf{E}_t^c \in \mathbb{SE}(3)$ for each frame $\mathbf{I}_t^c$ and camera c. When available, LiDAR scans $\mathbf{L}_t$ for each frame are also used as input. The goal of our method is to generate a coherent 4D representation and renderings of the object that maintain multi-view consistency, even in this sparse setup with limited viewpoints around the object. We assume the object of interest (denoted $\mathcal{O}$) appears in every frame, and we disregard any frames where $\mathcal{O}$ is absent. Figure 2 gives an overview of our method.

Mask Generation To generate dynamic masks $\{\mathbf{M}_t^c\}$, we combine point prompts in SAM2 [62] with tracking from CoTracker [32]. The initial object mask is taken from the ground-truth annotations. A dense grid of points is tracked across the sequence using CoTracker, and the tracked points at each frame serve as prompts to SAM2, producing per-frame segmentations for each dynamic object. Finally, we post-process the points by pruning all point trajectories that lie outside the masks generated by SAM2.

4.1. Construction of Canonical Model

Building the Initial Scaffold We begin by constructing the initial object scaffold. We first compute the depth maps $\mathbf{D}_t^c$ of the images $\mathbf{I}_t^c$ using the pre-trained metric depth network UniDepth [58]. We use this network, as it handles the scale disparity for the depth maps between different frames. We then compute the object point clouds $\mathbf{P}_t$ for each of the different time-steps by lifting pixels corresponding to the $\mathcal{O}$ to 3D by backprojection, *i.e.* if $\boldsymbol{p}_i \in \mathbb{R}^2$ is a pixel in $\mathbf{M}_t^c$, then $\mathbf{x}_i = \mathbf{D}_t^c(\boldsymbol{p}) \times (\mathbf{K}^c)^{-1}\tilde{\boldsymbol{p}}_i$, where $\tilde{\boldsymbol{p}}_i$ is $\boldsymbol{p}$ in homogeneous coordinates. We then warp the object point clouds to a single coordinate frame referred to as the canonical frame (typically the first frame where the whole object appears) by computing the 2D pixel tracks between the current frame and the canonical frame using [32]. This is done to warp point clouds across timesteps and also cameras within the same timestep for a multi-camera setting. We can likewise use the 2D tracks to warp the lidar point clouds corresponding to the $\mathcal{O}$ when available. We denote the object point cloud in the canonical frame obtained by this process as $\mathbf{P}_o$. Without loss of generality we drop the camera index c for less cluttered notation. Since this initial scaffold is based on noisy and partial observations of the object of interest, refinement is needed to prevent error propagation to subsequent frames. Thus, the scaffold initializes the Gaussian representation, which is then used to perform an initial training for the SDF network. Warping is done in a window of 5 time-steps from the initial detection of the object to the inital detection time-step.

Gaussian Initialization We then initialize the 3D Gaussians of $\mathcal{O}$ in the canonical frame, where each 3D Gaussian $\boldsymbol{g}_o = (\mu_o, \mathbf{R}_o, \boldsymbol{s}, \boldsymbol{o}_o, \boldsymbol{r}_o)$ denotes the Gaussians, where $\mu_o \in \mathbb{R}^3$ are the means and are initialized with the points in $\mathbf{P}_o$. $\mathbf{R}_o$ are the orientation, $\boldsymbol{s}$ the scale, $\boldsymbol{o}_o$ the opacity, and $\boldsymbol{r}_o$ the color of the Gaussians. We denote $\mathcal{G}_o$ as the set of initialized Gaussians, *i.e.* $\mathcal{G}_o = \{\boldsymbol{g}_o\}$. We denote the rendered image for extrinsic $\mathbf{E}_t$ and intrinsic $\mathbf{K}$ as $\tilde{I}_o = \mathcal{R}(\mathcal{G}_o, \mathbf{E}_t, \mathbf{K}^c)$, where $\mathcal{R}$ is the differentiable rasterizer [34]. The parameters of $\mathcal{G}_o$ are then learned by minimizing

$$\mathcal{L}_{gs} = \mathcal{L}_{L1}(\mathbf{I}_o(\mathbf{M}_o), \mathcal{R}(\mathcal{G}_o)) + \mathcal{L}_{ssim}(\mathbf{I}_t(\mathbf{M}_t), \mathcal{R}(\mathcal{G}_o)) \quad (2)$$

where $\mathcal{R}_o(\mathcal{G}_o)$ denotes the rendered image and $\mathbf{I}_o(\mathbf{M}_o)$ is the part of the image corresponding to the $\mathcal{O}$.

4.2. Gaussian Motion Representation

Since it is challenging to independently model the motion of each Gaussian $\boldsymbol{g}_o$, we adopt a common assumption (used in [1, 45, 84, 98]) that models the motion trajectory as a linear combination of learnable basis trajectories.

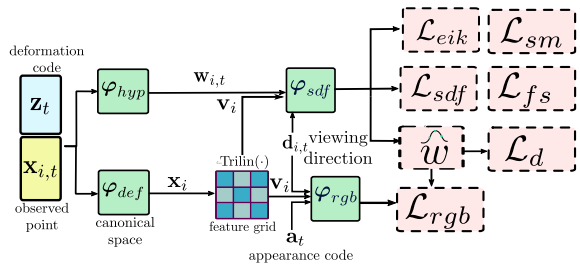

Figure 3. **SDF Deformation Network** The network takes as input the observed point $\mathbf{x}_{i,t}$ and outputs an SDF value $S_i(t)$ via the φ_{sdf} MLP.

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu}_o + \sum_{j=1}^{B} \mathbf{c}_j(t)\mathbf{b}_j^{\mu}(t) \quad (3)$$

$$\boldsymbol{q}(t) = \mathbf{q}_o + \sum_{j=1}^{B} \mathbf{c}_j(t)\mathbf{b}_j^{R}(t) \quad (4)$$

Here B the number of basis trajectories is a hyperparameter, $\mathbf{b}_j^{\mu}(t)$ and $\mathbf{b}_j^{R}(t)$ are learnable basis trajectories for the means $\mu_o \in \mathbb{R}^3$ and quaternions $\boldsymbol{q}_o \in \mathbb{R}^4$ respectively. The motion coefficients $\{\mathbf{c}_j(t) \in \mathbb{R}\}_{j=1}^{B}$ are shared across means and rotations. Both quaternion $\mathbf{q}_o$ and $\mathbf{R}_o$ represent the same rotation for each of the Gaussians. We additionally impose a sparsity penalty on the motion coefficients, forcing the Gaussian motion to be modeled by only a few basis trajectories to learn generalizable trajectories.

4.3. SDF Deformation Network

We model the SDF function for $\mathcal{O}$ as a combination of a multi-resolution feature grid [52] and shallow MLPs. The architecture of the network is given in Figure 3 and is a modified version of the architecture used in [51]. The feature grid consists of voxel grids of increasing resolution, each containing learnable features. To evaluate, each grid is queried via trilinear interpolation and the resulting features from all grids are concatenated. The network takes as input a learnable deformation code $\mathbf{z}_t$ and input point $\mathbf{x}_{i,t} \in \mathbb{R}^3$. These are both input into two different MLP networks: (1) *the deformation network* $\boldsymbol{\varphi}_{def}$ that outputs the 3D point corresponding to $\mathbf{x}_{i,t}$ in canonical space. *i.e.* $\mathbf{x}_i = \boldsymbol{\varphi}_{def}(\mathbf{x}_{i,t}, \mathbf{z}_t)$. We drop the t subscript for the canonical frame. (2) *the topology-aware network* $\boldsymbol{\varphi}_{hyp}$ which outputs a higher dimensional mapping $\mathbf{w}_{i,t} \in \mathbb{R}^m$ where m is the higher-dimensional space, *i.e.* $\mathbf{w}_{i,t} = \boldsymbol{\varphi}_{hyp}(\mathbf{x}_{i,t}, \mathbf{z}_t)$ These topology aware networks [57] are designed to handle varying topologies by mapping the canonical shape to a higher dimension. $\mathbf{x}_i$ is then input into the multi-resolution feature grid $\mathcal{V}$ obtaining the tri-linearly interpolated features $V^l(\mathbf{x}_i)$, at different resolutions (l is over different resolutions). We denote the concatenation of the multi-resolution

features $\{V^l(\mathbf{x}_i)\}$ as $\mathbf{v}_i$. The SDF output is obtained as $S_i = \boldsymbol{\varphi}_{sdf}(\mathbf{v}_i, \mathbf{w}_{i,t})$ where $\boldsymbol{\varphi}_{sdf}$ is an MLP. Likewise color is obtained as $\mathbf{c}_i^t = \boldsymbol{\varphi}_{rgb}(\mathbf{v}_{i,o}, \mathbf{w}_{i,t}, \mathbf{d}_{i,t}, \mathbf{a}_t)$. The color MLP $\boldsymbol{\varphi}_{rgb}$ takes as input in addition to $\mathbf{v}_i$ and $\mathbf{w}_{i,t}$ the viewing direction $\mathbf{d}_{i,t} \in \mathbb{R}^3$ and appearance code $\mathbf{a}_t$. We abstract away the details of the network and just denote the network as Γ, which can be thought to take input $\mathbf{x}_{i,t}$ and output a corresponding SDF Value $S_{\mathbf{i}}(t)$.

4.4. Optimization

SDF Loss The SDF network is trained by minimizing:

$$\mathcal{L}_{tot} = \mathcal{L}_{rgb} + \mathcal{L}_d + \mathcal{L}_{sdf} + \mathcal{L}_{fs} + \mathcal{L}_{eik} + \mathcal{L}_{sm} \quad (5)$$

where $\mathcal{L}_{rgb}$ and $\mathcal{L}_d$ are the The RGB and depth per-pixel rendering losses, $\mathcal{L}_{fs}$ is the free-space loss [54, 82], $\mathcal{L}_{eik}$ and $\mathcal{L}_{sm}$ are the eikonal regularization [24, 54, 83] and smoothness loss. $\mathcal{L}_{eik}$ and $\mathcal{L}_s$ both regularize the surface to be smooth in the absence of point information. $\mathcal{L}_{fs}$ enforces the network to predict large SDF values between the camera origin and the observed surface.

3DGS Loss We supervise the dynamic Gaussians with two sets of losses. The first set of losses minimize discrepancy between the per-frame pixelwise color, depth, and masks inputs. During each training step, we render the image $\hat{\mathbf{I}}_t$, depth $\hat{\mathbf{D}}_t$, and mask $\mathbf{M}_t$ from their training cameras $(\mathbf{K}, \mathbf{E}_t)$ via the differentiable rasterizer $\mathcal{R}$. These predication are supervised as:

$$\mathcal{L}_{L2}(\hat{\mathbf{I}}_t, \mathbf{I}_t) + \mathcal{L}_{L2}(\hat{\mathbf{D}}_t, \mathbf{D}_t) + \mathcal{L}_{L2}(\hat{\mathbf{M}}_t, \mathbf{M}_t) \quad (6)$$

where $\mathbf{M}_t$, $\mathbf{D}_t$ and $\mathbf{I}_t$. $\mathbf{I}_t$ is the original image, $\mathbf{D}_t$ is generated from UniDepth and $\mathbf{M}_t$ from SAM2 [62]. The second set of losses impose the standard constraints on motion using 2D tracks and depth as well as rigidity taken from [84]. The training is split into two stages, the initialization which trains the parameters of the Gaussians as well as the SDF network. We alternate between training the Gaussians and SDF network. We provide architecture and implementation details in the ***supp***.

4.5. Joint 3DGS and SDF Learning

Inaccurate underlying geometry on dynamic objects cause Gaussians to be rendered as floaters around the moving object. The original 3DGS [34] restricts its densification strategy solely to local operations like cloning or splitting, thereby posing challenges when generating new Gaussian primitives during densification in areas lacking Gaussians already. Methods like [14, 25, 92] steer the Gaussians to be close to planar surfaces, which work well for indoor scenes, as we are interested in handling non-planar moving objects we instead use the depth map to densify Gaussians.

SDF Guidance for Densification Let $\mathcal{G}_t$ denote the set of Gaussians at time t evolved via Eq. (3) and Eq. (4). We render $\hat{\mathbf{I}}_t = \mathcal{R}(\mathcal{G}_t, \mathbf{K}, \mathbf{E}_t)$ and from it generate depth map $\hat{\mathbf{D}}_t$

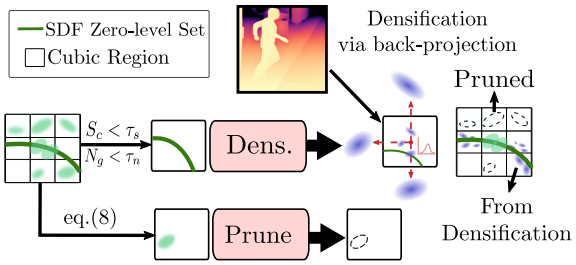

Figure 4. **SDF Guidance for Gaussian Primitive Distribution** Densification and Pruning of Gaussians is done by querying the SDF network. Points far away from the estimated SDF zero-level set are pruned, while sparse regions close to the zero-level set are chosen for densification. Dens. denotes densification.

via UniDepth. We partition the region around the object into N^3 cubic grids. The dimensions of the region around the object are based on the span of the object model after canonicalization (denoted $\mathbf{S}_c$). Then, we query the SDF value at the center of each grid. SDF values below the threshold τ_s indicates the grid is in proximity to the scene surface. In case, the Gaussians within the grid are below a certain number $N_g < \tau_n$ we back-project the depth map $\mathbf{D}_t$ creating a point cloud and sample 3 points closest to $\mathbf{c}$ to densify the Gaussians.

Subsequently, we enumerate the Gaussians from $\mathcal{G}_t$ within each grid. In cases where the number of Gaussian primitives were insufficient $N_g < \tau_n$, we select the K Gaussian neighbors of the grid's center point and generate K new Gaussians within the grid. The initial attributes of these newly generated Gaussian primitives are sampled from a normal distribution defined by the mean and variance of the K neighboring Gaussians.

SDF Guided Pruning For pruning, we integrate SDF information over a series of time-steps. If $\mathbf{x}_{i,t}$ is the position of a Gaussian at timestep t, we prune if

$$\sum_{t \in \text{prev timesteps}} \exp\left(\frac{-S_i(t) + \sum_{j \in \text{NN}(i)} S_j(t)}{\gamma}\right) < \tau_{pr} \quad (7)$$

Here, $S_i(t)$ is the SDF value corresponding to $\mathbf{x}$ at frame t and γ is a hyper-parameter to prevent the exponent from reaching zero. Similarly, $S_j(t)$ is the SDF value corresponding to a nearest neighbor of $\mathbf{x}_{i,t}$ at frame t and γ is a hyper-parameter to prevent the exponent from reaching zero. t is taken over a small window of time. j is taken over K nearest neighbors. τ_{pr} is a threshold hyper-parameter to determine when to prune. [92, 102] both propose pruning strategies that incorporate SDF and opacity information. In contrast, we don't use opacity as we found SDF values of a points neighbors to be a better indicator for pruning.

Gaussian-guided Point Sampling for SDF Accurate surface reconstruction is achieved by sampling as close to

Method	Input	Scene Reconstruction			Novel View Synthesis		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
StreetGS [94]	*M,T*	29.11	0.921	0.117	25.71	0.764	0.218
S3Gaussians [29]	-	31.35	0.911	0.106	26.82	0.788	0.226
4DGF [16]	*M,T*	29.08	0.929	0.110	28.31	0.859	0.206
OmniRe [13]	*T,M,S*	33.79	0.942	0.105	29.35	0.780	0.186
UGSDF (*Ours*)	*M,PT*	33.98	0.944	0.104	30.63	0.871	0.129
UGSDF w/o LiDAR(*Ours*)	*M,PT*	33.88	0.942	0.105	30.32	0.871	0.145

Table 1. Evaluation on the Waymo Open Dataset. We evaluate each method in terms of PSNR, SSIM, and LPIPS for full-image quality. The **Input** column covers the different inputs used by the compared methods. *T* denotes *3D tracklet*, *S* denotes *SMPL*, *M* denotes *Masks*, *PT* denotes point tracker. The *best* and *second* best results are highlighted. *T& S* are both 3D priors while *M* and *PT* are 2D priors.

Method	Scene Reconstruction			Novel View Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
StreetGS [94]	26.73	0.883	0.162	25.61	0.803	0.211
4DGF [16]	27.16	0.885	0.149	26.47	0.761	0.237
OmniRe [13]	27.95	0.895	0.147	26.59	0.816	0.191
UGSDF (*Ours*)	29.55	0.934	0.105	28.63	0.926	0.123

Table 2. Performance comparison on the KITTI Dataset. We evaluate each method in terms of PSNR, SSIM, and LPIPS for full-image quality. The *best* and *second* best results are highlighted.

the surface as possible. We employ the rasterized image from the 3D Gaussians to guide the point-sampling strategy. [92, 102] propose using the rasterized depth maps of the Gaussians as a coarse geometry for point sampling. However, we found a better way was to input the rasterized image $\hat{\mathbf{I}}_t$ into UniDepth and generate its aligned depth map $\hat{\mathbf{D}}_t$ and use it for guidance. Specifically, we leverage $\hat{\mathbf{D}}_t$ from the 3D Gaussians to narrow down the ray sampling range. Let $\mathbf{o}$ be the camera center and $\mathbf{d}$ be the viewing direction, we define the sampling range $\mathbf{r}$ of the SDF as:

$$\mathbf{r} = [\mathbf{o} + (\hat{\mathbf{D}}_t(\mathbf{p}) - \gamma|S|).\mathbf{d}, \mathbf{o} + (\hat{\mathbf{D}}_t(\mathbf{p}) + \gamma|S|).\mathbf{d}] \quad (8)$$

Here γ is a hyper-parameter and S is the SDF value of the Gaussian that intersects with the ray.

5. Experiments

Baseline methods We compare our method to four recent approaches for the tasks of full image reconstruction and novel-view synthesis: OmniRe [13], 4DGF [16], StreetGS [94], and S3Gaussians [29]. Additionally, for evaluating metrics on humans and vehicles, we compare our method against Periodic Vibration Gaussians (PVG) [12], DeformGS [98], and EmerNerf [96]. Since our method specifically focuses on modeling dynamic objects, we handle the background and sky in a manner similar to OmniRe [13]; specifically, we *minimize the same sky and background loss*. Additionally, we use SAM2 [62] to generate masks for dynamic objects captured by the cameras, assigning a unique object ID to label the same object across different frames and camera views. Our method only uses tracks from [32] and depth from UniDepth [58].

Figure 5. **Ablation Analysis** Our ablation analysis confirms the need for dense representations of thin objects like pedestrians and cyclists (rows 2 and 3). Furthermore, removing SDF guidance for Gaussian primitive distribution adversely impacts rendering quality (rows 1 and 3).

Datasets and Metrics We evaluate our approach on two benchmarks: the Nerf-On-The Road (NOTR) dataset [96] (a subset of the Waymo Open Dataset [67]) and the KITTI MOT dataset [21]. For NOTR, we use the dynamic-32 split of [96]. It consists of highly complex dynamic scenes that include typical vehicles, pedestrians, cyclists and even a tram. For KITTI, we evaluate on the MOT 21 sequences of the city category as both StreetGS and OmniRe both report numbers on sequences with 3D tracklets. For all methods, we test on every 8th frame and train on the rest for both datasets. We use the standard metrics of LPIPS, PSNR and SSIM to evaluate rendering quality. We show additional results on casually captured scenes from the iPhone dataset [20] in the **supp**.

5.1. Comparisons with State-of-the-art

Tab. 1 and Tab. 2 shows the comparison of our approach on NOTR and KITTI respectively. Remarkably, our method outperforms methods like OmniRe and 4DGF by a small margin even though both use object pose information for ve-

Methods	Scene Reconstruction				Novel View Synthesis			
	Human		Vehicle		Human		Vehicle	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
EmerNeRF[96]	22.88	0.578	24.65	0.723	20.32	0.454	22.07	0.609
DeformGS[98]	17.80	0.460	19.53	0.570	17.30	0.426	18.91	0.530
PVG[12]	24.06	0.703	25.02	0.787	21.30	0.576	22.28	0.679
StreetGS[94]	16.83	0.420	27.73	0.880	16.55	0.393	26.71	0.846
OmniRe [13]	28.15	0.845	28.91	0.892	24.36	0.727	27.57	0.858
UGSDF (Ours)	27.89	0.839	30.34	0.906	25.48	0.758	28.68	0.872
UGSDF w/o LiDAR (Ours)	27.89	0.825	29.91	0.902	25.14	0.742	28.23	0.847

Table 3. **Comparison of methods for Scene Reconstruction and Novel View Synthesis on Human and Vehicle categories** The *best* and *second* best results are highlighted.

Methods	Human		Vehicle	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Ours (Full)	27.89	0.839	30.34	0.916
w/o *SG4GP*	22.47	0.541	22.27	0.612
w/ *Spars.*	24.82	0.715	25.14	0.794
w/o *GRS4S*	25.82	0.762	27.83	0.863

Table 4. **Ablation Analysis** on the NOTR dataset.

hicles in the form of bounding boxes. This holds true even if we don't make use of any LiDAR data in the pipeline for most metrics. This speaks to the robustness of 2D priors for view synthesis. OmniRe in-fact also uses SMPL templates and body pose predictors [23]. Our method solely relies on depth from [58] and tracking information from [32]. Table 3 shows the results of our rendering on vehicles and humans. It shows that while OmniRe does do better on humans, it is not by a large margin. Secondly, our method is much better on vehicles. This explains the improvement our method has over OmniRe on the NOTR dataset as vehicles outnumber the pedestrians by in terms of pixels occupied.

Figure 6 shows qualitative results on the Waymo dataset. The results focus on non-rigid / atypical objects as all methods (baseline and ours) show reasonable results on most rigid objects. The first and third columns zoom in on cyclists. OmniRe, S3Gaussians and StreetGS all struggle to model the cyclists, while the tram is poorly modeled by all methods except for UGSDF. While 4DGF models the cyclists well, it does a poor job on the background (check for column 1). For the fourth column, OmniRe and UGSDF are the only ones that capture all the details of the pedestrian. Whereas OmniRe, uses an SMPL mesh as a template, our method does so without the template.**We show additional image and depth results in the *supp***. We also show the **runtime analysis**, **tracking evaluation** of our method versus that of other methods in the ***supp*, as well as *scene-editing results***.

5.2. Ablation and Analysis

Tab. 4 shows the results of the ablation analysis. To understand which components contribute to the improvements in our method, we perform the following ablations: *(1)* We disable *SDF Guidance for Gaussian Primitives* (*SG4GP*) and replace it with the standard adaptive density control used in the original 3DGS [33]. *(2)* We disable *Gaussian-guided Point Sampling for SDF* (*GPS4S*). *(3) (3)* A default design choice in our method is to maintain a dense representation of dynamic objects based on SDF guidance. To evaluate this choice, we create a sparser representation by increasing the threshold τ_n for densification (denoted *w/ sparse* in Table 4). We show additional results and ablation in the **supp**.

Among the components, *SG4GP* proves to be the most crucial, highlighting the importance of SDF representation for improving Gaussian primitive distribution. Maintaining a denser representation is also beneficial, especially for humans, showing a greater effect compared to vehicles.

Figure 5 presents qualitative results of the ablation study. As can be seen, removing SG4GP and using a sparser representation cause artifacting around the dynamic object especially thin ones like the cyclist and pedestrian. The SDF network tends to learn relatively smooth surface representations. When combined with a sparse representation, this can result in under-fitting of the dynamic object. By using a denser representation the under-fitting is mitigated, enabling the SDF network to capture finer details of the object.

6. Conclusions

We propose a method combining Signed Distance Function (SDF) networks for precise geometry and 3D Gaussians for high-quality rendering. This joint learning approach improves the accuracy of each representation, achieving strong results on the NOTR and KITTI datasets by reconstructing both rigid and non-rigid dynamic objects and enabling novel view synthesis—all without ground truth motion, object templates, or 3D bounding boxes. Our method also

Figure 6. **Qualitative comparison of baselines and our method on the NOTR Dataset with zoomed in regions** We show results on some atypical objects observed in urban scenes namely: a tram and two cyclists, as well as a pedestrian. Some of these are tricky cases not always very well modeled by some or all of the baselines. In contrast UGSDF achieves a very high level of fidelity in novel view synthesis for these objects.

supports tasks like object removal, scene composition, and novel view synthesis. Incorporating priors from video generative models would be a promising direction to explore. Also, we intend to explore jointly inferring motion segmentation [71, 74, 75], in addition to scene rendering.

Limitations UGSDF is sensitive to the 2D tracks generated by [32], which can occasionally be inaccurate, leading to poor dynamic estimates of the Gaussians. Additionally, while the SDF network does provide some amount of controllability of the object, it does not have the representation power of the SMPL template. Finally, as with all the baselines, UGSDF produces less satisfactory novel views when the camera deviates significantly from the training trajectories.

References

[1] Jad Abou-Chakra, Feras Dayoub, and Niko Sünderhauf. Particlenerf: A particle-based encoding for online neural radiance fields. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5975–5984, 2024. 4

[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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter 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. 2

[4] Wang Bing, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. 2

[5] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[6] Qi Cai, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. Objectfusion: Multi-modal 3d object detection with object-centric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18067–18076, 2023. 1

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 2

[8] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 2

[9] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *NeurIPS*, 2023. 2

[10] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance, 2023. 2

[11] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 172–181, 2023. 1

[12] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 2023. 1, 6, 7

[13] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 1, 2, 6, 7

[14] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation, 2024. 3, 5

[15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1

[16] Tobias Fischer, Jonas Kulhanek, Samuel Rota Bulò, Lorenzo Porzi, Marc Pollefeys, and Peter Kontschieder. Dynamic 3d gaussian fields for urban areas. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6

[17] Tobias Fischer, Lorenzo Porzi, Samuel Rota Bulo, Marc Pollefeys, and Peter Kontschieder. Multi-level neural scene graphs for dynamic urban environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21125–21135, 2024. 1, 2

[18] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022. 2

[19] 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. In *International Conference on 3D Vision (3DV)*, 2022. 2

[20] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022. 6

[21] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 6

[22] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P.J. Narayanan. Interactive Segmentation of Radiance Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[23] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023. 7

[24] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 5

[25] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2, 5

[26] 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. *arXiv preprint arXiv:2306.04988*, 2023. 1

[27] Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024. 2

[28] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2

[29] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3 gaussians: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 1, 2, 6

[30] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, CVPR 2024. 2

[31] Shengxiang Ji, Guanjun Wu, Jiemin Fang, Jiazhong Cen, Taoran Yi, Wenyu Liu, Qi Tian, and Xinggang Wang. Segment any 4d gaussians. *arXiv preprint arXiv:2407.04504*, 2024. 2

[32] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 2, 3, 4, 6, 7, 8

[33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 3, 7

[34] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 4, 5

[35] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19729–19739, 2023. 2

[36] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21530–21539, 2024. 2

[37] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[38] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 2

[39] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 1, 2

[40] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2

[41] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[42] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[43] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 2

[44] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[45] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023. 4

[46] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lv, Peng Wang, Wenping Wang, and Junhui Hou. Modgs: Dynamic gaussian splatting from causually-captured monocular videos. *arXiv preprint arXiv:2406.00434*, 2024. 2

[47] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 2

[48] Xiaoyang Lyu, Yang-Tian Sun, Yi-Hua Huang, Xiuzhe Wu, Ziyi Yang, Yilun Chen, Jiangmiao Pang, and Xiaojuan Qi. 3dgsr: Implicit surface reconstruction with 3d gaussian splatting, 2024. 2

[49] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 1

[50] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[51] Mirgahney Mohamed and Lourdes Agapito. Dynamicsurf: Dynamic neural rgb-d surface reconstruction with an optimizable feature grid. In *2024 International Conference on 3D Vision (3DV)*, pages 820–830. IEEE, 2024. 4

[52] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41 (4):102:1–102:15, 2022. 2, 4

[53] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[54] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *Robotics: Science and Systems*, 2022. 5

[55] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 1, 2

[56] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[57] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2, 4

[58] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 2, 4, 6, 7

[59] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[60] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. *arXiv preprint arXiv:2312.16084*, 2023. 2

[61] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[62] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 5, 6

[63] Christian Reiser, Stephan Garbin, Pratul P. Srinivasan, Dor Verbin, Richard Szeliski, Ben Mildenhall, Jonathan T. Barron, Peter Hedman, and Andreas Geiger. Binary opacity grids: Capturing fine geometric detail for mesh-based view synthesis. *SIGGRAPH*, 2024. 2

[64] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 1

[65] Reuters. Gm self-driving unit cruise to pay 15 million usd fine over crash disclosure, 2024. 1

[66] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, 2023. 2

[67] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin 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. 2, 6

[68] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 1

[69] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023. 1, 2

[70] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 1, 2

[71] Siddharth Tourani and K Madhava Krishna. Using in-frame shear constraints for monocular motion segmentation of rigid bodies. *Journal of Intelligent & Robotic Systems*, 82 (2):237–255, 2016. 8

[72] Siddharth Tourani, Alexander Shekhovtsov, Carsten Rother, and Bogdan Savchynskyy. Mplp++: Fast, parallel dual block-coordinate ascent for dense graphical models.

In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 251–267, 2018. 2

[73] Siddharth Tourani, Alexander Shekhovtsov, Carsten Rother, and Bogdan Savchynskyy. Taxonomy of dual block-coordinate ascent methods for discrete energy minimization. In *International conference on artificial intelligence and statistics*, pages 2775–2785. PMLR, 2020. 2

[74] Siddharth Tourani, Muhammad Haris Khan, Carsten Rother, and Bogdan Savchynskyy. Discrete cycle-consistency based unsupervised deep graph matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5252–5260, 2024. 8

[75] Siddharth Tourani, Jayaram Reddy, Sarvesh Thakur, K Madhava Krishna, Muhammad Haris Khan, and N Dinesh Reddy. Leveraging cycle-consistent anchor points for self-supervised rgb-d registration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2737–2744. IEEE, 2024. 8

[76] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, 2021. 2

[77] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 2

[78] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023. 1, 2

[79] Haithem Turki, Vasu Agrawal, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Deva Ramanan, Michael Zollhöfer, and Christian Richardt. Hybridnerf: Efficient neural rendering via adaptive volumetric surfaces. In *Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[80] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024. 3

[81] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021. 1

[82] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. In *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022. 5

[83] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, pages 27171–27183. Curran Associates, Inc., 2021. 2, 3, 5

[84] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. 2024. 2, 4, 5

[85] Shizun Wang, Xingyi Yang, Qiuhong Shen, Zhenxiang Jiang, and Xinchao Wang. Gflow: Recovering 4d world from monocular video. *arXiv preprint arXiv:2405.18426*, 2024. 2

[86] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[87] Zian Wang, Tianchang Shen, Merlin Nimier-David, Nicholas Sharp, Jun Gao, Alexander Keller, Sanja Fidler, Thomas Müller, and Zan Gojcic. Adaptive shells for efficient neural radiance field rendering. *ACM Trans. Graph.*, 42(6), 2023. 2

[88] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2

[89] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In *Advances in Neural Information Processing Systems*, pages 32653–32666. Curran Associates, Inc., 2022. 2

[90] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 1, 2

[91] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9421–9431, 2021. 2

[92] Haodong Xiang, Xinghui Li, Xiansong Lai, Wanting Zhang, Zhichao Liao, Kai Cheng, and Xueping Liu. Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. *arXiv preprint arXiv:2405.19671*, 2024. 2, 5, 6

[93] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2

[94] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pages 156–173. Springer, 2024. 2, 6, 7

[95] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views

with gaussian splatting. *ACM Transactions on Graphics*, 2024. 2

[96] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1, 2, 6, 7

[97] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 1, 2

[98] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 2, 4, 6, 7

[99] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2

[100] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. *arXiv*, 2023. 2

[101] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes, 2023. 2

[102] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved rendering and reconstruction. *arXiv preprint arXiv:2403.16964*, 2024. 2, 5, 6

[103] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 2

[104] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 2

[105] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. 1

[106] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:101790–101817, 2024. 2