2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

UnIRe: Unsupervised Instance Decomposition for Dynamic Urban Scene Reconstruction

Yunxuan Mao[1], Rong Xiong[1], Yue Wang[1], Yiyi Liao[1*]

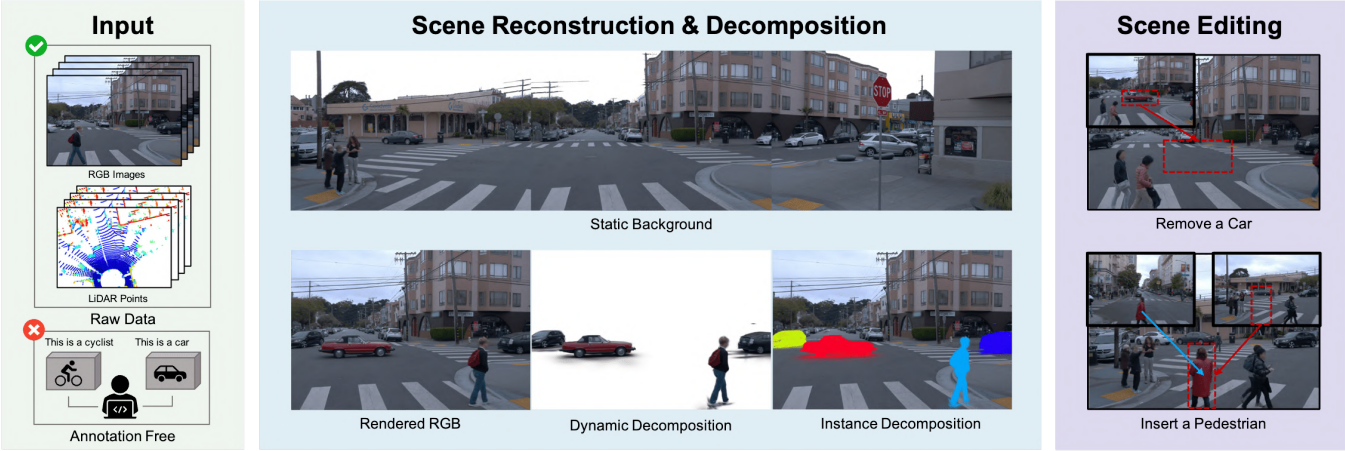


Fig. 1: **UnIRe.** Our method enables dynamic urban scene reconstruction and decomposition without requiring manual annotation. (a) **UnIRe** separates static and dynamic components while achieving instance-aware decomposition of dynamic objects. (b) **UnIRe** also supports scene editing and simulation applications, such as removing a vehicle or adding a pedestrian.

***Abstract*— Reconstructing and decomposing dynamic urban scenes is crucial for autonomous driving, urban planning, and scene editing. However, existing methods fail to perform instance-aware decomposition without manual annotations, which is crucial for instance-level scene editing. We propose UnIRe, a 3D Gaussian Splatting (3DGS) based approach that decomposes a scene into a static background and individual dynamic instances using only RGB images and LiDAR point clouds. At its core, we introduce 4D superpoints, a novel representation that clusters multi-frame LiDAR points in 4D space, enabling unsupervised instance separation based on spatiotemporal correlations. These 4D superpoints serve as the foundation for our decomposed 4D initialization, i.e., providing spatial and temporal initialization to train a dynamic 3DGS for arbitrary dynamic classes without requiring bounding boxes or object templates. Furthermore, we introduce a smoothness regularization strategy in both 2D and 3D space, further improving the temporal stability. Experiments on benchmark datasets show that our method outperforms existing methods in decomposed dynamic scene reconstruction while enabling accurate and flexible instance-level editing, making it a practical solution for real-world applications.**

I. Introduction

The reconstruction and decomposition of dynamic urban scenes is crucial for applications such as autonomous driving, urban planning, and scene simulation and editing. Recent advances in 3D Gaussian Splatting (3DGS) [1] have significantly improved scene reconstruction quality, enabling high-fidelity representations using only 2D supervision. However, achieving instance-level decomposition in dynamic urban scene reconstruction for arbitrary object classes, such as pedestrians and vehicles, remains a significant challenge.

Recently, several approaches have been proposed to address this challenge, broadly classified into scene graph-based methods and self-supervised decomposition methods. Scene graph-based methods [2], [3], [4] model dynamic scenes as structured graphs, where each instance is segmented with 3D bounding box annotations and assigned a canonical space. This formulation provides robust motion initialization and ensures instance-aware decomposition with structured 3D shapes, making it well-suited for scene editing. However, these methods heavily rely on manually labeled bounding boxes that are expensive to obtain, which limits their applicability across diverse urban environments. Self-supervised decomposition methods [5], [6], [7], [8] eliminate the need for manual annotations by learning to distinguish between static backgrounds and dynamic regions directly from RGB images and LiDAR data, enhancing practicality. PVG [5] represents dynamic scenes using short-lived Gaussians, assigning different Gaussians to the same dynamic object at different timestamps. SplatFlow [9] uses self-supervised scene flow to initialize dynamic Gaussians. However, these methods lack a canonical space for each dynamic instance, making it hard to merge information observed across frames. In addition, these methods only decompose

This work was supported by the National Nature Science Foundation of China under Grant 62373322 and Zhejiang Provincial Natural Science Foundation of China under Grant No. LD24F030001.

[1]Yunxuan Mao, Rong Xiong, Yue Wang, Yiyi Liao are with Zhejiang University, Hangzhou, China.
*Corresponding author.

static and dynamic regions, without further decomposing dynamic instances, making object-level editing challenging.

In this work, we propose UnIRe, a 3DGS-based framework that decomposes a scene into a static background and individual dynamic instances using only RGB images and LiDAR point clouds. At its core, UnIRe introduces 4D superpoints (akin to superpixels), a novel representation that clusters multi-frame LiDAR points in 4D space. We first generate over-segmented 4D superpoints by propagating per-frame clustering results using self-supervised flow estimation. Next, we cluster these 4D superpoints leveraging their spatiotemporal correlations, achieving instance-level decomposition for arbitrary dynamic classes, without the need for bounding boxes or object templates. This decomposition serves as the initialization for the canonical space and per-point deformation of the dynamic 3DGS. Furthermore, to prevent overfitting and unstable motion, we introduce a smoothness regularization strategy in both 2D and 3D, improving the motion consistency across different frames. Together, these components enable high-fidelity rendering and instance-aware decomposition, enabling flexible scene editing. Experiments on Waymo [10] and KITTI [11] datasets demonstrate that UnIRe achieves state-of-the-art performance in an annotation-free manner while enabling instance-level editing.

II. Related Work

Dynamic Scene Reconstruction: Neural scene representations [12], [13], [14], [15], [1], [16] have significantly advanced novel view synthesis, inspiring extensive research in dynamic scene reconstruction. NeRF-based methods [17], [18], [19] rely on neural deformation fields and canonical spaces to model motion but lack explicit geometry, limiting their applicability to large-scale, real-world urban environments. Similarly, recent works on 3D Gaussian Splatting (3DGS) [20], [21] employ neural deformation-based motion modeling. However, neural deformation fields are inadequate for capturing large-scale dynamic variations.

To overcome these limitations, recent approaches leverage the explicit nature of 3DGS to represent per-point motion. One strategy extends the spatial distribution of Gaussian points into a four-dimensional space [22], [23], embedding temporal variations directly into Gaussian parameters. This formulation models the same object with different Gaussians at different time steps, leading to increased memory consumption in large scenes and inconsistent motion. Another strategy employs per-point deformation, where each Gaussian is associated with a canonical space to maintain temporal consistency [24], [25], [26].

Urban Scene Reconstruction and Decomposition: Urban scene reconstruction methods can be broadly categorized into annotation-dependent scene graph methods and self-supervised scene decomposition methods. Annotation-dependent methods construct a scene graph where each object is explicitly decomposed into instances using 3D bounding box annotations, enabling dynamic scene representation with instance decomposition. Methods such as MARS [27], UniSim [28], DrivingGaussian [29], StreetGS [2], OmniRe [4], and HUGS [3] follow this paradigm. The scene graph served as an instance-aware canonical space, making object-level editing easy. However, they rely on manually labeled 3D bounding boxes [27], [28], [4] or accurate 3D tracking initialization [29], limiting their scalability across diverse urban environments.

In contrast, self-supervised decomposition methods eliminate the need for annotations by learning to separate static and dynamic components during training. EmerNeRF [6] and SUDS [7] estimate motion using implicit flow fields, constraining scene dynamics via multi-frame optimization. While these NeRF-based methods improve scalability, they often struggle with slow rendering speeds and limited reconstruction quality. PVG [5] and DeSiReGS [8] directly embed temporal variations into Gaussian representations, enabling motion-aware reconstruction without explicit deformation fields. Recently,SplatFlow [9] leverage scene flow to initialize Gaussians in dynamic scenes. However, these methods lack explicit per-instance decomposition and canonical space, making scene editing challenging.

III. Preliminaries

3D Gaussian Splatting: 3D Gaussian Splatting [1] (3DGS) represents a scene as a collection of learnable anisotropic Gaussians, $\mathcal{G} = \{g\}$. Each Gaussian $g = (\boldsymbol{\mu}, \boldsymbol{s}, \boldsymbol{r}, o, \boldsymbol{c})$ is parameterized by the following attributes: a position center $\boldsymbol{\mu} \in \mathbb{R}^3$, a scaling vector $\boldsymbol{s}$, a quaternion $\boldsymbol{r} \in \mathbb{R}^4$, an opacity scalar o, and a color vector $\boldsymbol{c}$, which is represented using spherical harmonics. The spatial distribution of each 3D Gaussian is given by:

$$G(\boldsymbol{x}) = \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}. \quad (1)$$

The covariance matrix is $\Sigma = \mathbf{RSS}^\top\mathbf{R}^\top$, where $\mathbf{S}$ is a diagonal scaling matrix and $\mathbf{R}$ is a rotation matrix, parameterized by the scaling vector $\boldsymbol{s}$ and the quaternion $\boldsymbol{r}$.

To render an image from a given viewpoint, the 3D Gaussian ellipsoids are projected onto a 2D image plane, forming 2D ellipses. The projected Gaussians are sorted in depth order, and the pixel color is obtained via alpha blending:

$$C = \sum_{i\in\mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad (2)$$

where α_i and c_i denote the opacity and color of the i-th Gaussian derived from the learned opacity and spherical harmonics (SH) coefficients of the corresponding Gaussian.

IV. Method

Our method enables dynamic urban scene reconstruction and editing using only RGB images and LiDAR data, without additional annotations. A key requirement for scene editing is the ability to independently manipulate dynamic objects, which requires decomposing dynamic instances and

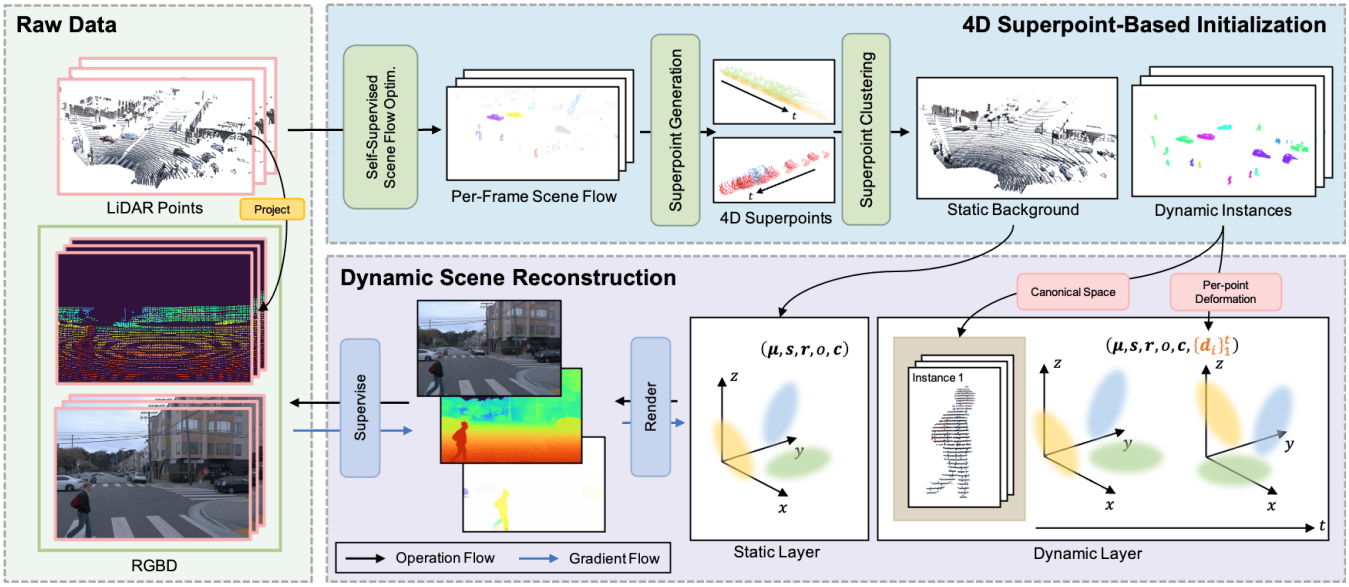


Fig. 3: **Method Overview.** Our method consists of two core components: 4D SuperPoint-Based Initialization and Dynamic Scene Representation. 4D SuperPoint-Based Initialization takes LiDAR points as input and estimates scene flow using a self-supervised optimization method. Then, 4D SuperPoint Generation and 4D SuperPoint Clustering decompose the scene into a static background and dynamic instances. The dynamic instances are further used to construct a canonical space and per-point deformation $\{d_i\}_1^t$. Dynamic Scene Representation utilizes the static background to initialize the static layer, while the canonical space and per-point deformation serve as the initialization for the dynamic layer. The model is supervised by ground truth images and depth maps projected from LiDAR points.

aggregating cross-time information in canonical space. Existing 2D RGB and 3D LiDAR-based detection and tracking methods struggle to deliver robust results in a label-free manner. To address this, we propose a simple yet effective unsupervised approach for scene decomposition based on the spatial-temporal correlation of LiDAR points.

As shown in fig. 3, our method consists of two core components: 4D superpoint-based initialization (section IV-A) and dynamic scene reconstruction (section IV-B), with the former provides spatial and temporal initialization for the latter. Our reconstruction is supervised by ground truth images and depth maps projected from LiDAR points, combined with smooth regularization to improve temporal consistency (section IV-C).

A. 4D Superpoint-Based Initialization

We propose a simple method that decomposes a sequence of LiDAR points based on unsupervised clustering. As shown in fig. 4, per-frame clustering-based decomposition can be inconsistent due to object motion and occlusions, leading to three major challenges:

- Inconsistent Cluster IDs: The same object may be assigned different cluster IDs across frames.
- Over-Decomposition: A single object may be divided into multiple clusters due to temporary occlusions or variations in the density of LiDAR points.
- Under-Decomposition: Spatially close but distinct objects may be mistakenly merged into the same cluster.

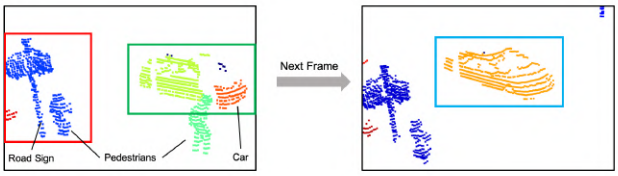


Fig. 4: **Visualization of Decomposition Challenges.** Under-Decomposition (red box), Over-Decomposition (green box), and Inconsistent Cluster IDs (cyan box).

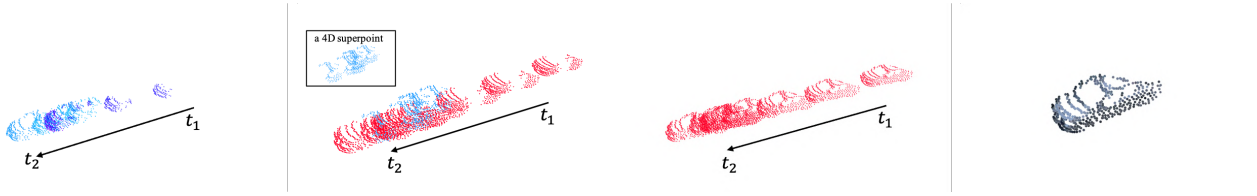
To resolve these challenges, we introduce **4D superpoint**, a spatiotemporal representation that ensures consistent instance decomposition throughout the sequence. Formally, a 4D superpoint is defined as a cluster of points over a sequence of time:

$$\mathcal{S} = \{\mathcal{C}^t\}_{t=1}^{T}, \tag{3}$$

where $\mathcal{C}^t$ is the cluster of points at frame t. The pipeline of 4D superpoint based initialization is shown in fig. 5.

4D Superpoint Generation: Given a sequence of LiDAR points $\{\mathcal{P}^t = \{\mathbf{p}_i^t\}\}_1^T$, we pre-process each frame by removing ground points to stabilize clustering results. Then, we apply DBSCAN [30] independently to each frame to group points into clusters. Next, we apply a self-supervised scene flow optimization method *let it flow* [31] to estimate the scene flow between every two adjacent frames, denoted as $\{\mathcal{SF}^t = \{\mathbf{f}_i^t\}\}_1^{T-1}$.

Next, we establish correspondences between DBSCAN-



(a) DBSCAN (b) 4D Superpoint Generation (c) 4D Superpoint Clustering (d) Canonical Space

Fig. 5: **Visualization of 4D Superpoint-Based Initialization.** To provide an intuitive understanding, we take a car as an example. First, DBSCAN is applied independently to each frame, resulting in inconsistent clustering. Then, we align clusters across frames and form temporally consistent 4D superpoints (different 4D superpoints are in different colors). Finally, these 4D superpoints are clustered to achieve instance-level decomposition and establish a canonical space.

generated clusters across frames. Given a set of clusters $\{\mathcal{C}_k^t\}_{k=1}^{K_t}$ and $\{\mathcal{C}_m^{t+1}\}_{m=1}^{K_{t+1}}$ in frame t and $t+1$, we match clusters based on scene flow consistency. Specifically, for each cluster $\mathcal{C}_k^t$, we find the most likely corresponding cluster in frame $t+1$ by maximizing the number of points that remain spatially aligned after applying scene flow:

$$\mathcal{C}_{\text{match}}^{t+1} = \arg\max_{\mathcal{C}_m^{t+1}} \sum_{\mathbf{p}_i^t \in \mathcal{C}_k^t} \mathbb{1}(\mathbf{p}_i^t + \mathbf{f}_i^t), \tag{4}$$

where $\mathbb{1}(\cdot)$ is an indicator function that counts the number of points in $\mathcal{C}_k^t$ whose scene flow displacement lands within $\mathcal{C}_m^{t+1}$. The cluster ID of $\mathcal{C}_k^t$ is then assigned to $\mathcal{C}_{\text{match}}^{t+1}$, ensuring that cluster IDs remain consistent across frames.

However, due to occlusions, motion variations, and object interactions, clusters may undergo three types of transformations: vanishing, emergence, and splitting:

- Vanishing: If $\mathcal{C}_k^t$ has no valid match in $t+1$, it is registered as vanishing and removed from further tracking.
- Emergence: If a cluster $\mathcal{C}_m^{t+1}$ has no corresponding cluster in t, it is registered as a new cluster.
- Splitting: If a cluster $\mathcal{C}_k^t$ is matched with multiple clusters in $t+1$, it is divided into multiple clusters, and each new cluster maintains its own separate identity.

After applying these alignment rules across the sequence, we obtain a set of 4D superpoints, each 4D superpoint corresponds to a cluster that maintains a consistent identity over time: $\mathcal{S}_k = \{\mathcal{C}_k^t\}_{t=t_1}^{t_2}$, where $\mathcal{C}_k^t$ denotes the cluster state at time t, spanning frames t_1 to t_2.

However, our cluster splitting can lead to over-decomposition, where a single object is unnecessarily divided into multiple clusters, as shown in fig. 5b. To mitigate this issue, we cluster 4D superpoint in the next step by leveraging spatiotemporal similarity.

4D Superpoint Clustering: Over-decomposition from the splitting process results in a single object being divided into multiple 4D superpoints due to inconsistencies in motion and occlusion. To refine these results, we leverage spatiotemporal similarity to cluster 4D superpoints into consistent instances while preserving distinct object identities.

We first estimate the spatiotemporal properties of each 4D superpoint, including its position and motion in each frame. Given a 4D superpoint $\mathcal{S}_k$ at time t, we compute:

$$\boldsymbol{\mu}_k^t = \frac{1}{|\mathcal{S}_k^t|} \sum_{\mathbf{p}_i^t \in \mathcal{S}_k^t} \mathbf{p}_i^t, \quad \mathbf{F}_k^t = \frac{1}{|\mathcal{S}_k^t|} \sum_{\mathbf{p}_i^t \in \mathcal{S}_k^t} \mathbf{f}_i^t, \tag{5}$$

where $\boldsymbol{\mu}_k^t$ represents the spatial centroid of the 4D superpoint, while $\mathbf{F}_k^t$ denotes its average scene flow.

Then, we compute the spatiotemporal similarity matrix $\mathcal{M}$ of the 4D superpoints, which integrates both motion direction and spatial proximity. Given two 4D superpoints $\mathcal{S}_k$ and $\mathcal{S}_l$ in frame t, we define their similarity as

$$\mathcal{M}_{k,l}^t = \lambda \frac{\mathbf{F}_k^t \cdot \mathbf{F}_l^t}{|\mathbf{F}_k^t||\mathbf{F}_l^t|} + (1-\lambda) \exp\left(-\frac{\|\boldsymbol{\mu}_k^t - \boldsymbol{\mu}_l^t\|^2}{\sigma^2}\right), \tag{6}$$

where the first term measures the motion direction similarity using cosine similarity, while the second term captures the spatial proximity, and λ controls the balance between motion and spatial similarity. The final spatiotemporal similarity matrix is obtained by aggregating frame-wise similarities $\mathcal{M} = \sum_t \mathcal{M}^t$.

Finally, we apply DBSCAN to $\mathcal{M}$, which merges overdecomposed 4D superpoints, producing a temporally consistent decomposition for dynamic objects, as shown in fig. 5c. The clustering result is used as the instance decomposition, denoted as $\mathcal{I} = \{\mathcal{I}^t\}_{t=1}^{T}$.

Canonical Space Initialization: For each dynamic instance, we establish a shared canonical shape for initializing the dynamic 3DGS. Specifically, we select a reference frame that provides the most complete observation of each instance. Given an instance $\mathcal{I}_k$ tracked from t_1 to t_2, we define its reference frame t^* as the frame that contains the maximum number of points:

$$t^* = \arg\max_{t \in [t_1, t_2]} |\mathcal{I}_k^t|, \tag{7}$$

where $\mathcal{I}_k^t$ denotes the set of points belonging to instance $\mathcal{I}_k$ at time t. This ensures that the canonical space is defined based on the most complete observation of the instance.

Per-point Deformation Initialization: After obtaining the canonical space, we compute the per-point deformation for each point in the scene, serving as temporal initialization for the dynamic 3DGS. The per-point deformation $\mathbf{d}_i^t$ at time t

is defined as:

$$\mathbf{d}_i^t = (\delta_{\mathbf{x_i}}^t, \delta_{\mathbf{s_i}}^t, \delta_{\mathbf{r_i}}^t) = (\sum_{\tau=t^*}^{t} \mathbf{F}^\tau(\mathcal{I}(\mathbf{p}_i)), 0, 0), \tag{8}$$

where $\delta_{\mathbf{x_i}}^t, \delta_{\mathbf{s_i}}^t, \delta_{\mathbf{r_i}}^t$ are of set $\mathbf{F}^\tau(\mathcal{I})$ represents the cumulative estimated scene flow of instance $\mathcal{I}$ up to time τ, and $\mathcal{I}(\mathbf{p}_i)$ denotes the instance ID of point $\mathbf{p}_i$.

B. Dynamic Scene Reconstruction

After initializing the 3D canonical space and per-point deformations, we build our dynamic scene reconstruction framework by integrating these motion priors into 3D Gaussian Splatting (3DGS). The scene is decomposed into a static layer and a dynamic layer, where instances are classified by thresholding the average scene flow magnitude. Static instances form a set of Gaussians $\mathcal{G}^{static}$, while dynamic instances are represented in a canonical space with per-point deformations applied to each Gaussian.

The complete representation is $\mathcal{G} = (\mathcal{G}^{static}, \mathcal{G}^{dynamic})$, where learnable parameters include Gaussian attributes and deformation fields. To enforce temporal consistency, we render optical flow between frames, using the projected Gaussian centers at two timestamps to compute $\mathbf{f}_i$, and composing them into a dense flow map $\mathcal{F}$ for smoothness regularization.

C. Loss Functions

Our dynamic 3DGS model is trained using RGB images and depth maps projected from LiDAR. However, we found that per-point deformation tends to overfit the training views, causing unstable motion during novel view synthesis. To improve generalization, we introduce two smoothness regularization terms in 2D and 3D space. These terms ensure temporal consistency and spatial coherence in dynamic object motion, promoting stable and smooth trajectories across different viewpoints.

2D Smoothness Regularization: 2D smoothness regularization is commonly used in unsupervised optical flow methods [32], [33], [34], [35], [35], where it helps enforce spatial coherence by penalizing abrupt motion changes. Inspired by these methods, we introduce a similar loss in UnIRe to mitigate overfitting to training views and improve motion consistency in novel view synthesis. Specifically, we define the first-order smoothness loss following [35] as:

$$\begin{aligned}\mathcal{L}_{\text{smooth}}^{\text{2D}} =& \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\lambda \sum_c \left|\frac{\partial I_c}{\partial x}\right|\right)\left|\frac{\partial \mathcal{F}}{\partial x}\right| \\ &+ \exp\left(-\lambda \sum_c \left|\frac{\partial I_c}{\partial y}\right|\right)\left|\frac{\partial \mathcal{F}}{\partial y}\right|,\end{aligned} \tag{9}$$

where I_c denotes the image intensity for color channel c, and λ controls the edge-aware weighting.

3D Smoothness Regularization: While 2D smoothness regularizes optical flow in image space, it does not guarantee consistency in the 3D deformation field. To enhance spatial coherence and suppress abrupt local motion, we introduce a 3D smoothness term that enforces local velocity consistency of per-point deformation.

Specifically, the velocity $\mathbf{v}i$ of each Gaussian $\mathcal{G}i$ is constrained to be close to the mean velocity of its K nearest neighbors:

$$\mathcal{L}\text{smooth}^{3D} = \frac{1}{N}\sum i = 1^N \left|\mathbf{v}_i - \tfrac{1}{K}\sum_k \mathbf{v}_j\right|^2, \tag{10}$$

where $\mathbf{v}_i = \mathbf{d}_i^{t+1} - \mathbf{d}_i^t$.

By combining 2D flow smoothness with 3D deformation smoothness, we mitigate overfitting to training views and improve motion stability in novel view synthesis, leading to more coherent reconstructions.

Full Training Loss: Our full training loss is shown below:

$$\begin{aligned}\mathcal{L} =& \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{opacity}}\mathcal{L}_{\text{opacity}} \\ &+ \lambda_{\text{2s}}\mathcal{L}_{\text{smooth}}^{\text{2D}} + \lambda_{\text{3s}}\mathcal{L}_{\text{smooth}}^{\text{3D}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}},\end{aligned} \tag{11}$$

where $\mathcal{L}_{\text{rgb}}$ supervises rendered images using L1 and SSIM losses, $\mathcal{L}_{\text{depth}}$ aligns the scene with sparse LiDAR depth, and $\mathcal{L}_{\text{opacity}}$ regularizes the opacity of Gaussians to align with the sky model, ensuring proper separation between foreground objects and the background sky. $\mathcal{L}_{\text{reg}}$ represents various regularization terms.

V. Experiments

Datasets: We conduct our experiments on two real-world datasets: Waymo Open Dataset [10] and KITTI Dataset [11]. We use the same scene selections as OmniRe [4] and PVG [5]. Following OmniRe [4], we evaluate our method on image reconstruction and novel view synthesis (NVS) tasks, using every 10th frame as the held-out test set for NVS.

Baselines: We compare our method with several state-of-the-art methods in dynamic urban scene reconstruction: EmerNeRF [6], PVG [5], DeSiRe-GS [8], HUGS [3], StreetGS [2], and OmniRe [4]. Among these methods, EmerNeRF is a NeRF-based self-supervised method. PVG and DeSiRe-GS are 3DGS-based self-supervised methods that incorporate temporal variations in 3D Gaussian representations. HUGS, StreetGS, and OmniRe are scene graph-based approaches that rely on 3D bounding box annotations.

Metrics: We adopt PSNR, SSIM [36] and LPIPS [37] as default settings for quantitative assessment of image reconstruction and novel view synthesis. Additionally, we also use PSNR and SSIM for dynamic regions, following OmniRe, to evaluate the quality of dynamic object reconstruction in the scene. For evaluating the quality of geometry reconstruction, we use Depth L1, which measures the absolute difference between the rendered depth and the ground truth obtained from projected LiDAR point clouds.

A. Experiment Results

Novel View Synthesis: As shown in tables I and II, our method not only achieves state-of-the-art performance across all rendering metrics among annotation-free methods like

Methods	Image Reconstruction								Novel View Synthesis							
	Full Image			Human		Vehicle			Full Image			Human		Vehicle		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	DL1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	DL1 ↓
Annotation-based methods (require GT bounding boxes)																
HUGS [3]	28.26	0.923	0.092	16.23	0.404	24.31	0.794	1.90	27.65	0.914	0.097	15.99	0.378	23.27	0.748	2.13
StreetGS [2]	28.82	0.932	0.087	16.56	0.411	26.65	0.853	2.10	27.19	0.889	0.099	16.28	0.376	23.89	0.775	2.17
OmniRe [4]	34.81	0.956	0.054	27.56	0.828	28.91	0.897	**1.49**	**33.03**	**0.944**	**0.060**	**24.20**	**0.718**	**27.78**	**0.867**	**1.50**
Annotation-free methods																
EmerNeRF [6]	31.51	0.891	0.112	22.73	0.563	24.76	0.735	1.89	29.53	0.878	0.139	21.37	0.483	21.98	0.619	1.97
PVG [5]	32.61	0.936	0.103	24.72	0.712	24.29	0.760	1.86	28.94	0.881	0.127	21.92	0.567	21.59	0.626	1.90
DeSiRe-GS [8]	32.71	0.949	0.103	24.87	0.731	24.51	0.787	1.57	30.67	0.933	0.118	22.53	0.590	22.70	0.658	1.55
Ours	**35.58**	**0.967**	**0.053**	**30.44**	**0.892**	**30.62**	**0.922**	1.63	31.56	0.935	0.074	22.75	0.640	24.82	0.769	1.64

TABLE I: **Quantitative comparison on Waymo Open Dataset.** Best results are highlighted as **first**, second, and third. UniRe performs the best among annotation-free methods, and is also on par with annotation-based methods. DL1 refers to depth L1 (m).

(a) DeSiRe-GS [8] (Annotation-Free) (b) StreetGS [2] (Annotation-Based)

(c) **Ours** (Annotation-Free) (d) GT

Fig. 6: **Novel View Synthesis comparison on Waymo Open Dataset.**

(a) DeSiRe-GS [8] (b) StreetGS [2]

(c) **Ours** (d) GT

Fig. 7: **Novel View Synthesis on KITTI Dataset.**

Methods	Image Reconstruction			Novel View Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Annotation-based methods (require GT bounding boxes)						
HUGS [3]	27.14	0.908	0.082	23.91	0.750	0.094
StreetGS [2]	27.59	0.910	0.065	24.15	0.793	0.084
OmniRe [4]	28.22	0.916	0.072	**26.52**	**0.881**	**0.081**
Annotation-free methods						
EmerNeRF [6]	26.95	0.831	0.197	25.11	0.801	0.227
PVG [5]	27.40	0.895	0.097	24.34	0.819	0.121
DeSiRe-GS [8]	28.62	0.921	0.085	25.32	0.846	0.096
Ours	**28.92**	**0.929**	**0.064**	26.10	0.884	0.079

TABLE II: **Quantitative comparison on KITTI Dataset.**

PVG and DeSiRe-GS, but also delivers performance comparable to OmniRe, which leverages ground-truth bounding boxes and the SMPL model at the cost of requiring extensive annotations and templates.

Qualitative comparisons in fig. 6 and fig. 7 further demonstrate the effectiveness of our approach for reconstructing dynamic objects, whereas DeSiRe-GS often fails to preserve motion details, highlighting the benefit of canonical space. StreetGS also struggles with human reconstruction due to the lack of human-specific modeling.

In contrast, our approach achieves competitive vehicle and human reconstruction results. These results indicate that our 4D-superpoint initialization provides a strong alternative to manual annotations, and that per-point deformation offers a scalable, template-free solution for human reconstruction in urban environments.

Geometry: For geometry evaluation, as shown in fig. 6, our method outperforms all annotation-free methods in depth L1, except for DeSiReGS, which benefits from additional supervision via normal maps predicted by a pre-trained model. This result further demonstrates that improved geometry enhances rendering quality.

	Init.	Image Reconstruction			Novel View Synthesis		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Waymo	casa	34.62	0.936	0.061	30.17	0.902	0.086
	4D-S.P.	**35.58**	**0.967**	**0.053**	**31.56**	**0.935**	**0.074**
KITTI	casa	27.15	0.901	0.087	25.28	0.861	0.102
	4D-S.P.	**28.92**	**0.929**	**0.064**	**26.10**	**0.884**	**0.079**

TABLE III: **Ablation studies on 4D Superpoint (4D-S.P.) Initialization.** Init. indicates initialization method.

(a) AS. DBSCAN (b) PF. DBSCAN (c) Ours

Fig. 8: **Ablation of 4D superpoint Initialization.** AS. DBSCAN refers to applying DBSCAN clustering over the entire sequence, while PF. DBSCAN denotes performing DBSCAN independently on each frame throughout the sequence.

B. Ablation Study

To verify the effectiveness of our proposed methods, we conduct ablation studies on the Waymo and KITTI datasets.

4D Superpoint-Based Initialization: We first evaluate the impact of our 4D initialization by replacing it with bounding boxes predicted by casa [38], [39] for canonical space and per-point deformation. As shown in table III, compared with initialization with predicted bounding boxes, our 4D superpoint-based initialization improves reconstruction accuracy. This highlights the role of our initialization to provide a good prior to the scene representation and preserve high-quality reconstructions across large urban environments. Next, we compare our 4D initialization against naive instance clustering methods. As shown in fig. 8, applying DBSCAN over the entire sequence leads to under-segmentation, while per-frame DBSCAN fails to maintain consistent instance IDs across frames, resulting in inconsistent instance tracking. In contrast, our method effectively decomposes individual instances throughout the sequence.

Temporal Smoothness Regularization: To evaluate the effect of smoothness regularization, we conduct experiments with and without the 2D and 3D smoothness losses. Results in fig. 9 and table IV show that removing smoothness losses leads to unstable motion trajectories in novel view synthesis (NVS) and reduced motion consistency across frames. This ablation experiment emphasizes the importance of smoothness regularization in ensuring temporal consistency and improving generalization in dynamic scene reconstruction.

Settings	Full PSNR ↑		Human PSNR ↑		Vehicle PSNR ↑	
	Recon.	NVS	Recon.	NVS	Recon.	NVS
w/o 2D smooth	35.21	30.97	30.39	21.92	29.79	24.22
w/o 3D smooth	35.32	31.34	30.14	22.61	30.02	24.32
Full model	**35.58**	**31.56**	**30.44**	**22.75**	**30.62**	**24.82**

TABLE IV: **Ablation studies on Smooth Regularization.**

(a) GT (b) w/o 2D s.r. (c) w/ 2D s.r.

Fig. 9: **Ablation of 2D smoothness regularization (s.r.).**

(a) GT (b) Rendering (c) Decomposition

Fig. 10: **Visualization of dynamic instance decomposition.**

Fig. 11: **Scene Editing.** An example of scene editing, where a pedestrian is replaced with another in a different scene.

C. Application

Dynamic Instance Decomposition: Our method excels in the decomposition of dynamic objects in urban environments, offering robust and temporally consistent instance identification across a sequence of frames. By leveraging unsupervised 4D initialization, we achieve accurate and scalable instance decomposition in dynamic urban scene. As shown in fig. 10, UnIRe successfully decomposes vehicles, pedestrians, and other dynamic objects.

Scene Editing: Beyond instance decomposition, our method is also capable of manipulating dynamic urban scenes through scene editing. This includes operations such as object removal, replacement, and motion editing, all while preserving the overall scene structure and consistency. As illustrated in fig. 11, our approach allows for editing of dynamic scenes, enabling realistic modifications with minimal artifacts. This demonstrates the potential of UnIRe for applications in AR/VR, and autonomous driving simulations.

VI. Conclusion

In this paper, we introduce UnIRe, a 3DGS-based approach that decomposes a scene into a static background and individual dynamic instances using only RGB images and LiDAR point clouds, eliminating the need for bounding boxes or object templates. By incorporating 4D superpoints, a novel representation that clusters multi-frame LiDAR points in 4D space, UnIRe facilitates unsupervised instance separation through spatiotemporal correlations, leading to state-

of-the-art performance in image reconstruction and enabling instance-level editing.

REFERENCES

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[2] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians: Modeling dynamic urban scenes with gaussian splatting," in *European Conference on Computer Vision*. Springer, 2024, pp. 156–173.

[3] H. Zhou, J. Shao, L. Xu, D. Bai, W. Qiu, B. Liu, Y. Wang, A. Geiger, and Y. Liao, "Hugs: Holistic urban 3d scene understanding via gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 336–21 345.

[4] Z. Chen, J. Yang, J. Huang, R. de Lutio, J. M. Esturo, B. Ivanovic, O. Litany, Z. Gojcic, S. Fidler, M. Pavone, *et al.*, "Omnire: Omni urban scene reconstruction," *arXiv preprint arXiv:2408.16760*, 2024.

[5] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang, "Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering," *arXiv preprint arXiv:2311.18561*, 2023.

[6] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, *et al.*, "Emernerf: Emergent spatial-temporal scene decomposition via self-supervision," *arXiv preprint arXiv:2311.02077*, 2023.

[7] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, "Suds: Scalable urban dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 375–12 385.

[8] C. Peng, C. Zhang, Y. Wang, C. Xu, Y. Xie, W. Zheng, K. Keutzer, M. Tomizuka, and W. Zhan, "Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes," *arXiv preprint arXiv:2411.11921*, 2024.

[9] S. Sun, C. Zhao, Z. Sun, Y. V. Chen, and M. Chen, "Splatflow: Self-supervised dynamic gaussian splatting in neural motion flow field for autonomous driving," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 27 487–27 496.

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

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[13] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.

[14] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5470–5479.

[15] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[16] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 conference papers*, 2024, pp. 1–11.

[17] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[18] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *arXiv preprint arXiv:2106.13228*, 2021.

[19] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[20] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 331–20 341.

[21] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.

[22] Z. Yang, H. Yang, Z. Pan, and L. Zhang, "Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting," *arXiv preprint arXiv:2310.10642*, 2023.

[23] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen, "4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[24] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 800–809.

[25] Q. Wang, V. Ye, H. Gao, J. Austin, Z. Li, and A. Kanazawa, "Shape of motion: 4d reconstruction from a single video," *arXiv preprint arXiv:2407.13764*, 2024.

[26] Y. Lin, Z. Dai, S. Zhu, and Y. Yao, "Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 136–21 145.

[27] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, *et al.*, "Mars: An instance-aware, modular and realistic simulator for autonomous driving," in *CAAI International Conference on Artificial Intelligence*. Springer, 2023, pp. 3–15.

[28] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1389–1399.

[29] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 634–21 643.

[30] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[31] P. Vacek, D. Hurych, T. Svoboda, and K. Zimmermann, "Let it flow: Simultaneous optimization of 3d flow and object clustering," *arXiv preprint arXiv:2404.08363*, 2024.

[32] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[33] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[34] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4884–4893.

[35] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova, "What matters in unsupervised optical flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 557–572.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[37] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[38] H. Wu, J. Deng, C. Wen, X. Li, and C. Wang, "Casa: A cascade attention network for 3d object detection from lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[39] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, "3d multi-object tracking in point clouds based on prediction confidence-guided data association," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5668–5677, 2021.