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

ArmGS: Composite Gaussian Appearance Refinement for Modeling Dynamic Urban Environments

Guile Wu[1], Dongfeng Bai[1], and Bingbing Liu[1]

***Abstract*— This work focuses on modeling dynamic urban environments for autonomous driving simulation. Contemporary data-driven methods using neural radiance fields have achieved photorealistic driving scene modeling, but they suffer from low rendering efficacy. Recently, some approaches have explored 3D Gaussian splatting for modeling dynamic urban scenes, enabling high-fidelity reconstruction and real-time rendering. However, these approaches often neglect to model fine-grained variations between frames and camera viewpoints, leading to suboptimal results. In this work, we propose a new approach named ArmGS that exploits composite driving Gaussian splatting with multi-granularity appearance refinement for autonomous driving scene modeling. The core idea of our approach is devising a multi-level appearance modeling scheme to optimize a set of transformation parameters for composite Gaussian refinement from multiple granularities, ranging from local Gaussian level to global image level and dynamic actor level. This not only models global scene appearance variations between frames and camera viewpoints, but also models local fine-grained changes of background and objects. Extensive experiments on multiple challenging autonomous driving datasets, namely, Waymo, KITTI, NOTR and VKITTI2, demonstrate the superiority of our approach over the state-of-the-art methods.**

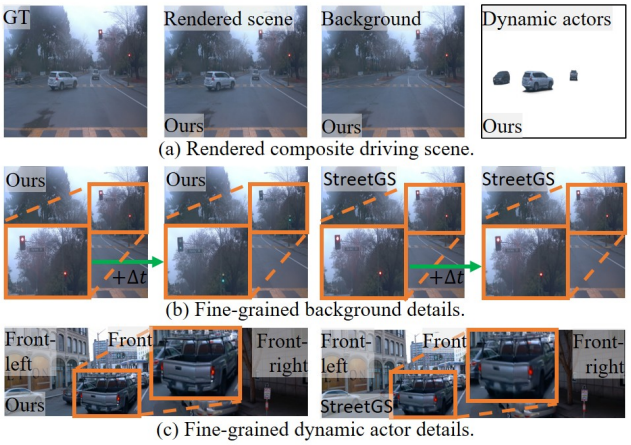

Fig. 1. An illustration of our ArmGS for modeling urban scenes. Our approach is capable of modeling fine-grained changes of background scenes and objects across frames and camera viewpoints.

I. INTRODUCTION

Autonomous driving has made remarkable advancements in recent years, but how to effectively validate the safety and reliability of an autonomous driving system remains an open question. In real-world scenarios, there are many corner cases which cannot be easily collected but are critical to the safety of autonomous driving. Simulation is one of the most useful ways to alleviate this problem. It enables the reconstruction and generation of various challenging driving scenes for downstream closed-loop evaluation.

Traditionally, virtual-engine-based methods, such as CARLA [1], use classic graphics rendering systems for driving scene simulation, but the simulated scenes often have significant gaps from the real-world ones. On the other hand, data-driven simulation enables photorealistic driving scene modeling in a low-cost and efficient way [2]–[7]. With the great success of Neural Radiance Fields (NeRFs) for neural rendering, some researchers have adapted NeRFs for scene modeling in autonomous driving [3], [6]–[8]. Although NeRF-based methods have shown great potential for reconstructing realistic driving scenes, they mostly suffer from long training time and fail to achieve real-time rendering. More recently, some researchers have resorted to 3D Gaussian Splatting (3DGS) [9] to learn explicit 3D scene representations for driving scene modeling [2], [4], [5], [10]–[12]. Due to the efficiency of explicit representations and differentiable tile rasterizers, 3DGS-based methods have shown promising results for high-fidelity reconstruction and real-time rendering. However, existing approaches for driving scene modeling mostly neglect to model fine-grained variations across frames and camera viewpoints. In autonomous driving simulation, scene images are captured in the wild along the movement of the ego vehicle. This unavoidably brings significant appearance changes of scenes and objects across frames and camera viewpoints due to lighting and camera exposure. Thus, failing to model these variations can lead to the loss of fine-grained details and suboptimal results.

In this work, we propose to exploit composite 3D **G**aussian **S**platting with **A**ppearance **R**efine**M**ent (ArmGS) from multiple granularities for dynamic urban environments modeling in autonomous driving. Our approach constructs a composite driving scene model based on 3DGS representations and employs a multi-level appearance modeling scheme to optimize a set of transformation parameters for composite Gaussian refinement. Specifically, we represent a driving scene with composite 3DGS and refine 3DGS at local Gaussian level, global image level and dynamic actor level, respectively. For local Gaussian level refinement, we extract latent Gaussian appearance representation to learn Gaussian-wise transformation parameters for Gaussian appearance refinement. This helps to learn local fine-grained appearance variations in the 3D Gaussian space. For global image level refinement, we extract latent image appearance representation to learn

[1]The authors are with Huawei Noah's Ark Lab. {guile.wu, baidongfeng, liu.bingbing}@huawei.com

global image-wise transformation parameters for image color refinement. This helps to learn global image-wise appearance changes across frames and camera viewpoints. Moreover, to model dynamic variations of moving road actors (such as vehicles), we employ a light-weight spatial-temporal deformation model to transform dynamic actor Gaussians across time. This multi-level appearance modeling scheme allows for refining the composite driving scene Gaussians at multiple granularities. Consequently, scene and road actors appearance variations are collectively modeled to reconstruct richer fine-grained cues of driving scenes. Our experiments on multiple challenging autonomous driving datasets, namely, Waymo [13], KITTI [14], NOTR [3] and VKITTI2 [15], for driving scene modeling and novel view synthesis show that our approach not only reconstructs global scene appearance variations between frames and camera viewpoints, but also models local fine-grained changes of background and dynamic objects. An illustration is shown in Fig. 1. In summary, our **contributions** are: **(I)** Our method is the first method that proposes to explicitly embed multi-granularity appearances to model fine-grained changes for dynamic urban environments modeling, which fills a gap left by existing methods; **(II)** Our method employs a set of transformation parameters for composite Gaussian refinement, which is simple, effective, and preserves the differentiability of the splatting process; **(III)** Our thorough experiments on Waymo, KITTI, NOTR and VKITTI2 show the superiority of our method over the state-of-the-art methods.

II. RELATED WORK

A. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [9] is a recently proposed real-time rendering approach. Unlike NeRFs that employ a continuous radiance field to implicitly represent 3D scenes, 3DGS represents a 3D scene with a collection of explicit 3D Gaussian primitives and utilizes splatting-based rasterization [16] to project 3D Gaussians to 2D image space. Since the success of [9], researchers have made many efforts to improve 3DGS from different aspects [17]–[21], such as anti-aliasing [18], compression [22], surface reconstruction [23], etc. Among them, SWAG [20] proposes to model image appearance changes with image embeddings for novel-view synthesis from unconstrained image collections. Our approach significantly differs from [20] in that we devise a novel multi-level appearance modeling scheme to optimize a set of transformation parameters for composite Gaussian refinement, rather than modeling transient objects and landmarks from unconstrained images. In addition, 4DGS [24] proposes to explore deformation fields with HexPlane representations for real-time dynamic 4D scene rendering, which is subsequently modified by [5] for dynamic urban scene reconstruction. Although we also employ deformation fields for dynamic actor level refinement, our method is different in that we construct a light-weight network structure and encode position, timestamp and spherical harmonics to learn actor variations, rather than constructing HexPlane representation [5], [24].

B. Autonomous Driving Scene Modeling

Autonomous driving simulation [1]–[3], [6], [25], [26] is critical for real-world closed-loop evaluation. For dynamic driving scene modeling, there are primarily two types of methods, namely, virtual-engine-based methods [1], [15] and data-driven methods [2], [3], [6], [10]. While virtual-engine-based methods typically lack scalability and suffer from significant domain gaps, data-driven methods are becoming prevailing in recent years due to their photo-realism, scalability and efficiency. For data-driven simulation, NeRF-based methods [3], [8], [27], [28] have shown compelling performance, but require point dense sampling along each ray to model the interaction between light and scene elements. This leads to expensive computational costs and low rendering speed. On the other hand, 3DGS-based methods [2], [4], [5], [29], [30] make full use of the advantages of 3DGS and achieve real-time rendering for modeling driving scenes. StreetGS [2] proposes to model dynamic urban scenes with road asset pose optimization and 4D spherical harmonics appearance model, but neglects to model fine-grained driving scene variations between frames and camera viewpoints. S^3Gaussian [5] extends dynamic 4DGS [24] for driving scene reconstruction in a self-supervised manner, but the rendering results are suboptimal with some artifacts and do not explicitly model fine-grained changes of driving scenes. Our work also builds on 3DGS, but different from existing works, our approach refines composite driving scene Gaussians with appearance modeling at multiple granularities, ranging from local Gaussians to global images and dynamic actors. Our approach is capable of modeling global scene appearance variations, local fine-grained changes of background and dynamic objects in autonomous driving scenes.

III. METHODOLOGY

A. Preliminaries

a) 3D Gaussian Splatting: Formally, 3DGS [9] consists of a collection of learnable parameters, where each Gaussian is defined by the position (mean) parameter $\mu \in \mathbb{R}^3$, the covariance matrix Σ (which is determined by the rotation parameter $r \in \mathbb{R}^4$ and the scaling parameter $s \in \mathbb{R}^3$), the opacity parameter $o \in \mathbb{R}^1$ and the spherical harmonics parameter $h \in \mathbb{R}^d$ (d is determined by the degrees of spherical harmonics). These parameters collectively represent Gaussians in a 3D space as:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \tag{1}$$

where point x is centered at μ. To render images from different camera viewpoints, 3D Gaussians are projected onto the corresponding image plane and the color C of each pixel is calculated with N-ordered 2D splats using α-blending as:

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

where c_i is the color derived from h and α_i is determined by o and the contribution of the Gaussian.

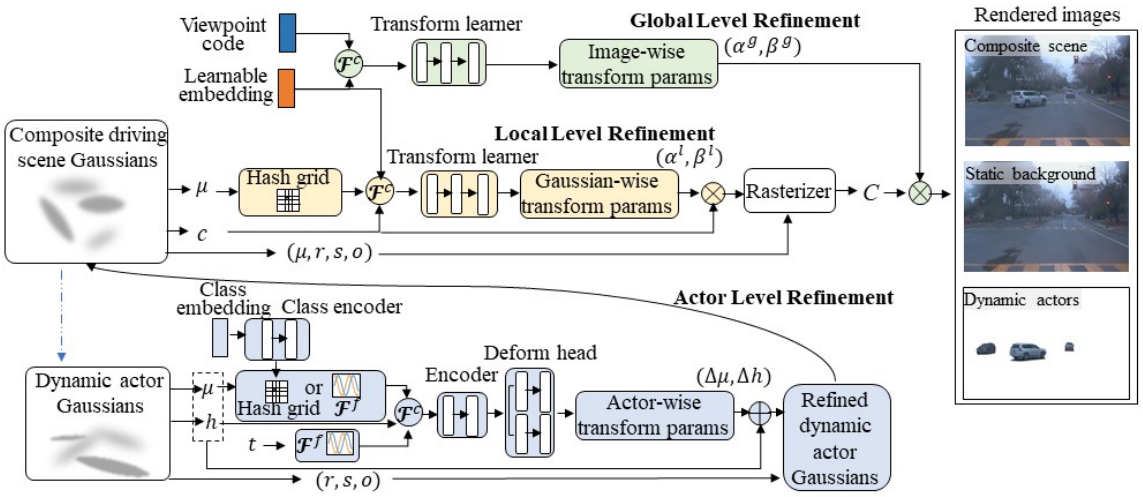

Fig. 2. An overview of our approach. Our approach refines composite driving scene Gaussians with appearance modeling at multiple granularities, ranging from local Gaussians level to global images level and dynamic actors level. The modules for local level refinement, global level refinement and actor level refinement are indicated in yellow, green and blue, respectively.

b) Composite Driving Scene Gaussians: Inspired by [2], [8], [31], we represent a driving scene with composite 3DGS models, including a background Gaussian model, dynamic actor Gaussian models and a sky Gaussian model. In the background Gaussian model, we use a set of 3D Gaussians to model the background scene in the world coordinate space. For each dynamic actor (such as moving vehicle) in a driving scene, we represent it with a separate Gaussian model in the object-centric coordinate space. These dynamic actor Gaussians are transformed to world coordinates with pose transformation matrices to form the composite scene Gaussians. For distant sky region, we construct the sky Gaussian model with explicit cubemap representations [2] and model sky color from view direction. These models collectively form the composite Gaussians to be rendered via a differentiable tile-based rasterizer.

Although composite scene Gaussians can reconstruct 3D driving scenes in an efficient way, existing methods [2], [30], [31] lack the ability to effectively model driving scene appearance variations between frames and camera viewpoints, leading to missing fine-grained details and suboptimal simulation results. To resolve this problem, we present composite 3DGS appearance refinement via a multi-level appearance modeling scheme.

B. The Proposed Approach

An overview of our ArmGS approach is depicted in Fig. 2. Our approach realizes driving scene appearance refinement from three levels, namely, local composite Gaussian level, global image level and dynamic actor level.

a) Local Composite Gaussian Level Refinement: With composite driving Gaussians, we propose to model fine-grained scene variations by learning a set of transformation parameters to refine composite Gaussians. First, to model changes across scene frames, we construct a low-dimensional learnable embedding ϵ for each frame. Then, we employ a multiresolution Hash grid $\mathcal{H}$ [32] to extract richer Gaussian position code and merge the low-dimensional embedding, the position code and the Gaussian color as the *latent Gaussian appearance representation* f^l of each Gaussian. This is formulated as:

$$f^l = \mathcal{F}^c(\mathcal{H}(\mu), \epsilon, c), \tag{3}$$

where the merge function $\mathcal{F}^c$ is concatenation in this work. Then, we employ a local Gaussian appearance affine transformation learner $\mathcal{D}^l$ (a light-weight MLP) with f^l as the input to learn local Gaussian-wise transformation parameters $(\alpha^l \in \mathbb{R}^3, \beta^l \in \mathbb{R}^3)$ for Gaussian appearance refinement, which is defined as:

$$c' = \alpha^l c + \beta^l, \tag{4}$$

where $(\alpha^l, \beta^l) = \mathcal{D}^l(f^l)$. c' is then used in Eq.(2) for color computation. In this way, the contribution of each Gaussian to the rendered image is modulated across frames, so fine-grained scene variations are effectively encoded into local Gaussian-wise appearance variations across frames. During rendering, to obtain a low-dimensional embedding ϵ for a novel view, we consider that the appearance variation of a novel view should be consistent with its adjacent frame. Thus, for each novel view, we calculate the index of its nearest neighbor training frames based on camera index and timestamp to get its corresponding ϵ, and then use Eqs.(3) and (4) for the transformation.

b) Global Image Level Refinement: Although the local Gaussian-wise appearance refinement encourages each Gaussian to learn local fine-grained variations, it does not encourage global consistent variations across camera viewpoints. For example, image-wise appearance variation can be consistently changed due to direct sunlight or camera exposure, which cannot be resolved by image preprocessing and requires the holistic scene appearance refinement. Thus, to compensate for the global image-wise variation, we further refine driving scene appearance at the global image

level. Specifically, we encode camera location and viewing direction as a camera viewpoint code ϕ and merge it with the embedding ϵ as the *latent image appearance representation* f^g, which is defined as:

$$f^g = \mathcal{F}^c(\epsilon, \phi). \tag{5}$$

We then employ an image appearance affine transformation learner $\mathcal{D}^g$ with f^g as the input to learn global image-wise transformation parameters ($\alpha^g \in \mathbb{R}^{3\times3}, \beta^g \in \mathbb{R}^{1\times3}$) for image color transformation, which is formulated as:

$$C' = \alpha^g C + \beta^g, \tag{6}$$

where $(\alpha^g, \beta^g) = \mathcal{D}^g(f^g)$. With the collaboration of local Gaussian-wise appearance modeling and global image-wise appearance modeling, the composite driving scene Gaussians are modulated at different granularities.

c) Dynamic Actor Level Refinement: In dynamic driving scene modeling, dynamic road actors usually have more complex motion and appearance variations than the background and static objects. For example, a moving vehicle needs to be placed at different locations along its trajectory, while its appearance is affected not only by the scene but also by its actions, such as using brake signals. To deal with this problem, we also refine composite driving Gaussians at the dynamic actor level. Specifically, we employ a low-dimensional learnable embedding for each class of dynamic actors and construct a light-weight class encoder to learn the weights of the dynamic actor Hash grid, inspired by [6], [33]. This helps to encode class information into the Hash grid for position encoding. Alternatively, to reduce training parameters, we observe that the simple yet effective sinusoidal encoding [34] also works well. Then, to model the complex appearance variations specific to each dynamic actor, we construct a light-weight spatial-temporal encoder and a light-weight deformation head to transform position and spherical harmonics attributes of dynamic actor Gaussians across time. Instead of using the HexPlane [24] to learn the spatial-temporal structure, we encode position μ with the encoding function $\mathcal{F}^a$ (class-wise Hash grid or sinusoidal encoding), timestamp t with a sinusoidal function $\mathcal{F}^f$ and merge them with the spherical harmonics h to learn the *spatial-temporal actor representation* f^a of the dynamic actor. This is defined as:

$$f^a = \mathcal{D}^a(\mathcal{F}^c(\mathcal{F}^a(\mu), \mathcal{F}^f(t), h)), \tag{7}$$

where $\mathcal{D}^a$ is a shared spatial-temporal representation encoder. Then, we use a deformation head (a multi-head MLP) $\mathcal{D}^h$ with f^a as the input to learn deformation for Gaussian position and color refinement, as:

$$\{\mu', h'\} = \{\mu + \Delta\mu, h + \Delta h\}, \tag{8}$$

where $\{\Delta\mu, \Delta h\} = \mathcal{D}^h(f^a)$. Note that, although Gaussian deformation is inspired by [24], our design differs from existing works [5], [24], [29] in that we construct light-weight encoders and deformation heads and encodes position, timestamp and spherical harmonics to learn actor variations, rather than constructing HexPlane representation [5], [24] or using neural fields for color computation [29]. Moreover, the collaboration of the local level refinement, the global level refinement and the dynamic actor level refinement brings superior performance for dynamic urban environments modeling in autonomous driving.

C. Composite Driving Gaussian Optimization

a) Training Objective: We optimize our approach in an end-to-end differentiable rendering manner. The training objective loss is defined as:

$$\mathcal{L} = (1 - \lambda_1)\mathcal{L}_{rgb} + \lambda_1\mathcal{L}_{ssim} + \lambda_2\mathcal{L}_d + \lambda_3\mathcal{L}_s + \lambda_4\mathcal{L}_f, \tag{9}$$

where λ_i is the weight coefficient, $\mathcal{L}_{rgb}$ and $\mathcal{L}_{ssim}$ are the image reconstruction loss between rendering images and ground-truth images following [9], $\mathcal{L}_d$ is a depth loss calculated by a L1 loss between rendering depth maps and LiDAR depth, $\mathcal{L}_s$ is a sky mask loss computed by a binary cross-entropy loss between rendering sky masks and pre-extracted sky masks [35], and $\mathcal{L}_f$ is a foreground decomposition loss [2] calculated by an entropy loss on the accumulated alpha values of dynamic actors.

b) Actor Pose Optimization: For pose transformation parameters (the rotation quaternion R_t and the translation T_t) of each actor, we set them as learnable parameters and initialize them with the provided tracked boxes. The position and rotation of each Gaussian of each actor in the world space are defined as $\mu = R_t\mu^a + T_t$ and $r = R_t r^a$, where μ^a and r^a are dynamic actor poses in the object-centric coordinate space. These transformation parameters are directly optimized with scene reconstruction through gradient back-propagation. When rendering novel views, we interpolate actor poses based on timestamps and the optimized poses.

IV. EXPERIMENT

A. Datasets and Evaluation Protocol

a) Datasets: We conduct experiments on four challenging autonomous driving datasets, including the Waymo [2], [13], KITTI [14], NOTR [3] and Virtual KITTI 2 (VKITTI2) [15]. On Waymo, following [2], we select eight challenging sequences with dynamic actors and complex background conditions and use the tracked boxes provided by [2] for experiments. We select every fourth frame as the testing novel view and use the remaining frames as the training views. We set the image resolution to 1066×1600 and evaluate both the novel view synthesis setting with the testing views and the image reconstruction setting with the training views. On KITTI, following [27], [36], we select three sequences and use the official tracklets for evaluation in the novel view synthesis setting. We select every second frame as the testing novel view while the others are used for training. The image resolution is set to 375×1242. On NOTR, following [3], we employ two splits, namely, the static-32 and dynamic-32 splits, which consist of 64 multi-camera driving scene sequences with various challenges, e.g., cross-camera background variations, various dynamic actors, etc.. The image resolution is set to 640×960. On VKITTI2, following [31], we select two sequences for experiments and

TABLE I

QUANTITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON WAYMO AND KITTI.

Method	Waymo Reconstruction			Waymo Novel View			KITTI		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSG [8]	-	-	-	28.3	0.862	0.346	21.3	0.659	0.266
SUDS [36]	-	-	-	-	-	-	23.1	0.821	0.135
3DGS [9]	32.5	0.938	0.089	29.6	0.918	0.117	19.2	0.739	0.174
S^3Gaussian [5]	35.6	0.946	0.088	32.8	0.931	0.097	-	-	-
EmerNeRF [3]	34.5	0.913	0.123	30.9	0.905	0.133	-	-	-
ML-NSG [27]	-	-	-	-	-	-	27.5	0.898	0.055
StreetGS [2]	36.3	0.948	0.074	34.6	0.938	0.079	27.8	0.889	0.069
OmniRe [4]	35.9	0.953	0.108	31.9	0.899	0.126	-	-	-
SplatFlow [37]	-	-	-	-	-	-	27.9	0.927	0.093
ArmGS (Ours)	**38.1**	**0.957**	**0.064**	**35.7**	**0.944**	**0.074**	**30.3**	**0.931**	**0.036**

Fig. 3. Qualitative comparison with the state-of-the-arts on Waymo. From the first to the fourth row, we show results of scene modeling under foggy, sunny, cloudy, and rainy conditions. We highlight some fine-grained details, e.g., traffic lights, vehicles and trees.

use the 50% dropout rate evaluation protocol. The image resolution is set to 375×1242.

b) Evaluation Metrics: Following [2], [3], [5], we use PSNR, SSIM [38] and LPIPS [39] as the evaluation metrics.

c) Compared Methods.: We compare our ArmGS approach with several state-of-the-art methods which can be categorized into two types: (1) 3DGS-based methods, including StreetGS [2], OmniRe [4], SplatFlow [37], S^3Gaussian [5], 3DGS [9] and HUGS [31]; (2) NeRF-based methods, including EmerNeRF [3], ML-NSG [27], SUDS [36], NSG [8], StreetSurf [28] and MARS [7].

B. Implementation Details

We implement our approach with Python and PyTorch. We initialize driving scene Gaussians with LiDAR point clouds and SfM points extracted by COLMAP, and train our model with 30000 iterations using the Adam optimizer. We adopt the training and evaluation splits for each dataset as described in Sec. IV-A. We set the initial position learning rate to $1.6e^{-4}$ and decay it to $1.6e^{-6}$. The learning rates for rotation, scaling, opacity and spherical harmonics are set to $1e^{-3}$, $5e^{-3}$, $5e^{-2}$ and $2.5e^{-3}$, respectively. We construct the local affine transformation learner with three linear layers, the global affine transformation learner with four linear layers and the class encoder with two linear layers, where ReLU activations are used between layers. For the spatial-temporal encoder and the deformation head, we employ two linear layers with ReLU activations between layers. By default, we use $\mathcal{F}^f$ for dynamic actor position encoding. For training loss, we empirically set λ_1=0.2, λ_2=0.01, λ_3=0.05 and λ_4=0.1. Gaussian splitting and merging for adaptive density control are performed every 100 iterations from 500 iterations and until 15000 iterations.

C. Evaluation on Waymo

We present the quantitative results on Waymo in Tab. I. We can see that our approach achieves significantly better performance compared with the state-of-the-arts in both the image reconstruction and the novel view synthesis settings. Specifically, for image reconstruction, our approach achieves the best PSNR of 38.1 dB, SSIM of 0.957 and LPIPS of 0.064, outperforming the state-of-the-art competitors, such as StreetGS, OmniRe and EmerNeRF. For novel view synthesis, our approach still achieves the best results compared with the state-of-the-art methods. For instance, our approach achieves PSNR of 35.7 dB, which significantly surpass the

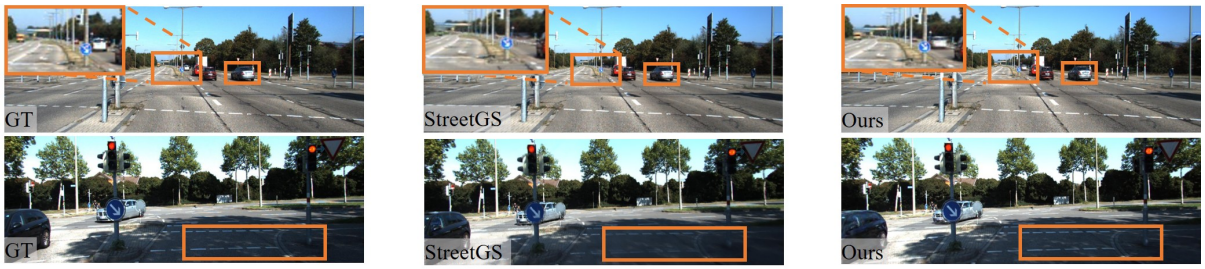

Fig. 4. Qualitative comparison with the state-of-the-art methods on KITTI. We highlight some fine-grained details, e.g., vehicles and lane lines.

TABLE II
RESULTS ON NOTR. PSNRS ARE REPORTED.

Method	Static-32	Dynamic-32
StreetSurf [28]	26.2	-
EmerNeRF [3]	29.1	28.9
StreetGS [2]	29.6	29.7
ArmGS (Ours)	**30.5**	**30.5**

second-best StreetGS by 1.1 dB. Furthermore, in Fig. 3, we present some qualitative results of scene modeling under foggy, sunny, cloudy and rainy conditions on Waymo. It can be seen that our approach renders more fine-grained details of the background scenes and dynamic actors. For example, as shown in the first row, most methods fail to render the appearance change of traffic lights, while our approach render this fine-grained variation; and as shown in the second row, our approach is able to model appearance of the distant vehicles, while other methods render results with artifacts. Moreover, our experimental results show that the rendering speeds of the compared NeRF-based methods, e.g., EmerNeRF, are usually less than 1 FPS; in comparison, the 3DGS-based methods (including our approach) are able to achieve real-time rendering (>30FPS).

D. Evaluation on KITTI

We report the quantitative and qualitative results on the KITTI dataset in Tab. I and Fig. 4, respectively. Overall, our approach achieves significantly better performance compared with the state-of-the-art methods. Specifically, from Tab. I, we can see that our approach achieves the best PSNR of 30.3 dB, SSIM of 0.931 and LPIPS of 0.036, significantly outperforming the state-of-the-art methods. Furthermore, from Fig. 4, we can see that compared with StreetGS, our approach is able to render better image quality with more fine-grained details, such as vehicles highlighted in the first row and the lane line highlighted in the second row.

E. Evaluation on NOTR

To further probe the upper bound reconstruction capability of our approach, we follow [3] to conduct the scene reconstruction experiment on the NOTR dataset [3]. We report the quantitative and qualitative results in Tab. II and Fig. 5, respectively. As shown in Tab. II, compared with the state-of-the-art StreetSurf [28], EmerNeRF [3] and StreetGS [2], our approach achieves better performance on both the static-32 and dynamic-32 splits. From Fig. 5, we can see that our approach is able to model fine-grained details of driving scenes across cameras; in comparison, StreetGS renders driving scenes with more artifacts, such as blurred vehicles. These results show that our approach can faithfully model complex driving scenes and render high-quality results.

Fig. 5. Qualitative comparison on NOTR. We highlight some fine-grained details, e.g., vehicles and pedestrians.

F. Evaluation on VKITTI2

Our approach is not limited to modeling real-world driving scenes, but can also be used to model synthetic driving scenes. To show the performance of our approach for synthetic driving scenes modeling, we follow [31] conducting experiments on the VKITTI2 dataset [15]. We report the results in Tab. III. Although synthetic scenes usually have fewer appearance changes of scenes and objects across frames, our approach still achieves better results compared with the state-of-the-art HUGS.

Fig. 6. Urban scene modeling under diverse conditions using our approach.

TABLE III
RESULTS ON VKITTI2.

Method	PSNR↑	SSIM↑	LPIPS↓
NSG [8]	21.0	0.704	0.316
MARS [7]	22.2	0.869	0.131
3DGS [9]	20.6	0.892	0.103
HUGS [31]	26.4	0.916	0.035
ArmGS (Ours)	**31.1**	**0.957**	**0.016**

TABLE IV
EFFECTIVENESS ANALYSIS OF DYNAMIC ACTOR MODELING ON WAYMO. PSNRS OF THE MOVING OBJECT REGIONS ARE REPORTED.

Method	Novel	Recon.
Ours	**30.9**	**35.8**
Ours w/ class-wise actor encoding	**30.9**	35.7
Ours w/o dynamic actor refinement	30.7	35.4
Ours w/ 4DGS HexPlane	30.6	35.7
EmerNeRF	21.7	26.8
StreetGS	30.2	34.8

TABLE V
EFFECTIVENESS ANALYSIS OF LOCAL AND GLOBAL APPEARANCE REFINEMENT ON WAYMO. RESULTS ARE IN TERMS OF PSNR.

Method	Novel	Recon.
Ours	**35.7**	**38.1**
Ours w/o local level refinement	35.0	36.8
Ours w/o global level refinement	35.4	37.7
Ours w/ HUGS Appearance	34.8	36.5

TABLE VI
ABLATION ANALYSIS OF MORE COMPONENTS.

Method	PSNR↑
ArmGS	38.1
without LiDAR depth loss	37.9
without LiDAR initial points	37.5
without sky GS model and mask	37.6
without actor pose optimization	37.7
add camera pose optimization	38.1

G. Ablation Study

a) Dynamic Actor Modeling Effectiveness Analysis: To evaluate the effectiveness of our dynamic actor level refinement, we follow [2] projecting the 3D bounding box onto the 2D image plane to obtain the mask regions of moving objects and compute PSNR within the mask regions. This mitigates the effect of background on dynamic actors when evaluating dynamic actors. As shown in Tab. IV, our approach achieves better results for the rendered dynamic actor regions compared with StreetGS and EmerNeRF. When using class-wise actor encoding, our approach achieves similar performance. When the dynamic actor level refinement is removed, the performance of our approach degrades in both reconstruction and novel view synthesis settings. When replacing our design with the 4DGS HexPlane [24], the result becomes worse. We conjecture that, compared with our design, the 4DGS HexPlane requires structured factorization for spatial-temporal modeling which constrains its capability to capture variations of dynamic actors.

b) Local and Global Appearance Refinement Effectiveness Analysis: In Tab. V, we evaluate the effectiveness of the local composite Gaussian level refinement and the global image level refinement. It can be seen that our approach with both local and global appearance refinement performs the best, while without using local level refinement or global level refinement, the performance of our approach degrades. Besides, using HUGS appearance modeling in lieu of our design results in worse performance.

c) Ablation Analysis of More Components: In Tab. VI, we report ablation analysis of more components. We can see that: (1) LiDAR depth loss has a minor impact on our approach; (2) Without LiDAR initial points, the performance of our approach only slightly decreases; (3) Sky Gaussians can slightly affect the model performance; (4) Actor pose optimization helps the model to achieve better performance; (5) Adding additional camera pose optimization to our approach does not bring performance improvement.

d) Urban Scene Modeling under Diverse Weather Conditions: Real-world urban scenes often involve diverse weather conditions. Although our approach is not specifically designed for urban scene modeling under extreme weather conditions, our approach still performs well for different urban scene conditions, e.g., foggy, sunny, cloudy, rainy and nighttime. Fig. 6 shows some results of our approach for urban scene modeling under diverse conditions.

V. CONCLUSION

This work presents a composite Gaussian appearance refinement approach to urban environments modeling. The key idea is to learn a set of transformation parameters to refine 3D Gaussians from multi-level granularities, ranging from local Gaussian level to global image level and dynamic actor level. Our thorough experiments on four autonomous driving datasets demonstrate that our approach is capable of rendering more fine-grained details of driving scenes across frames and camera viewpoints, achieving superior performance compared with the state-of-the-art methods.

REFERENCES

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[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] 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," in *International Conference on Learning Representations*, 2024.

[4] Z. Chen, J. Yang, J. Huang, R. de Lutio, J. M. Esturo, B. Ivanovic, O. Litany, Z. Gojcic, S. Fidler, M. Pavone, L. Song, and Y. Wang, "Omnire: Omni urban scene reconstruction," in *The Thirteenth International Conference on Learning Representations*, 2025.

[5] N. Huang, X. Wei, W. Zheng, P. An, M. Lu, W. Zhan, M. Tomizuka, K. Keutzer, and S. Zhang, "S^3gaussian: Self-supervised street gaussians for autonomous driving," *arXiv preprint arXiv:2405.20323*, 2024.

[6] 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.

[7] 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.

[8] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide, "Neural scene graphs for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2856–2865.

[9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[10] S. Miao, J. Huang, D. Bai, X. Yan, H. Zhou, Y. Wang, B. Liu, A. Geiger, and Y. Liao, "Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 286–11 296.

[11] 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.

[12] 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.

[13] 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.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[15] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

[16] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa volume splatting," in *Proceedings Visualization, 2001. VIS'01.* IEEE, 2001, pp. 29–538.

[17] G. Feng, S. Chen, R. Fu, Z. Liao, Y. Wang, T. Liu, B. Hu, L. Xu, Z. Pei, H. Li, *et al.*, "Flashgs: Efficient 3d gaussian splatting for large-scale and high-resolution rendering," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 652–26 662.

[18] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-splatting: Alias-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 447–19 456.

[19] D. Zhang, C. Wang, W. Wang, P. Li, M. Qin, and H. Wang, "Gaussian in the wild: 3d gaussian splatting for unconstrained image collections," *arXiv preprint arXiv:2403.15704*, 2024.

[20] H. Dahmani, M. Bennehar, N. Piasco, L. Roldao, and D. Tsishkou, "Swag: Splatting in the wild images with appearance-conditioned gaussians," in *ECCV*, 2024.

[21] C. Qian, Y. Guo, W. Li, and G. Markkula, "Weathergs: 3d scene reconstruction in adverse weather conditions via gaussian splatting," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 185–191.

[22] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang, "Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps," in *Advances in Neural Information Processing Systems*, 2024.

[23] X. Lyu, Y.-T. Sun, Y.-H. Huang, X. Wu, Z. Yang, Y. Chen, J. Pang, and X. Qi, "3dgsr: Implicit surface reconstruction with 3d gaussian splatting," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–12, 2024.

[24] 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.

[25] Y. Liu, Z. Yang, G. Wu, Y. Ren, K. Lin, B. Liu, Y. Liu, and J. Shan, "Vqa-diff: Exploiting vqa and diffusion for zero-shot image-to-3d vehicle asset generation in autonomous driving," in *European Conference on Computer Vision*. Springer, 2024, pp. 323–340.

[26] Y. Liu, K. Zhu, G. Wu, Y. Ren, B. Liu, Y. Liu, and J. Shan, "Mv-deepsdf: Implicit modeling with multi-sweep point clouds for 3d vehicle reconstruction in autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8306–8316.

[27] T. Fischer, L. Porzi, S. R. Bulo, M. Pollefeys, and P. Kontschieder, "Multi-level neural scene graphs for dynamic urban environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 125–21 135.

[28] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li, "Streetsurf: Extending multi-view implicit surface reconstruction to street views," *arXiv preprint arXiv:2306.04988*, 2023.

[29] T. Fischer, J. Kulhanek, S. R. Bulò, L. Porzi, M. Pollefeys, and P. Kontschieder, "Dynamic 3d gaussian fields for urban areas," in *Advances in Neural Information Processing Systems*, 2024.

[30] Y. Ren, G. Wu, R. Li, Z. Yang, Y. Liu, X. Chen, T. Cao, and B. Liu, "Unigaussian: Driving scene reconstruction from multiple camera models via unified gaussian representations," *arXiv preprint arXiv:2411.15355*, 2024.

[31] 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.

[32] 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.

[33] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[34] 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.

[35] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[36] 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.

[37] 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.

[38] 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.

[39] 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.