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

LangEditor: Natural Language-Driven 4D Editing for Improved Controllability of Dynamic Driving Scenes

Xiaoyu Liang[1], Linhui Wang[1], Chunlam Li[1], Junhong Lin[1], Wei Gao[1]

***Abstract*— Diverse and realistic data are essential for developing reliable autonomous driving (AD) systems, yet collecting and annotating large-scale real-world driving datasets is costly and time-consuming. Recent advances in synthetic scene generation and editing have enabled the creation of diverse driving scenarios. However, fully synthetic scenes often lack real-world grounding, while existing editing approaches are either limited to pure video manipulation or involve cumbersome manual operations. To solve this, we present LangEditor, the natural language-driven 4D editing framework for dynamic driving scenes. LangEditor automatically grounds free-form language instructions to target vehicles and their editable attributes, generating physically plausible trajectories consistent with scene semantics. To ensure spatiotemporal coherence and visual fidelity, we propose a joint refinement strategy that integrates a Dynamic Illumination-Aware Shadow Module for lighting consistency across space-time, and an Appearance Refinement module for synthesizing high-quality textures and material properties. Extensive experiments on realistic driving datasets demonstrate that LangEditor enables intuitive, fine-grained, and photorealistic 4D scene manipulation, outperforming existing baselines in both editing quality and controllability. Our approach bridges the gap between realistic scene editing and user-friendly controllability, offering a powerful tool for data augmentation and simulation in AD research.**

I. INTRODUCTION

Diverse and high-quality data is essential for training robust autonomous driving (AD) systems. To meet this need, numerous large-scale open-source datasets for AD have been released [1]–[3]. However, collecting and annotating real-world driving data remains both costly and logistically challenging. Consequently, recent researchs [4]–[6] have increasingly explored the synthesis of photorealistic driving scenes or the editing of new scenarios from existing driving logs. While fully synthetic scenes offer flexibility and scalability, they often lack grounding in real-world conditions, which can limit their reliability for safety-critical applications. Scene editing methods, in contrast, leverage real-world driving logs as a foundation, enabling the simulation of diverse traffic layouts and behaviors while retaining realism. Several approaches have been proposed to enhance the diversity of driving scenes through editing. Video-based editing models [4], [7] formulate the task as a mask-driven video editing problem, where predefined masks dictate the spatial placement of vehicles. However, these methods primarily operate in 2D or 2.5D space, without explicit modeling of the full 4D spatiotemporal structure, resulting in limited spatial understanding and temporal coherence. Moreover, creating accurate mask sequences is labor-intensive and non-trivial for users, as it requires precise grounding and manual adjustments.

Fig. 1. The diagram of the proposed LangEditor. We use a text prompt to identify the object to be edited and determine the new trajectory, then modify the position of the vehicle within the scene for 4D editing.

Reconstruction-based methods [8]–[12] use 3D Gaussian Splatting or Neural Radiance Fields (NeRF) to represent driving scenes in a fully 4D way. This approach captures complex spatial and temporal relationships. However, these methods are mainly designed for the scene reconstruction task, not editing. As a result, object manipulation is cumbersome and often leads to low-quality or unrealistic results.

To address these limitations, we propose LangEditor, a natural language-driven 4D scene editing framework for dynamic driving environments. LangEditor enables intuitive and precise control of scene vehicles by automatically grounding natural language instructions to the corresponding entities and editable attributes, followed by the generation of physically plausible trajectories that respect scene semantics. To ensure both spatiotemporal consistency and visual fidelity, we introduce a joint refinement strategy: Dynamic Illumination-Aware Shadow Modeling (DIASM) dynamically adjusts lighting conditions across spatiotemporal coordinates, ensuring consistent illumination under changing viewpoints and time steps. Appearance Refinement synthesizes high-fidelity textures and material properties,

[1]School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China

*This work was supported by National Science and Technology Major Project (2024ZD01NL00101), Natural Science Foundation of China (62271013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017, SYSPG20241211173440004), Shenzhen Fundamental Research Program (GXWD20201231165807007-20200806163656003). (*Corresponding author: Wei Gao, email: gaowei262@pku.edu.cn*)

preserving realism while accommodating the edited geometry and motion. This integrated pipeline enables coherent 4D scene manipulation, surpassing the limitations of conventional video editing paradigms and bridging the gap between realism, controllability, and efficiency. Our main contributions can be summarized as follows:

- To enable improved intuitive and controllable manipulation of scene vehicles, this paper proposes the LangEditor framework, which utilizes the natural language-driven 4D editing approach for dynamic driving scenes.
- We design an automatic vehicle selection and trajectory editing mechanism that maps natural language instructions to target entities and their editable attributes, leveraging 4D scene representations to maintain spatial and temporal consistency.
- We propose a joint refinement strategy, including a Spatial Illumination Adaptation module and an Appearance Refinement module, to produce photorealistic and physically plausible results.
- Extensive experiments demonstrate that LangEditor achieves state-of-the-art performance in both editing quality and user controllability.

II. RELATED WORKS

A. Driving Scene Editing

Recent research in driving scene editing has explored various techniques for scene manipulation. MultiEditor [13] enables controllable multimodal object editing in driving scenes using 3D Gaussian Splatting priors. SceneCrafter [14] allows multi-view controllable scene editing, focusing on visual consistency. DriveEditor [4] provides 3D information-guided object editing with high precision, but does not offer easy-to-use language-based editing. GenMM [15] generates temporally and geometrically consistent multimodal data for video and LiDAR, though it lacks direct editing capabilities. These methods significantly advance scene reconstruction and editing but still face challenges in intuitive, dynamic, and language-driven scene manipulation. Our work addresses these limitations by providing an interactive, language-guided 4D editing framework.

B. Reconstruction for Autonomous Driving

Reconstruction-based methods such as DrivingRecon [16], Desire-GS [17], DrivingGaussian [10], Street Gaussians [8], and EmerNerf [11] leverage Gaussian Splatting or NeRF to reconstruct photorealistic driving scenes with temporal coherence, providing a solid foundation for 4D editing. However, they primarily focus on scene reconstruction rather than interactive manipulation. Rotating vehicles in such reconstructions often reveals unseen areas not captured in the original data, which these models cannot reliably synthesize. Extensions like AutoSplat [18] and CoDa-4DGS [19] improve fidelity and handle dynamic changes, yet they still lack intuitive, prompt-driven editing mechanisms. In contrast, our work addresses both the unseen-area challenge and the need for user-friendly, language-driven 4D editing.

C. Diffusion Models for Video Editing

Diffusion models have emerged as powerful tools for video editing, enabling high-fidelity, temporally consistent modifications guided by text or masks. T2V-Zero [20] enables zero-shot video editing using a textual prompt, but the generation quality is average. Tune-A-Video [21] adapts pretrained text-to-image diffusion models for video-to-video editing via attention-based keyframe tuning, but struggles with complex scene geometry. Fresco [22] shows high motion-prompt undertstanding, but is not suitable for outdoor complex scenes. Video-P2P [23] extends Prompt-to-Prompt editing to videos, preserving structure while altering appearance, though it is limited to viewpoint-consistent 2D edits. FateZero [24] improves temporal coherence through cross-frame attention and latent space warping, but still lacks explicit 3D or 4D scene understanding.

While these approaches achieve impressive visual quality, they operate in 2D space and require dense user-provided masks or keyframes, making them less suited for physically plausible driving scene editing. Our method bridges this gap by integrating diffusion-based refinement with 4D scene representations, enabling natural language-driven, spatially grounded, and semantically consistent video edits.

III. METHODS

A. Overall

Figure 2 shows the complete pipeline of LangEditor. We input the real video and one vehicle behavior prompt into the pipeline. The real video provides supervision for the separate Gaussian Splatting reconstruction. We follow [8] to divide the whole scene into dynamic moving vehicles and static background, but offer the Dynamic Illumination-Aware Shadow Modeling to model the shadow when editing the positions of vehicles. The vehicle behavior prompt is used for vehicle selection and trajectory editing. The former selects an editing vehicle by relative position, lanes, order, and appearance. The latter manages the suitable trajectory for the edited vehicle. With new trajectories, vehicles are placed in different positions by the prompt, composing an edit 4D Gaussian Splatting scene. However, due to missing perspectives of real video, blur and incomplete often show in the rendered video. To solve this, we use a post diffusion model to refine rendered-coarse video by under score masks, which are generated by a 4D Gaussian Splatting Scene. Finally, the pipeline produces an edited video.

B. Dynamic Illumination-Aware Shadow Modeling

Shadow modeling is a key factor in ensuring scene realism. However, existing reconstruction methods [8] mainly support static illumination modeling in the temporal dimension. Such an assumption is not sufficient for editing tasks. To address this, we propose a Dynamic Illumination-Aware Shadow Modeling (DIASM) module. This module generates consistent and realistic shadows by modeling the illumination direction at each moment and adjusting the spherical harmonics of Gaussian spheres in the shadow region.

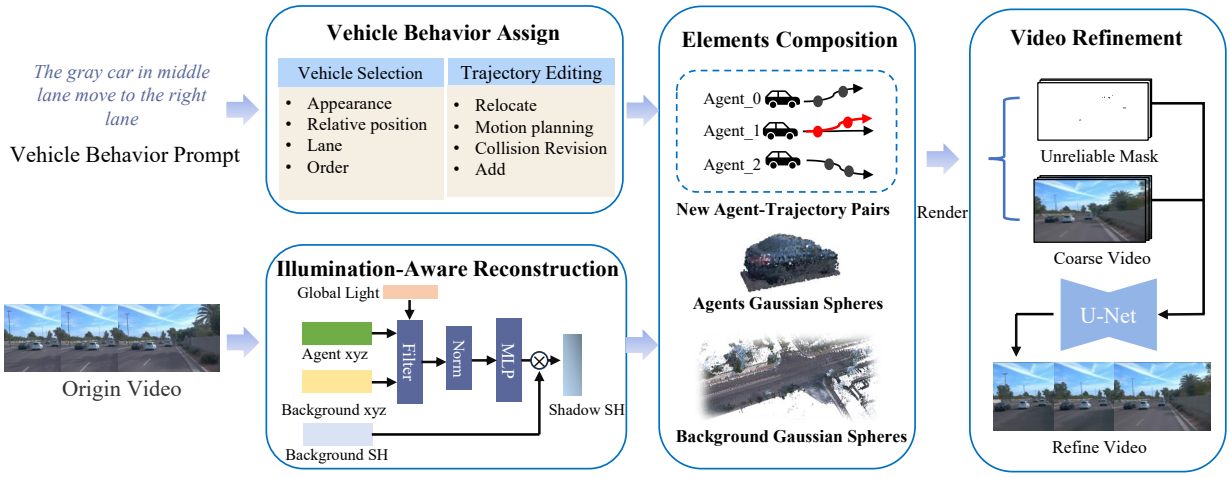

Fig. 2. LangEditor takes a real video and a vehicle behavior prompt as input. The video guides the Gaussian splatting reconstruction, and the prompt is used to select the vehicle and calculate its new trajectory for editing. A post-diffusion model refines the rendered video to address blurring and incompleteness due to missing perspectives.

We define the forward direction of the vehicle as the x-axis, the left direction as the y-axis, and the upward direction as the z-axis. Specifically, we first compute the global illumination direction (d_y, d_z) for each frame from real video [25]. The d_x is set to 0. During the reconstruction training process, the projection position of each point can be calculated as:

$$(x', y', z') = (x, y - \frac{z - g_z}{d_z} d_y, g_z), \tag{1}$$

where (x, y, z) is the coordinate of the point, (x', y', z') is the coordinate of projection point, and g_z is the z-axis height of the ground. For each movable vehicle a with a set of gaussian spheres G_a, we define a 2D projection region R_a as:

$$\begin{aligned} R_a = \{(x, y) | x \in [[X'_a]_{min}, [X'_a]_{max}], \\ y \in [[Y'_a]_{min}, [Y'_a]_{max}]\}, \end{aligned} \tag{2}$$

$$\hat{G_a} = \{g | (x_g, y_g) \in R_a\}, \tag{3}$$

where X'_a is the set of projection position X' of all points in G_a. The gaussian spheres of background which have centers within R_i is considered as the possible shadow gaussian sphere, denoted as $\hat{G_i}$. Then, we define the size of G_a as:

$$S_a = \Delta(X_a, Y_a, Z_a), \tag{4}$$

where Δ represents the difference between the maximum and minimum values of the center coordinates of Gaussian spheres in the corresponding axis, reflecting the size along that axis.

Finally, given a vehicle a and a set of possible shadow Gaussian spheres $\hat{G_a}$, a lightweight MLP network adaptively adjusts the color of $\hat{G_i}$ to learn fine-grained shadow distributions:

$$W = MLP[\frac{(\hat{x_a}, \hat{y_a}, \hat{z_a}) - ([X'_a]_{max}, [Y'_a]_{max}, [Z'_a]_{max})}{S_a}], \tag{5}$$

$$S\hat{H}_a = S\hat{H}_a \times W, \tag{6}$$

where $S\hat{H}_i$ are the spherical harmonics coefficients of $\hat{G_i}$, which presents colors. In editing, we follow the same steps but use the trained MLP weights to adjust the colors of the shadow region gaussian spheres.

C. Appearance Refinement

Due to the limitations of the reconstruction perspective, artifacts and holes may occur after editing. To refine the coarse video, we have designed a post-processing quality enhancement solution based on generative priors. To begin with, we filter the regions that need to be redrawn by rendering the reliability of Gaussian spheres. Then, we use a generative method to fill in the possible holes.

Previous studies [26] have shown that the size of Gaussian primitives is closely correlated with the visual reliability of reconstructed regions, where smaller Gaussians typically correspond to finer details and higher-quality rendering. Building upon this insight, we design an adaptation tailored for autonomous driving scenarios. In dynamic driving scenes, moving vehicles and background regions exhibit significant differences in Gaussian size distributions. Leveraging this property, we introduce a domain-specific selection mechanism that focuses on editing moving vehicles: we select Gaussian spheres associated with moving vehicles adaptively and render them as an unreliable mask, thereby localizing the regions most in need of refinement. The 3-channel confidence

map can be rendered as:

$$C_{conf} = \sum_{i \in M} s_i \alpha_i \prod_{j=i}^{i} (1 - \alpha_j), \tag{7}$$

where M means the Gaussian spheres belong to moving vehicles, s_i is the scale, and α is the opacity. For computational convenience, we define pixel reliability as:

$$P_{conf} = C_{conf}.mean(). \tag{8}$$

The unreliable mask U_{mask} is filtered by a preservation rate r_s:

$$U_{mask} = 1 - P_{conf} > max(P_{conf}) * r_s. \tag{9}$$

Finally, the coarse images are converted into mask images:

$$I_m = I_c * U_{mask}. \tag{10}$$

To improve the fidelity of these unreliable regions, we incorporate a Video Diffusion–based inpainting strategy that has demonstrated strong performance in general video editing tasks. In particular, we adopt the video generation network of ViewCrafter(VC) [27], originally developed for challenging hole-filling tasks. It takes masked videoV_m and coarse video V_c as input, then output the refined video V_r:

$$V_r = VC(V_m, V_c). \tag{11}$$

By combining our domain-specific reliability modeling with the generative capabilities of diffusion models, we are able to produce more consistent and visually plausible reconstructions of moving vehicles under complex driving scenarios.

D. 4D Intuitive Control via Natural Language

The *LangEditor* control system is a multi-agent framework designed for dynamic scene editing in autonomous driving scenarios. It interprets natural language instructions and generates modified 4D scenes that align with user intent.

1) Vehicle Selection via Multi-Modal Localization

Compared to traditional localization tasks, vehicle localization in autonomous driving presents unique challenges and characteristics. Unlike conventional approaches that rely primarily on spatial coordinates and geometric features, vehicle localization in autonomous driving benefits from leveraging a wide range of large-scale spatial and contextual information, such as lane assignments, the relative positioning between vehicles, and their relative order within a traffic scene. This spatial context enriches the localization process, facilitating more accurate and efficient identification of target vehicles.

To address these challenges, we propose an innovative vehicle localization method that combines large language model (LLM) decomposition with modular implementation for enhanced flexibility and scalability. Our approach involves using an LLM to decompose the input instruction into four distinct components: appearance description, lane position description, relative position description, and relative order description. Each of these components provides essential information that informs the localization task.

For the appearance description, we apply a dynamic-static separation strategy during the vehicle reconstruction process. This strategy ensures that each object is rendered individually, enhancing the accuracy of visual representation. Furthermore, we integrate multi-frame CLIP [28] features to build a robust and comprehensive visual representation, effectively capturing the dynamic features of the vehicle and its surroundings:

$$V_a = \frac{1}{len(F_a)} \sum_{i \in F_a} CLIP(I_i), \tag{12}$$

where V_a means the visual dynamic features of vehicle a, F_a is the set of frames that contain vehicle a. I_i is the rendering image of frame i. By comparing the appearance description to the visual features, we compute the similarity between the two and rank candidate vehicles accordingly. Given the appearance description text T_a and visual feature V_a, we rank candidate vehicles by computing the similarity between the appearance description and the visual features. The cosine similarity is used to measure the matching degree between these two:

$$\text{Sim}(T_a, V_a) = \frac{T_a \cdot V_a}{\|T_a\| \|V_a\|}, \tag{13}$$

where $\| \cdot \|$ represents the vector norm.

In parallel, we incorporate additional constraints to refine the ranking process. These constraints are derived from the lane position P_{lane}, relative position P_{rel}, and relative order descriptions P_{order}, which ensure that the spatial context is considered when selecting the target vehicle. This multi-faceted ranking strategy improves the robustness and accuracy of the vehicle identification process, enabling it to account for complex real-world driving scenarios:

$$S(a) = f(\text{Sim}(T_a, V_a)) + P_{\text{lane}}(a) + P_{\text{rel}}(a) + P_{\text{order}}(a), \tag{14}$$

where $S(a)$ is the combined ranking result, $f(\cdot)$ is a scales function. Finally, based on the combined ranking results, we select the vehicle that best matches the input instruction, ensuring that the identified vehicle is the one intended for editing. This method not only facilitates accurate vehicle localization but also supports downstream tasks, such as targeted occlusion during rendering. Once the vehicle is identified, it can be easily occluded in subsequent frames, providing a powerful tool for realistic scene editing in autonomous driving applications. This module represents a significant step forward in the field of vehicle localization for autonomous driving, offering a more reliable and context-aware solution for complex driving scenarios.

2) Trajectory Editing via Specialized Agents

Directly applying video generation models can easily lead to spatial and temporal inconsistencies among dynamic agents in the given scene. To address this issue, we employ a dynamic-static decoupled editing approach based on gaussian scene representation: First, leveraging the vehicle selection module to identify the specific vehicle to be replanned. Second, we propose a novel text-to-motion method with a hierarchical multi-agent workflow, generating per-frame trajectory waypoints in world coordinates for the specific vehicle. Finally, during the reconstruction phase, the position

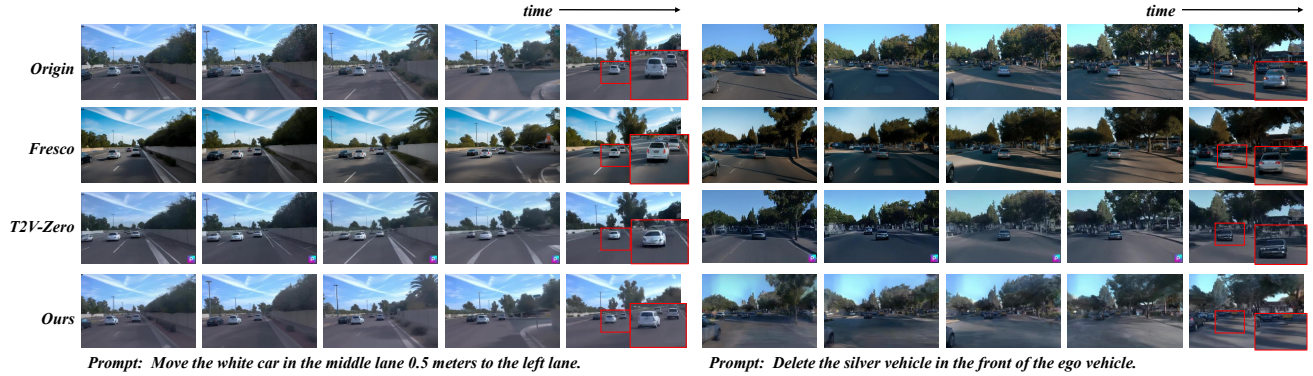

Fig. 3. Qualitative comparison of our proposed method with T2V-Zero [20] and Fresco [22]. The results show that our method preserves the original visual style and fine-grained details while demonstrating superior fidelity to the intended prompts.

TABLE I
RESULTS OF VISUAL QUALITY EVALUATION

Method	FID	CLIP-I	FVD
Reconstruction	70.92	89.90	788.60
T2V-Zero	112.31	73.15	1619.77
Fresco	96.81	83.42	760.04
Ours	**74.95**	**87.47**	**752.28**

TABLE II
EFFECTIVENESS OF VEHICLE SELECTION MODULE

Method	ALL	Only App.	App. and Pos.
VisProg	34.48	63.64	16.67
Florence-SAM	58.62	63.64	55.55
Ours	**86.21**	**90.91**	**83.33**

and orientation of the vehicle are updated frame by frame according to the newly planned trajectory coordinates. In the second step mentioned above, we find that directly employing a single LLM agent struggles with multi-step reasoning and cross-referencing. To overcome this limitation, we designed the hierarchical multi-agent workflow mentioned above. In this framework, each agent is specialized in operational editing roles:

Manager Agent. At the top level of the hierarchical workflow, the Manager Agent serves as the coordinator of the entire multi-agent system. It directly processes the user's raw text prompt, performs intent segmentation based on user instructions, and determines which agents to use and when to invoke or dispatch messages.

Relocation Agent. The Relocation Agent translates natural language relocation commands (e.g., "shift the car left by 5m") into quantifiable spatial offsets along the x and y axes of the ego-vehicle coordinates. The output can be used for editing tasks that shift the position of the vehicle by uniformly applying the transformation to all trajectory points of the target vehicle during rendering.

Motion Planning Agent. This agent generates physically plausible trajectories based on user-specified motion descriptions. It receives a segmented command from the Manager Agent and then structurally parses it into two components: action (including straight, stop, lane change, turn) and speed profile (including default, slow, fast). The parsed command is then forwarded to a planning module that operates in the BEV map. A key advantage of planning in this 2D domain is the ability to efficiently conduct collision detection and trajectory refinement. The planning process incorporates kinematic constraints and scene context in WOD [2] to ensure plausible trajectory generation.

Collision Revision Agent. We further implement a dynamic collision detection and correction module. Upon completion of the Motion Planning Agent, the Manager Agent invokes the Collision Revision Agent to ensure plausibility and safety of the trajectory. It executes two distinct strategies: for frontal collisions of the given vehicle, it inserts "safety interpolation frames" before the collision frame to reduce speed; for rear collisions, it takes the scene before collision as input, increases the speed, and re-executes the planning module to generate a new trajectory.

Adding Agent. This agent places new vehicles according to user-defined attributes and integrates them into the scene with compliant trajectories. Specifically, messages transmitted by the Manager Agent are first parsed into structured data: (a) Reference Vehicle: The vehicle relative to which the new vehicle is to be placed. (b) Relative Offset: The desired position relative to the reference vehicle along the x and y axes. Based on this information, the precise placement coordinates are computed. The Manager Agent then invokes the Motion Planning module to generate a complete and compliant trajectory for the newly added vehicle.

IV. EXPERIMENTS

A. Experimental Setups

Our experiments are conducted on the Waymo dataset, where we select 23 editing scenarios and collect 23 videos, each consisting of 100 frames. The resolution for reconstruction and coarse editing is set to 1600×1024, while the final

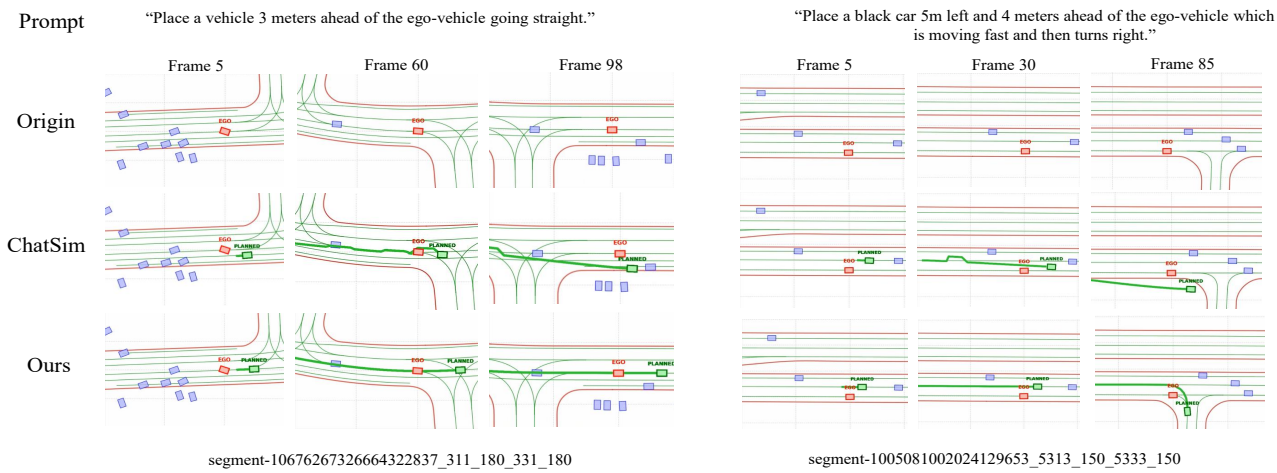

Fig. 4. Qualitative comparison of our proposed method with ChatSim [29]. Segments are selected from the training set of Waymo Open Dataset [2]. Both experiments are adding and planning tasks. We use a visualization script to produce bird's-eye-view (BEV) trajectories from per-frame trajectory points generated by two methods.

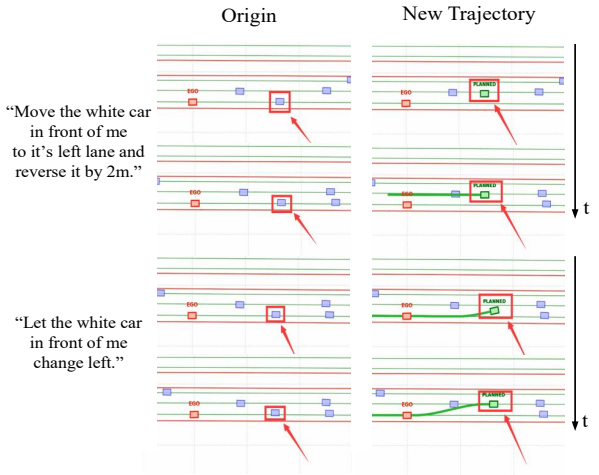

Fig. 5. Qualitative analysis of moving and turning editing results by visualization script in BEV map.

refinement is performed at 900×576. The editing instructions include object addition, deletion, and trajectory modification. For evaluation, we use the front-view camera data from the Waymo dataset, although our method is capable of editing across multiple viewpoints. All experiments are carried out on a single NVIDIA RTX 3090 GPU. For all LLMs used, we use Qwen3 [30], the single-language modality version.

B. Main Results

Visual Quality. Since open-source text-driven editing methods tailored for autonomous driving scenarios are extremely limited, we benchmark the editing quality against two representative approaches. We select T2V-Zero [20] as a classical text-to-video editing baseline and Fresco [22] as a method that demonstrates superior performance in motion control. Besides, we compare the reconstruction results. For quantitative evaluation of visual quality, we adopt FID [31] and CLIP-I [28] for visual quality assessment [32]–[34], and FVD [35] for video fidelity. The results are shown in Table I. Compared to other methods, our method has the lowest FID, FVD and the highest CLIP-I, indicating better visual quality. We also show the reconstruction results in line 1. Besides, some cases are shown in Figure 3. Our method effectively preserves the original visual style and fine-grained details, while demonstrating higher fidelity to the intended prompts.

Vehicle Selection Quality. To validate our approach, we evaluate the success rate (%) of three methods: Visual Programming [36], Florence-SAM, and our proposed vehicle selection module. Visual Programming leverages an LLM to interpret prompts and provides a SELECT function to ground the vehicle. Florence-SAM combines Florence2 [37] and SAM2 [38]: the former handles open-vocabulary detection and vision-language alignment, while the latter performs zero-shot segmentation. For Visual Programming, we input each video frame individually and determine the video-level result by majority voting. For Florence-SAM, we adopt the video-based variant of the model. Experiments are conducted on 5 scenarios with 29 prompts. As shown in Table II, our method consistently outperforms the baselines. The advantage is most evident when prompts include both appearance and positional descriptions, since our module is tailored for driving-scene understanding. Even prompts contain only appearance descriptions, our method still achieves better performance than the other two.

Trajectory Editing Quality. In our qualitative experiments, we compare the trajectories of adding and planning tasks generated by our method, the baseline ChatSim [29], and the original driving videos, as illustrated in Figure 4. Both methods start from an initial position that complies with the linguistic command. ChatSim [29] uses a limited static map (100m ahead, 40m laterally) and linear interpolation, failing to respect road geometry. For example, under a "go straight" instruction, the vehicle drives off the road,

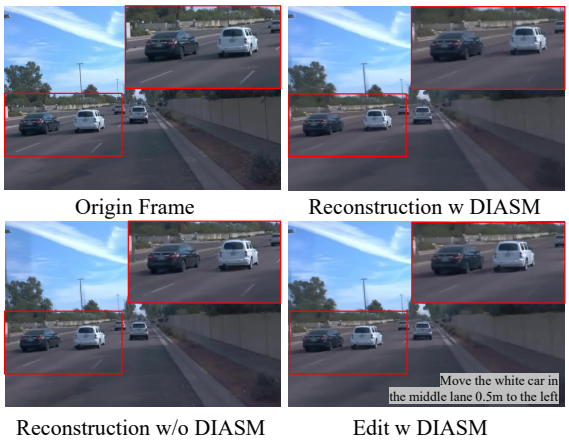

Fig. 6. Qualitative analysis of the results of the DIASM module.

TABLE III
ABLATION STUDY OF THE DIASM MODULE

Reconstruction			
	PSNR	SSIM	LPIPS
w/o DIASM	23.0964	0.8292	0.2998
w DIASM	**23.7033**	**0.8336**	**0.2955**
Edit			
	FID	CLIP-I	FVD
w/o DIASM	73.76	89.24	803.60
w DIASM	**73.21**	**89.26**	**790.98**

and under a "turn right" instruction, it turns too early. Its collision revision module also causes unnatural motion patterns, as seen in the trajectories in Frames 60 (left) and 30 (right) in Figure 4. In contrast, our method interprets "go straight" as driving along the lane and executes a turn until reaching an intersection. The collision detection and correction in our approach yield smoother and more natural motion adjustments. We have also achieved trajectory editing for changing lanes, which has not been implemented in ChatSim [29], enabling the generation of corner cases such as "a vehicle quickly cutting in from the right lane". Some other results are shown in Figure 5. The results highlight that our method achieves superior performance, with more accurate and realistic trajectory adjustments. The edited trajectories proficiently preserve spatiotemporal consistency and scene semantics, demonstrating our framework's high-quality dynamic scene manipulation.

C. Ablation Study

Ablation on DIASM. We compare the performance of reconstruction and editing with and without our DIASM module. For reconstruction, we use metrics including PSNR, LPIPS, and SSIM. For editing, we use FID, CLIP-I, and FVD score. The results are shown in Table III. With this module, the PSNR, SSIM, and LPIPS increase, indicating better reconstruction quality. Besides, the quality of editing videos also improves. We also conducted a qualitative analysis of the results of the DIASM module, as shown in Figure 6. Compared to the results without the DIASM module, the shadow modeling with DIASM is much closer to the ground truth. Additionally, after editing the objects within the scene (as shown in the bottom-right corner), the light and shadow were also reasonably modeled.

TABLE IV
ABLATION STUDY OF THE APPEARANCE REFINEMENT MODULE

	FID	CLIP-I	FVD
w/o refine	75.27	87.19	826.52
w refine	**74.95**	**87.47**	**752.28**

TABLE V
ABLATION ON VEHICLE SELECTION MODULE.

App.	Pos.	Success rate
✓		33.33
	✓	55.55
✓	✓	**83.33**

Ablation on Appearance Refinement. We present the effectiveness of the Appearance Refinement module through an ablation study in Table IV. After incorporating the Appearance Refinement module, all visual quality metrics show improvement, especially in FVD. It indicates that the module enhances the alignment between the edited video and the reference video, leading to more realistic and visually consistent outputs. These results demonstrate that the Appearance Refinement module significantly improves both the perceptual and objective quality of the generated scenes.

Ablation on Vehicle Selection Module. We evaluate the success rate (%) of our method when it considers only appearance, only position, and both. The prompts include both appearance and position descriptions. When only one element is considered, the success rate drops significantly. Especially when only considering appearance descriptions, it is because there are many vehicles with similar appearances within the same road scene. It highlights the importance of combining both aspects for optimal performance.

V. CONCLUSIONS

We introduce LangEditor, a natural language-driven 4D scene editing framework for dynamic driving environments. LangEditor enables intuitive control of scene vehicles and ensures spatiotemporal consistency and visual fidelity through innovative modules like Spatial Illumination Adaptation and Appearance Refinement. Unlike previous methods that are limited to 2D or 2.5D editing, LangEditor provides a fully 4D approach, allowing for realistic and physically plausible scene modifications. Extensive experiments demonstrate its superior performance in editing quality and user controllability. Our approach bridges the gap between realistic driving scene generation and user-driven editing, offering a scalable solution for creating diverse and high-quality scenarios for

autonomous driving training.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] 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.

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[4] Y. Liang, Z. Yan, L. Chen, J. Zhou, L. Yan, S. Zhong, and X. Zou, "Driveeditor: A unified 3d information-guided framework for controllable object editing in driving scenes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 5, 2025, pp. 5164–5172.

[5] R. Gao, K. Chen, E. Xie, L. Hong, Z. Li, D.-Y. Yeung, and Q. Xu, "Magicdrive: Street view generation with diverse 3d geometry control," *arXiv preprint arXiv:2310.02601*, 2023.

[6] R. Gao, K. Chen, Z. Li, L. Hong, Z. Li, and Q. Xu, "Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes," *arXiv preprint arXiv:2405.14475*, 2024.

[7] Y. Wei, Z. Wang, Y. Lu, C. Xu, C. Liu, H. Zhao, S. Chen, and Y. Wang, "Editable scene simulation for autonomous driving via collaborative llm-agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 077–15 087.

[8] 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.

[9] N. Huang, X. Wei, W. Zheng, P. An, M. Lu, W. Zhan, M. Tomizuka, K. Keutzer, and S. Zhang, "S3 gaussian: Self-supervised street gaussians for autonomous driving," *arXiv preprint arXiv:2405.20323*, 2024.

[10] 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.

[11] 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.

[12] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[13] S. Lu, Z. Lin, C. Lu, H. Wang, G. Zhuo, and L. Zheng, "Multieditor: Controllable multimodal object editing for driving scenarios using 3d gaussian splatting priors," *arXiv preprint arXiv:2507.21872*, 2025.

[14] Z. Zhu, Y. Zou, C. M. Jiang, B. Sun, V. Casser, X. Huang, J. Wang, Z. Yang, R. Gao, L. Guibas *et al.*, "Scenecrafter: Controllable multi-view driving scene editing," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6812–6822.

[15] B. Singh, V. Kulharia, L. Yang, A. Ravichandran, A. Tyagi, and A. Shrivastava, "Genmm: Geometrically and temporally consistent multimodal data generation for video and lidar," *arXiv preprint arXiv:2406.10722*, 2024.

[16] H. Lu, T. Xu, W. Zheng, Y. Zhang, W. Zhan, D. Du, M. Tomizuka, K. Keutzer, and Y. Chen, "Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving," *arXiv preprint arXiv:2412.09043*, 2024.

[17] 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," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6782–6791.

[18] M. Khan, H. Fazlali, D. Sharma, T. Cao, D. Bai, Y. Ren, and B. Liu, "Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 8315–8321.

[19] R. Song, C. Liang, Y. Xia, W. Zimmer, H. Cao, H. Caesar, A. Festag, and A. Knoll, "Coda-4dgs: Dynamic gaussian splatting with context and deformation awareness for autonomous driving," *arXiv preprint arXiv:2503.06744*, 2025.

[20] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan, and H. Shi, "Text2video-zero: Text-to-image diffusion models are zero-shot video generators," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 954–15 964.

[21] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 7623–7633.

[22] S. Yang, Y. Zhou, Z. Liu, , and C. C. Loy, "Fresco: Spatial-temporal correspondence for zero-shot video translation," in *CVPR*, 2024.

[23] S. Liu, Y. Zhang, W. Li, Z. Lin, and J. Jia, "Video-p2p: Video editing with cross-attention control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8599–8608.

[24] C. Qi, X. Cun, Y. Zhang, C. Lei, X. Wang, Y. Shan, and Q. Chen, "Fatezero: Fusing attentions for zero-shot text-based video editing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 932–15 942.

[25] J. Ge, Z. Liu, L. Fan, Y. Jiang, J. Su, Y. Li, Z. Zhang, and S. Chen, "Unraveling the effects of synthetic data on end-to-end autonomous driving," *arXiv preprint arXiv:2503.18108*, 2025.

[26] X. Liu, C. Zhou, and S. Huang, "3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors," *Advances in Neural Information Processing Systems*, vol. 37, pp. 133 305–133 327, 2024.

[27] W. Yu, J. Xing, L. Yuan, W. Hu, X. Li, Z. Huang, X. Gao, T.-T. Wong, Y. Shan, and Y. Tian, "Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis," *arXiv preprint arXiv:2409.02048*, 2024.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[29] Y. Wei, Z. Wang, Y. Lu, C. Xu, C. Liu, H. Zhao, S. Chen, and Y. Wang, "Editable scene simulation for autonomous driving via collaborative llm-agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 077–15 087.

[30] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[32] S. Sun, X. Liang, S. Fan, W. Gao, and W. Gao, "Ve-bench: Subjective-aligned benchmark suite for text-driven video editing quality assessment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 7, 2025, pp. 7105–7113.

[33] S. Sun, X. Liang, B. Qu, and W. Gao, "Content-rich aigc video quality assessment via intricate text alignment and motion-aware consistency," *arXiv preprint arXiv:2502.04076*, 2025.

[34] S. Sun, B. Qu, X. Liang, S. Fan, and W. Gao, "Ie-bench: Advancing the measurement of text-driven image editing for human perception alignment," *arXiv preprint arXiv:2501.09927*, 2025.

[35] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[36] T. Gupta and A. Kembhavi, "Visual programming: Compositional visual reasoning without training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14 953–14 962.

[37] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan, "Florence-2: Advancing a unified representation for a variety of vision tasks," *arXiv preprint arXiv:2311.06242*, 2023.

[38] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.