SceneCrafter: Controllable Multi-View Driving Scene Editing

Zehao Zhu[1, 2], Yuliang Zou[1], Chiyu Max Jiang[1], Bo Sun[1], Vincent Casser[1], Xiukun Huang[1]
Jiahao Wang[3], Zhenpei Yang[1], Ruiqi Gao[4], Leonidas Guibas[4], Mingxing Tan[1], Dragomir Anguelov[1]
[1]Waymo, [2]University of Texas at Austin, [3]Johns Hopkins University, [4]Google DeepMind

Figure 1. SceneCrafter is a versatile and dexterous editor for realistic 3D-consistent manipulation of driving scenes captured from multiple camera angles. It allows users to seamlessly insert or remove arbitrary objects in the foreground (second row) and modify global features like weather (third row) and time of day (fourth row), while preserving fine-grained details of scene layout and geometry.

Abstract

Simulation is crucial for developing and evaluating autonomous vehicle (AV) systems. Recent literature builds on a new generation of generative models to synthesize highly realistic images for full-stack simulation. However, purely synthetically generated scenes are not grounded in reality and have difficulty in inspiring confidence in the relevance of its outcomes. Editing models, on the other hand, leverage source scenes from real driving logs, and enable the simulation of different traffic layouts, behaviors, and operating conditions such as weather and time of day. While image editing is an established topic in computer vision, it presents fresh sets of challenges in driving simulation: (1) the need for cross-camera 3D consistency, (2) learning "empty street" priors from driving data with foreground occlusions, and (3) obtaining paired image tuples of varied editing conditions while preserving consistent layout and geometry. To address these challenges, we propose SceneCrafter, a versatile editor for realistic 3D-consistent manipulation of driving scenes captured from multiple cameras. We build on recent advancements in multi-view diffusion models, using a fully controllable framework that scales seamlessly to multi-modality conditions like weather, time of day, agent boxes and high-definition maps. To generate paired data for supervising the editing model, we propose a novel framework on top of Prompt-to-Prompt [15] to generate geometrically consistent synthetic paired data with global edits. We also introduce an alpha-blending framework to synthesize data with local edits, leveraging a model trained on empty street priors through novel masked training and multi-view repaint paradigm. SceneCrafter demonstrates powerful editing capabilities and achieves state-of-the-art realism, controllability, 3D consistency, and scene editing quality compared to existing baselines.

1. Introduction

Simulation is a core component in autonomous vehicle (AV) development for enabling system-level evaluation and quality hillclimbing at lower cost and faster turnaround times than real-world testing. Critical use-cases are evaluating the performance of the AV in new traffic scenarios (different agent location and behavior) as well as new operating conditions (such as different weather).

Most existing works in the driving simulation literature reduce simulation to an agent behavior simulation problem [7, 13, 38] so as to enable closed-loop simulation of motion planners. This formulation bypasses the perception system, which limits their applicability and can cause simulations to be untrustworthy under behavioral changes due to pose divergence. These methods are unable to simulate changing operating conditions challenging the perception system, nor can they support evaluation of end-to-end planners [22, 23] requiring realistic sensor inputs.

Sensor simulation approaches have benefited from recent advances in reconstructive modeling through neural fields/primitives [27, 37] or image/video generation models [45, 46]. While reconstructive techniques are faithful to real scenes, they lack the flexibility of efficiently simulating varied lighting and weather conditions, or manipulating existing objects. Generative sensor simulation methods are typically formulated as an image or video generation task conditioned on attributes such as text descriptions or scene layout. However, such scenes are usually not grounded in real scenes and do not inspire confidence in their actual occurrence under real driving conditions. Editing models, on the other hand, offer the best combination of realism-grounding as well as the ability to rely upon large data priors for flexible edits.

While there is a large body of work in generic image editing tasks, there is limited work, to our knowledge, in applying this in the context of multi-view driving imagery. In particular, to facilitate the two aforementioned sensor simulation tasks of evaluating under new traffic scenarios and operating conditions, we require two types of editing modalities: local foreground editing such as agent removal and injection conditioned on agent location, size and type, as well as global editing pertaining different operating conditions, such as varying weather conditions and time of day. To train the final editing model, we require paired image tuples before and after the edit to supervise the final editing model. Obtaining paired supervision data for local and global editing poses different sets of challenges.

Local editing requires pairs of "empty streets" (with no agents) and "populated streets". However as we show in Sec. 4.4, directly applying inpainting techniques such as RePaint [33] to erase agents does not result in high quality results due to the model not having good priors of "empty streets" as almost all training data contains "populated streets". We develop a novel training paradigm that we coin *masked training* that enables us to learn to priors of empty streets from datasets of populated streets, resulting in high quality "empty street" priors. It enables us to generate high quality pairs of populated and unpopulated scenes, from which we can selectively curate pairs of partially populated scenes via alpha blending to train the editing model.

Global editing poses a different set of challenges. While seminal work such as Prompt-to-Prompt [15] and InstructPix2Pix [5] offer an exciting new avenue for creating paired synthetic images, directly applying Prompt-to-Prompt to the multi-camera setting does not perform well due to the absence of text conditions whose attention-weights are frozen during Prompt-to-Prompt inference. Through this work, we come to the interesting finding that instead of freezing the image-to-condition cross-attention weights, freezing image-to-image self-attention weights resulted in much more consistent geometry across synthetic pairs in pixel-level. Furthermore, inclusion of more scene conditions such as agent 3D locations and high-definition (HD) maps helps to improve the geometric consistency across paired synthetic images.

Finally, we train a unified editing model: SceneCrafter, to jointly learn the aforementioned editing tasks. We propose novel 3D consistency metrics for multi-view image generation tasks, demonstrating significant realism, controllability, editing quality, and 3D consistency improvement on top of available baselines.

In summary, our key contributions are as follows:

- We present SceneCrafter, a multi-view driving scene editing model supporting local foreground object removal and injection, as well as global editing of operating conditions such as weather and time of day (See Fig. 1).
- We propose a masked training paradigm and a multi-view repaint algorithm to remove agents from input images, along with an alpha-blending method for generating synthetic data, enabling training an editing model for flexible object insertion, removal, and replacement operations.
- Our novel approach extends Prompt-to-Prompt [15] for synthetic data generation in driving scenes. By integrating a modified attention-weight replacing mechanism and conditioning on scene layouts, we achieve significant improvements in both the realism and geometric consistency of generated data pairs across diverse operating conditions.
- We propose *3D LPIPS*, a metric to measure multi-view image consistency. Our results demonstrate a marked increase in realism, controllability, 3D consistency, and editing quality over established baselines.

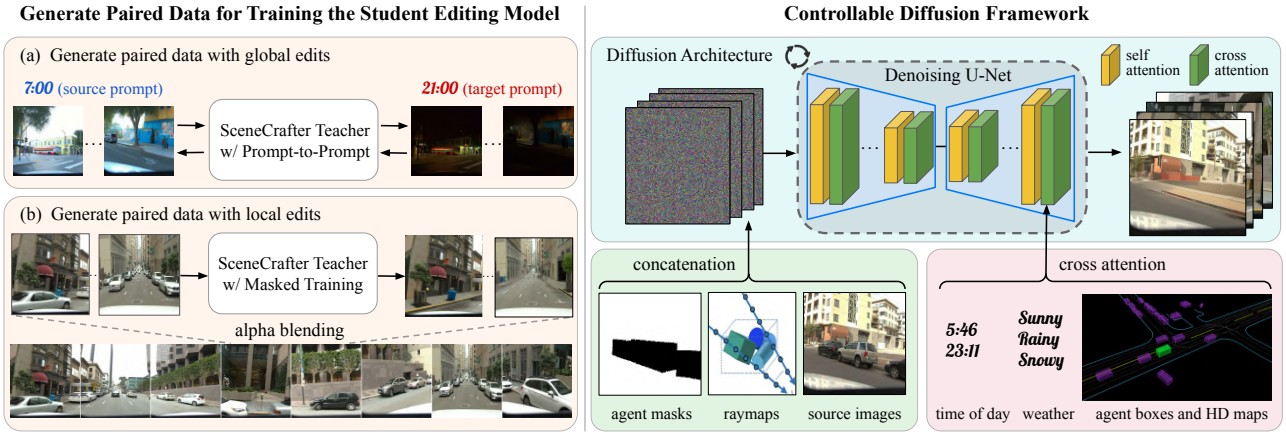

Figure 2. **Overview.** Our method consist of two main stages. First, we train two teacher models to synthesize a large-scale paired dataset with several novel ideas (Sec. 3.3). We then train a unified student model with the generated data for 3D-consistent scene editing (Sec. 3.4).

2. Related Work

Diffusion models. Diffusion models [45, 46] have shown promising generation results in various domains, such as image [4, 39, 41, 44], video [6, 14, 19], text [30], and audio [28]. Latent diffusion models (LDMs) [43] in particular mark a major milestone. By learning the diffusion models in latent instead of pixel space, LDM greatly reduce the learning difficulty in the high-dimensional pixel space, and thus enable easier training and generation at higher resolutions. In this work, we build our controllable multi-view scene editing model on top of the state-of-the-art LDM-based novel view synthesis model [10].

Image editing. Image editing is an important task in computer vision and graphics research. In general, there are two ways to perform generative image editing: training-free and training-based. Training-free approaches aim to leverage image priors from well-trained generative models without training a specific editing model. To utilize the StyleGAN priors [25, 26], previous approaches invert [1–3] or encode [8, 42, 47] the input images into the StyleGAN latent space and then perform editing by manipulating the latent vectors. For diffusion models, Prompt-to-Prompt [15] can edit generated images by updating the input text prompt and manipulating the cross attention weights so that the edit can be grounded to a specified region. However, Prompt-to-Prompt mainly focuses on editing images generated by the text-conditioned diffusion model, instead of user-specified input images. To edit real imagery, SDEdit [36] uses a pre-trained diffusion model to add noise and then denoise the input images, but it struggles to maintain the fine-grained geometry of the edited results. On the other hand, we can train specialist feed forward models to directly conduct the editing tasks. In this work, we propose to generate training data with ideas from training-free approaches, and then train a unified editing model with synthetic data.

Synthetic training data generation. Training deep neural networks usually requires large amounts of data. However, collecting high-quality data at scale is not a trivial task and usually involves a significant amount of human efforts. As generative models are evolving in visual fidelity, more and more research works [29, 52, 53, 57] have been exploring training deep models with these generated synthetic data. Our method is inspired by the pioneering work InstructPix2Pix [5], which leverages GPT3 [35] and Prompt-to-Prompt to generate the editing prompts and training image pairs. We extend Prompt-to-Prompt for multi-view image editing, replacing text prompts with driving-specific signals for precise control. We also leverage Repaint algorithm to generate local editing data. Unlike DriveEditor [32], which masks objects randomly for faster processing, our method removes identified vehicles, ensuring structured and realistic results.

Simulation for autonomous driving. Historically, most simulation works focus on driving policy simulation and scenario generation [7, 13, 38], which are then used to evaluate and improve performance of motion planners. However, this formulation fails to evaluate the whole AV system in a closed-loop setting. Given the recent trend of end-to-end driving models [12, 21–23], how to realistically generate sensor data for full system closed-loop simulation becomes a critical problem. Recently generative world models [9, 11, 20, 31, 34, 48–50, 54, 55, 58] are capable of generating photo-realistic future frames, conditioned on language or action prompts. Panacea [51] can even generate multi-view videos given vehicle actions, agent locations, and the environment map. However, these methods mainly focus on generating the future or a completely new scene, but not editing real driving footage. In this paper, we aim to fill this gap by proposing a versatile editor for realistic 3D consistent driving scene editing.

3. Method

In this section, we first provide a preliminary review of the multi-view diffusion model in Sec. 3.1. We then provide details on the architecture in Sec. 3.2 and explain the process of creating synthetic datasets using the teacher model in Sec. 3.3. Specifically, we train two separate teachers models to generate data for global edits and local edits, respectively. Lastly, in Sec. 3.4 we describe how to train a unified student editing model with the generated paired data. An overview of our pipeline for synthetic dataset generation and the controllable diffusion framework is illustrated in Fig. 2.

3.1. Preliminary: Multi-View Diffusion Models

Our method is based on multi-view diffusion models [10], which extends the latent diffusion models (LDM) [43] by enabling view consistency and camera pose control over the image generation. The goal of this architecture is to estimate the joint distribution of multi-view images under given camera poses, and generate photo-realistic and view-consistent images. Specifically, given a set of N camera poses $\mathbf{p} = p_{0:N}$ where $N = 8$ in our setting, the model learns to estimate the distribution of N images $\mathcal{I} = I_{0:N}$ under the corresponding camera poses: $P(\mathcal{I}|\mathbf{p})$.

Following [43], the base architecture is a latent diffusion model with an encoder $\mathcal{E}$, a denoiser U-Net ϵ_θ and a decoder $\mathcal{D}$. The images are encoded into latents $\mathbf{z} = z_{0:N}$ and then denoised in latent space: $\mathcal{I} = \mathcal{D}(\mathbf{z}), \mathbf{z} = \mathcal{E}(\mathcal{I})$.

Two key architecture changes are adopted to achieve the aforementioned goal: a view-spatial joint attention module and camera pose conditioning generation.

View-spatial joint attention. The 2D attention blocks in the LDM are replaced with 3D attention blocks (2D in space and 1D cross views) to perform the attention mechanism among multi-view images. To reuse the parameters from the original LDM, the 2D attention module is directly inflated to 3D without introducing extra parameters. All other 2D blocks are applied to each image separately.

Camera poses conditioning. The camera poses are represented via raymaps, which encode the ray origin and direction at each spatial location. All camera poses are normalized w.r.t. the first camera so the raymaps are invariant to global rigid transformations. The diffusion U-Net takes the the concatenated noisy latents and raymaps as input, and outputs the denoised latents.

3.2. Teacher Model for Scene Generation

In this section, we introduce how we accommodate various conditional modalities in the teacher model. The model estimates the multi-view image distribution given groups of conditions as $p(\mathcal{I}|\mathbf{p}, \mathbf{m}, \mathbf{c}^g, \mathbf{c}^l)$.

Specifically, the diffusion model takes four types of conditions: global conditions $\mathbf{c}^g$ (weather and time of day), local conditions $\mathbf{c}^l$ (HD map and agent boxes), foreground masks $\mathbf{m}$ and raymaps $\mathbf{p}$. The global conditions are essential for simulation under novel operating conditions, and the latter two help the model capture the scene layout and geometry. We encode all the conditions and integrate them into the cross-attention blocks of the diffusion U-Net.

Weather. Based on the driving log data, we use the CLIP [40] text encoder to encode the text "sunny", "rainy", "foggy", or "snowy" as the weather condition c_w.

Time of day. Given the local time of day and the geographic location of the recorded driving logs, we compute the sun angles for each frame, which is then encoded using positional encoding and used as time of day condition c_t.

High-definition map. We also utilize a high-definition (HD) environment map and process it to the local HD map conditions c_r. The map is represented by lane segments and lane types. We sample up to 4,096 lane segments, feed the segment start and end locations, and lane types to a PerceiverIO [24] to reduce the token size to 512, and use a MLP to encode the features into the condition c_r.

Agent boxes. We use the AV perception system's 3D object detector to gather up to 256 agents in the scene, including both foreground and background objects. Each agent is represented by an 8-dimensional feature tensor comprising the center coordinates (x, y, z), dimensions (length, width, height), heading angle (yaw), and agent type. We apply an one-hot encoding of agent type and concatenate it with the remaining features. We then feed the concatenated features to a MLP to get output c_b as the agent condition.

Foreground mask. We project all 3D bounding boxes of foreground objects into each camera view to generate binary masks. We then resize them and append them to the latents along the channel dimension similar to raymaps.

We apply a 10% dropout rate to each conditions during training. So our model does not rely on these conditions and works robustly without them.

3.3. Synthetic Data Generation with Teacher Model

3.3.1 Generating Data for Global Edits

To generate paired data for global edits (time of day and weather), an intuitive idea is to feed the initial and edited conditions into a well-trained conditional generation model to generate the **source** and **target** images respectively. However, this simple strategy fails to generate geometrically consistent source and target image pairs, as shown in the first and second rows of Fig. 5. This is because generative models inherently lack guarantees for image coherence, even when only minor changes are made to the conditioning prompt [5]. To address this issue, we employ Prompt-to-Prompt [15], a method designed to maintain similarity across multiple outputs of a conditional diffusion model. However, the original Prompt-to-Prompt approach falls short in terms of controllability and realism

Figure 3. **Qualitative Results on Global Editing.** Given multi-view image inputs, our model performs versatile edits like changing the time of day (daytime to dawn/night) and weather (sunny to snowy/foggy) while preserving geometric consistency. *Best viewed zoomed in.*

for our multi-view driving scene setting. Thus, we make the following adaptation to Prompt-to-Prompt.

Replacing self-attention weights. Different from the original Prompt-to-Prompt approach that manipulates image-to-condition cross-attention weights, we extend this to image-to-image self-attention layers to better handle global conditions. While Prompt-to-Prompt uses text tokens and substitutes semantically meaningful text tokens, our setting handles various multi-modal tokens and replaces certain tokens associated with global effects, while keeping other conditions unaffected. The editing should affect all parts of the image, but still retain the original layout, such as the location and underlying appearance of the vehicles. Therefore, we manipulate pixel-level attention weights in all self-attention layers instead of the cross-attention layers.

Conditioning on more control signals. We observed that incorporating more control signals into our model enables generating more realistic paired results with consistent geometry. With more control signals, the model better preserves fine-grained details which ground the geometry specifically while primarily modifying high-level attributes such as weather. So we incorporate more local conditions into our teacher model, like agent boxes and HD maps.

Choice of source images. The time of day choice for source images plays a crucial role in our method, as it impacts the overall style of the generated results. We empirically found that using only daytime for source images produces superior results compared to using nighttime or using both. Thus we sample the time of day for source images from daytime only and sample target time from all times across a day. We randomly flip the order of source and target images as the paired synthetic data.

3.3.2 Generating Data for Local Edits

In local edits, we aim to remove or insert arbitrary agents into the scene. Our log data is populated with vehicles, but if we can extract "empty streets" data from it, we can then perform alpha blending for the pairs, which enables generating synthetic data with arbitrary numbers of agents. Inpainting methods like RePaint [33] have been widely applied to erase objects in 2D images, however it faces two challenges in our settings: It can not produce multi-view consistent results; and our "populated streets" training data makes it difficult to ensure inpainted results are free of agents when directly training a diffusion model. Thus, we propose a novel framework with two key ideas: *masked training* and *multi-view repaint* to tackle these issues. Masked training enables the model to learn priors of empty streets in a self-supervised manner and multi-view repaint ensures view-consistent results.

Masked training. We propose a simple yet effective training strategy that enables our model to learn the "empty street" prior. Specifically, given a set of image foreground mask for each camera views, we resize them to the same shape with latents, denoted as $\mathbf{m} = m_{0:N}$. We denote foreground latents as $\mathbf{m} \odot \mathbf{z}$ and background latents as $(1 - \mathbf{m}) \odot \mathbf{z}$.

We apply different noise levels when computing $\mathbf{z}_t$. We maintain a noise level of zero for the foreground and apply regular noise to background, as:

$$\mathbf{z}_t = (1 - \mathbf{m}) \odot (\alpha_t \mathbf{z}_0 + \sigma_t \epsilon) + \mathbf{m} \odot \mathbf{z}_0 \quad (1)$$

where α_t and σ_t are predefined noise scheduling terms. Consequently the model focuses on learning to denoise the background while leaving the foreground unchanged.

We only compute training loss on background pixels, and minimize the following training objective,

$$\mathbb{E}_{z\sim\mathcal{E}(I),t,\epsilon\sim\mathcal{N}(0,1)} \left\| (1 - \mathbf{m}) \odot (\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c}^g, \mathbf{c}^l, \mathbf{m})) \right\|_2^2 \quad (2)$$

The model ϵ_θ is trained with foreground-free priors and thus can generate scenes of "empty streets".

Multi-view repaint. We propose a 3D-aware approach to get view-consistent inpainting examples, replacing foreground pixels with backgrounds. Note that we only conduct multi-view repaint with the teacher model with masked training. Specifically, we split the images into masked and unmasked regions representing foreground and background, respectively. In each reverse step, we modify the foreground region $\mathbf{m} \odot \mathbf{z}_t$ while preserving the correct properties of the corresponding distribution. Background region $(1 - \mathbf{m}) \odot \mathbf{z}_t$ is sampled at any time step t given known latents $\mathbf{z}_0$. Thus, we have:

$$\mathbf{z}_{t-1}^{\text{background}} \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1 - \bar{\alpha}_t)\mathbf{I}\right)$$

$$\mathrm{z}_{t-1}^{\text{foreground}} \sim \mathcal{N}\left(\mu_\theta(\mathbf{z}_t, t, \mathbf{c}^g, \mathbf{c}^l), \Sigma_\theta(\mathbf{z}_t, t, \mathbf{c}^g, \mathbf{c}^l)\right) \quad (3)$$

Methods	Time of day editing			Weather editing		
	FID↓	CLIP Score↑	User Study↑	FID↓	CLIP Score↑	User Study↑
SDEdit [36]	60.4	0.204	2.7%	78.3	0.203	1.8%
P2P* [15]	46.8	**0.223**	13.6%	55.4	0.207	12.7%
SceneCrafter	**37.2**	0.220	**83.6%**	**38.9**	**0.221**	**85.5%**

Table 1. **Quantitative Comparison on Global Editing.** SceneCrafter shows clear improvement over two baselines in terms of realism, controllability and editing quality, achieving preference rates of over 80% across both editing benchmarks.

Here, $\mathbf{z}_{t-1}^{\text{background}}$ is sampled based on the known pixels in the given image, while $\mathbf{z}_{t-1}^{\text{foreground}}$ for all camera views is sampled simultaneously from the multi-view diffusion model, using the previous iteration $\mathbf{z}_t$. Finally, we merge these two latents where the foreground regions are progressively denoised to background:

$$\mathbf{z}_{t-1} = \mathbf{m} \odot \mathbf{z}_{t-1}^{\text{foreground}} + (1 - \mathbf{m}) \odot \mathbf{z}_{t-1}^{\text{background}} \quad (4)$$

Alpha blending. After applying multi-view repaint to "populated streets" data, we obtain the corresponding "empty streets" data, denoted as $\mathcal{I}^{\text{empty}}$ and $\mathcal{I}^{\text{full}}$. We then employ alpha blending to generate paired data for object insertion or removal. More specifically, we first project all agent bounding boxes onto 2D planes to create a vehicle mask. We then sample any desired number of these masks to form a new composite mask, $\mathbf{m}^{\text{sampled}}$, which contains varying numbers of agents.

The alpha blending of I^{empty} and $\mathcal{I}^{\text{full}}$ is performed as:

$$\mathcal{I}^{\text{sampled}} = \mathbf{m}^{\text{sampled}} \odot \mathcal{I}^{\text{empty}} + (1 - \mathbf{m}^{\text{sampled}}) \odot \mathcal{I}^{\text{full}} \quad (5)$$

which allows us to seamlessly blend the images, creating realistic scenes with any specified number of agents.

3.4. Student Model for Scene Editing

We frame the image-to-image editing as a generation task, conditioning on source images $\mathcal{I}^{\text{source}}$ and the control signals. To effectively incorporate the source images into the model, we introduce an additional conditioning branch that concatenates the latent of the source images $\mathbf{z}_0^{\text{source}}$ directly with the latents $\mathbf{z}_t$. We find that concatenating pixel-level features, such as source images, masks, and raymaps yields better results than using cross-attention techniques, as shown in Sec. 4.4.

For training the editing model, we employ the synthetic dataset detailed in Sec. 3.3, which includes changes in weather, time of day and agents boxes. The editing model's weights are initialized from the weights of the global editing teacher model. We maintain the original conditioning mechanisms of the generation model and also update their parameters during training. This integration ensures that the editing model can leverage the detailed contextual data from the source images while maintaining the generative capabilities of the original model.

At test time, our model takes source images and arbitrary target prompts, then generates target images that preserve the geometric structure and layout of the source images while aligning with the specified prompts. It's worth noting that our model uses box conditions for agent editing rather than masks, and agent insertion or removal are controlled by agent types. Unlike mask-based methods that often produce imprecise boundaries, our box conditioning offers more precise control, especially for smaller objects illustrated in Fig. 1. This allows us to achieve fine-grained edits while maintaining geometric consistency with the original image.

4. Experiments

4.1. Metrics

Evaluating generative 3D editing models presents unique challenges. Metrics for novel view synthesis typically emphasize generation quality, but additional criteria are essential for 3D editing tasks. Specifically, we assess models based on realism, controllability, 3D consistency and editing quality. We utilize existing metrics such as Fréchet Inception Distance (FID) [17] and CLIP Score [16], run a user study, and propose a novel 3D LPIPS metric to comprehensively assess these aspects.

Realism. FID is the most commonly used metric for estimating the realism of generated images by measuring how similar the distribution of edited images is to the distribution of the origin images in a feature space.

Controllability. The CLIP score originally aims to measure how well an image caption aligns with the semantics of an image. We adapt CLIP Score to evaluate controllability by converting a control signal into a text to describe how to edit a image, then calculate the cosine similarity between the text and edited image embeddings.

Editing quality. We conduct a user study to evaluate the editing quality of our method compared to baselines. The user study involved 11 human raters evaluating 20 groups of images, where each group corresponds to a unique edit prompt. For each group, we show users the images generated by three methods (randomized order) and the source images. Users were asked *"which of these images is the most faithful result of editing the source image according to the given prompt?"* and then selected the image they thought most aligned with the question. We compiled their selections to summarize user preferences.

3D Consistency. We leverage the overlapping field of view between cameras to measure consistency across multiple

Figure 4. **Qualitative Comparison on Time of Day Editing.** SceneCrafter is able to conduct realistic editing to 8PM while still maintaining the geometric structure of the input images.

Figure 5. **Attention Weights Manipulation in Prompt-to-Prompt.** Manipulating self-attention weights enables accurate conditioning while preserving the same geometry.

Method	FID↓	
	Removal	Insertion
2D-RePaint [33]	30.6	31.9
MV-RePaint	26.0	28.5
SceneCrafter	**23.5**	**21.7**

Table 2. **Quantitative Comparison on Local Editing.** SceneCrafter, which uses agent boxes to manipulate agents, shows obvious advantages over two mask-based methods, which often struggle with imprecise segmentation, where masks exceed object boundaries. Our box-based conditioning enables more accurate, fine-grained edits, especially for small objects.

Method	FID ↓	3D LPIPS ↓
Real	11.5	0.186
CAT3D [10]	121.3	0.249
SceneCrafter (w/o cond.)	68.5	0.254
SceneCrafter (full)	**36.2**	**0.187**

Table 3. **Quantitative Results on Generation Task.** With conditioning signals encoded, SceneCrafter can generate more realistic multi-view images compared to the baselines. SceneCrafter can even achieve the same degree of 3D consistency as real log data, measured by the novel 3D LPIPS metric.

views. More specifically, for each neighboring view pair (C_i, C_{i+1}), we project C_i into C_{i+1}, as well as C_{i+1} into C_i, and compare the overlapping regions. We use the LPIPS metric as it is known to be sufficiently robust to effects such as exposure differences and motion blur [56], and subsequently also refer to this metric as 3D LPIPS.

4.2. Experimental Settings

Dataset. Since our multi-view driving scene editing and generation tasks are unique, we require a novel dataset to enable control over both global and local scene conditions, as well as camera poses. We curated a dataset that consists of 13,867,496 unique segments of driving videos for training the teacher diffusion models. Each segment consists of 17 frames of 8 surrounding cameras, captured in the frequency of 10 Hz. The log data contains versatile labels such as camera poses, weather, time of day, HD map, and agent bounding boxes, estimated by the AV onboard stack. These labels enable our teacher model to associate control signals with scene generation. We held out 1% videos for testing and used the remaining data for teacher model training.

Implementation details. We base our two teacher models for global and local edits on a pre-trained multi-view diffusion model [10] and fine-tune the global teacher model for final student editing model. We train our models on 128 Google TPU v5 for 100k iterations, with learning rate of $1e^{-5}$ and batch size 128. We generate 1M synthetic paired data to train the student model. We resize all the inputs to 512×512 to align with the pre-trained VAE. At inference time, we use 50 denoising steps with classifier-free guidance [18].

4.3. Results

Qualitative editing results. We demonstrate our qualitative editing results in Fig. 3. Given arbitrary multi-view source images as input (first row), our model can perform many challenging edits, including changing time from day to dawn and night (left), and changing weather from sunny to snowy and foggy (right), while keeping the scene geometrically consistent. We show more creative editing results in the supplemental material.

Comparison to other editing baselines. For global edits, we provide quantitative comparisons with SDEdit [36] and P2P* in Tab. 1, along with qualitative comparisons on time of day editing in Fig. 4. SDEdit is a general image editing method where a partially noised image is denoised to generate a new edited one. We extend Prompt-to-Prompt for editing tasks by replacing the latents of the source images with that of the images to be edited, and the generated target images are the image after editing, denoted as P2P*. In Tab. 1, we observe that our realism, controllability and editing quality are consistently better than other baselines in

Cross Attention	Increase Conditioning	Daytime for Source Images	FID↓	CLIP Score↑
✗	✗	✗	57.1	0.204
✓	✗	✗	41.5	0.202
✓	✓	✗	39.9	0.214
✓	✓	✓	**36.2**	**0.223**

Table 4. **Ablation Study on Prompt-to-Prompt Design.** We show three key components to improve the quality of synthetic data. First, replacing self-attention weights led to better geometric consistency for generated pairs. Second, using additional conditions (agent features and HD maps) improved controllability. Finally, using only daytime source images enhanced the generation quality by a large margin.

all but one instance. SceneCrafter outperforms SDEdit and P2P* by 40.6 and 16.5 in FID and achieves a 83.6% and 85.5% preference rating from our user study, respectively, demonstrating that our method conditions precisely on the source images and achieves accurate editing. Visually, as shown in Fig. 4, our method excels at generating consistent geometry while keeping highly detailed textures. Other baselines provide results aligned to the target prompts, but exhibit sub-optimal geometric consistency.

We also compare our method against 2D-RePaint and MV-RePaint for local editing tasks, as shown in Tab. 2. For the 2D-RePaint baseline, we utilize a pretrained Stable Diffusion model [43], applying it independently to each camera view. For the MV-RePaint baseline, we use our editing model without conditioning on source images and adopt the multi-view repaint method in Sec. 3.3.2, using 2D-projected agent bounding boxes as masks. We evaluated on agent insertion and removal scenarios. Our method consistently demonstrated superior object editing quality compared to both approaches, and achieves the best FID scores. 2D RePaint lacks editing type control and MV-RePaint, on the other hand, relies heavily on masks, which might not always accurately align with object boundaries. Our method conditions on specific agent bounding boxes, allowing precise insertion and removal.

Comparison to other generation baselines. To validate that our editing model also possesses generation ability, we present some pure generation results in Tab. 3. We compare with CAT3D [10] by using a single view as the conditional input and generating the other seven views. We also test our SceneCrafter without any conditions. Notably, the full SceneCrafter model surpasses the baselines across both metrics. Moreover, our quantitative results closely match real image data, particularly in the 3D LPIPS metric. This superior performance can be attributed to the use of local conditions, which effectively grounds the geometric structure of 3D scenes and enables the creation of highly 3D consistent scenes well aligned with the real-world. In contrast, the other two baselines lack such geometrical constraints and show lower 3D consistency.

Method	FID ↓	CLIP Score ↑
Cross-attention	50.3	0.203
Concatenation	**37.2**	**0.220**

Table 5. **Ablation Study on Source Image Conditioning.** Concatenating the multi-view source images clearly outperforms applying the cross-attention operation as woth other conditions.

4.4. Ablation Study

We ablate our design choices for Prompt-to-Prompt in Tab. 4 and different ways for source image condition in Tab. 5. We conduct ablation studies with time of day edits.

Design choices for prompt-to-prompt. Tab. 4 demonstrates the improvement on generating synthetic data by introducing several key components, such as replacing self-attention weights, including more conditions and using only daytime for source images. Starting from the original Prompt-to-Prompt (first row), we find that replacing self-attention weights results in more consistent geometry in pixel-level image-to-image translation (second row). Including more scene conditions such as agent boxes and HD maps also helps to improve the condition quality across paired synthetic images (third row). Using daytime as the time of day for source images further enhances the generation quality and controllability (fourth row).

We also show qualitative visualizations of the effects of replacing different attention layers in Fig. 5. We compare not replacing any attention layers, replacing cross-attention layers, and replacing self-attention layers. We find that replacing self-attention layers only preserves the geometric consistency of the generated pairs.

Source image conditioning. We explore methods for conditioning source images in Tab. 5. For cross-attention, we encode images using the VAE and feed them directly, along with other conditions, into the cross-attention module. We find that concatenation yields better performance, notably improving FID by 13.1. This suggests that concatenation is more effective for pixel-level features like source images, masks, and raymaps, while cross-attention performs better with global and local conditions.

5. Conclusion

We present SceneCrafter, a versatile editor for realistic 3D consistent multi-view driving scene image editing. We decompose the problem into two steps. First, we train teacher models to generate high-quality synthetic data using novel training paradigms. Next, we distill knowledge from these teacher models by training a unified student model on the generated dataset. We show that our student model is able to conduct several challenging editing tasks, with either global or local editing prompts.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019. 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020.

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, 2022. 3

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2023. 3

[5] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 3, 4

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[7] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 2, 3

[8] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. In *ICLR*, 2021. 3

[9] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024. 3

[10] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. In *NeurIPS*, 2024. 3, 4, 7, 8

[11] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *NeurIPS*, 2024. 3

[12] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *CVPR*, 2023. 3

[13] Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *NeurIPS*, 2024. 2, 3

[14] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *ECCV*, 2024. 3

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023. 1, 2, 3, 4, 6, 7

[16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 6

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 6

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 7

[19] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3

[20] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving, 2023. 3

[21] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 3

[22] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 2

[23] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, James Guo, Dragomir Anguelov, and Mingxing Tan. Emma: End-to-end multimodal model for autonomous driving, 2024. 2, 3

[24] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 4

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 3

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

[28] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 3

[29] Daiqing Li, Huan Ling, Amlan Kar, David Acuna, Seung Wook Kim, Karsten Kreis, Antonio Torralba, and

Sanja Fidler. Dreamteacher: Pretraining image backbones with deep generative models. In *ICCV*, 2023. 3

[30] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. In *NeurIPS*, 2022. 3

[31] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023. 3

[32] Yiyuan Liang, Zhiying Yan, Liqun Chen, Jiahuan Zhou, Luxin Yan, Sheng Zhong, and Xu Zou. Driveeditor: A unified 3d information-guided framework for controllable object editing in driving scenes, 2024. 3

[33] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2, 5, 7

[34] Enhui Ma, Lijun Zhou, Tao Tang, Zhan Zhang, Dong Han, Junpeng Jiang, Kun Zhan, Peng Jia, Xianpeng Lang, Haiyang Sun, Di Lin, and Kaicheng Yu. Unleashing generalization of end-to-end autonomous driving with controllable long video generation, 2024. 3

[35] Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 3

[36] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 3, 6, 7

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[38] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. In *NeurIPS*, 2024. 2, 3

[39] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 3

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[41] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[42] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021. 3

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 4, 8

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 3

[45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 3

[46] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 2, 3

[47] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM TOG*, 2021. 3

[48] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving, 2023. 3

[49] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *CVPR*, 2024.

[50] Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[51] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *CVPR*, 2024. 3

[52] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. 2023. 3

[53] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *ICCV*, 2023. 3

[54] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized Predictive Model for Autonomous Driving. In *CVPR*, 2024. 3

[55] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *CVPR*, 2023. 3

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[57] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021. 3

[58] Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv:2403.06845*, 2024. 3