LANGDRIVEEDIT: LANGUAGE-DRIVEN IMAGE EDITING FOR STREET SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensuring the safety of autonomous driving systems requires rigorous evaluation across diverse street scene conditions within the Operational Design Domain (ODD), such as lighting, weather, traffic, and road variations. Yet collecting real-world data to cover this spectrum is costly, time-consuming, and often impractical. Recent advances in language-driven image editing offer a promising alternative by simulating diverse scenarios through text-based modifications. However, progress has been limited by the absence of a dedicated dataset for driving-scene editing. To address this gap, we introduce, to the best of our knowledge, the first dataset specifically designed for **language-driven editing of driving scenes**. Our dataset combines real-world and synthetic street scene images and supports 12 distinct editing tasks, spanning global modifications (e.g., weather, season, time of day) and fine-grained local edits (e.g., altering vehicle or pedestrian attributes). Crucially, each edit is paired with **detailed textual and visual instructions**, and, together with our proposed supervised and unsupervised fine-tuning objectives, enables state-of-the-art image editing models to follow instructions faithfully and preserve critical content. Experimental results demonstrate that training language-driven editing models with our dataset and objectives yields substantial gains in prompt alignment, visual fidelity, generation realism, and downstream driving-task performance on edited street scene images, across diverse driving domains.

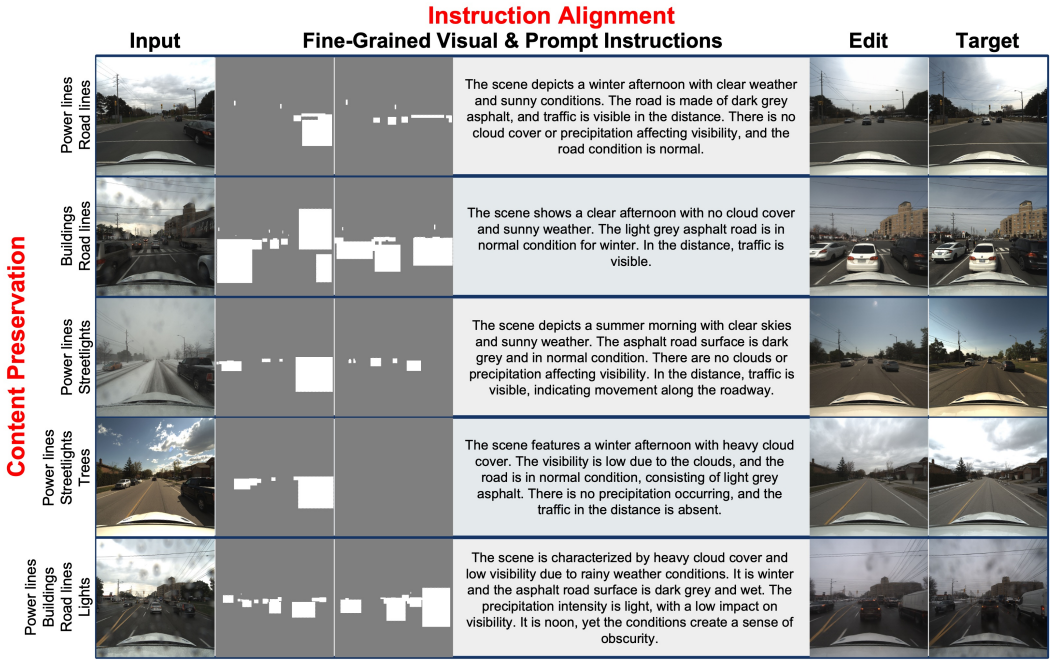

Figure 1: LangDriveEdit targets improving two critical requirements in driving scene editing: **content preservation** and **instruction alignment**. These two requirements are supported via fine-grained visual and prompt instructions (Sec. 3) and carefully designed training objectives (Sec. 4). Given a mask of dynamic objects to remove and add respectively, and a global editing prompt, we visualize our edits of given input images.

1 INTRODUCTION

Ensuring safety is a central challenge in deploying autonomous driving. Real-world testing within the Operational Design Domain (ODD) is limited by dynamic factors such as lighting, weather, traffic, and road conditions, making comprehensive data collection infeasible (Mehlhorn et al., 2023). Generative models offer a scalable alternative for synthesizing diverse environments (Gao et al., 2023), and instruction-guided image editing in particular enables fine-grained, language-based control while preserving realism through large-scale pretrained sources (Ramesh et al., 2022; Betker et al.; Rombach et al., 2022; Brooks et al., 2023b; Zhao et al., 2024).

Nonetheless, two properties critical to autonomous driving scenes are not explicitly enforced when applying generative instruction-guided editing: **Content Preservation** and **Instruction Alignment**. **Content preservation** focuses on retaining the unedited elements of a driving scene during transformation (Shi et al., 2024). Editing models may unintentionally modify or remove essential content—such as traffic signs, lane markings, or surrounding vehicles, potentially leading to incorrect visual signals for downstream perception and planning systems by altering safety-critical elements. Meanwhile, **instruction alignment** refers to the accurate execution of detailed, multi-attribute natural-language instructions (Shi et al., 2024). Strong instruction alignment not only supports human-in-the-loop workflows but also enables systematic exploration of the combinatorial space of scene factors (weather, illumination, traffic composition, viewpoint, and beyond), ensuring that each syntactic variation of an instruction is faithfully realized in the output, thereby achieving a degree of diversity and granularity unattainable through unguided random sampling.

Therefore, we pose two fundamental questions for current instruction-guided editing models:

> ***Q1**: How can instruction-guided editing models generate variability in the environment while preserving unedited portions of a driving scene unchanged?*
> ***Q2**: How can instruction-guided editing models' generation be controlled given precise editing instructions in driving scenes?*

The primary bottleneck in addressing these questions lies in the lack of **paired datasets** of **fine-grained visual or textual prompts**. We introduce the *LangDriveEdit* Dataset (Figure 1). It includes large-scale paired real-world driving images with fine-grained instructions (Sec.3.1), with a supplementary synthetic part (Sec.C.1). Our **real-world** images capture **a large amount of environmental variations**, such as season, lighting, and weather, alongside multiple concurrent object-level differences among road users (Sec.3.1). To enable fine-grained control in diffusion-based generation, we pair precise editing instructions with pixel-level masks, created using large language models integrated into a multi-modal vision pipeline. Unlike object-centric image generation, traffic scenes are densely populated with vehicles, pedestrians, and buildings, making natural language prompts alone insufficient (Figure 3). Our masks encode localized semantics at the pixel level, ensuring accurate description and manipulation of complex driving scenes. Furthermore, to scale driving scene editing to various traffic conditions, we introduce unsupervised training methods that encourage content preservation via cycle and identity objectives, and also instruction alignment via CLIP similarity objectives and adversarial training. We demonstrate that models trained this way on the *LangDriveEdit* dataset achieve strong **content preservation** and **instruction alignment** (Sec.5). Our contributions can be summarized as follows:

1. **New Paired Datasets for Driving Scene Editing.** To the best of our knowledge, *LangDriveEdit* is the first dataset of large-scale paired images with the support of diverse editing types and fine-grained instructions, designed for instruction-guided editing of driving scenes.
2. **Automatic Generation of Prompts and Visual Masks as Instructions.** Our pipeline streamlines the generation of editing prompts and pixel-level masks for *both* real-world and synthetic environments, employing a novel annotation framework that leverages vision-language models and depth estimation to extract environmental and object-level details.
3. **Significant Improvements in Driving Scene Editing.** Across different state-of-the-art editing models, our comprehensive experiments demonstrate that after our fine-tuning on our *LangDriveEdit* dataset, both **content preservation** and **instruction alignment** are largely improved.
4. **Impact on Downstream Driving Tasks.** We demonstrate that edited images produced with our dataset can **improve road segmentation performance** on an out-of-distribution driving dataset, highlighting the potential of instruction-guided editing for safety-critical applications.

2 Related Work

Image Editing Dataset. Building image-editing datasets is more challenging than Text-2-Image (Betker et al.; Ramesh et al., 2021; 2022; Rombach et al., 2022), with data scarcity a key bottleneck (Wang et al., 2023a; Hui et al., 2024). Existing efforts such as MagicBrush (Zhang et al., 2024b), InstructPix2Pix (Brooks et al., 2023a), and SeedEdit (Shi et al., 2024) either rely on manual annotation, synthetic data, or iterative refinement, but remain limited to simple object edits. **Image Editing via Generation.** Advances in large diffusion models (Kawar et al., 2022; Saharia et al.; Chen et al., 2023) have enabled instruction-driven editing, with methods like InstructPix2Pix, HIVE, and UltraEdit pushing the field forward, though primarily in generic domains. **Image Editing for Autonomous Driving.** Driving-scene editing has been explored through NeRFs, Gaussian Splatting, and multi-condition generation (Liang et al., 2025; Gao et al., 2023), yet instruction-guided editing remains underexplored due to the absence of datasets. We address this gap with *LangDriveEdit*, the first instruction-driven editing dataset tailored to autonomous driving. Due to space limits, we refer readers to Section A for a more comprehensive survey of related works.

3 LangDriveEdit Dataset

The construction of the dataset involves a detailed annotation process to capture varying edits in driving scenes while maintaining consistency. We overview our dataset construction pipeline in Figure 2. In this section, we explain our data collection and annotation.

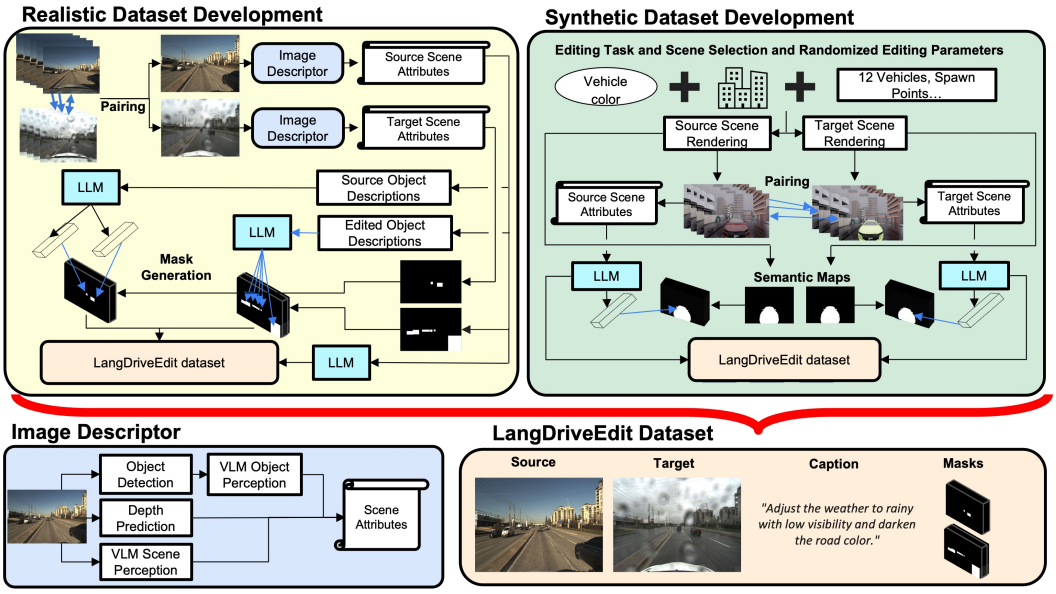

Figure 2: **LangDriveEdit Construction**. Real-world data are paired by camera pose and annotated using an image descriptor pipeline (Sec. 3.1.1) passed to an LLM to produce instructions (Sec. 3.1.2). For synthetic image pairs we simulate two frame sequences performing one of 12 editing tasks (Sec. C.1.1). Our dataset is composed of image pairs, editing instructions, and two masks indicating objects to add and to remove.

3.1 Semantic Alignment for Unpaired Real-World Driving Environments

Our primary data contribution consists of large-scale paired real-world driving images with hierarchical, fine-grained annotations. As demonstrated in the experiments (Section 5), these high-quality driving scene images form the foundation for training image editing models to generate diverse driving environments. We show our dataset statistics in Table 1.

We align images and semantics in Boreas (Burnett et al., 2023), which is a multi-season autonomous driving dataset collected by driving a repeated route throughout one year. In addition to ideal climates, Boreas features adverse weather conditions (rain, snow, fog) that are critical for rigorously evaluating and expanding operational design domains.

Table 1: **Real-World Dataset Statistics**. Left: distribution of edit types across the Boreas. Right: distribution of the number of edit types per example.

Edit type	Count	Percentage
Road Conditions	146,159	15%
Time of Day	235,710	24%
Traffic	263,064	27%
Traffic Light	55,998	6%
Weather	269,715	28%

Partition by Number of Subedits		
1 Edit Type	16,623	4.86%
2 Edit Types	95,295	27.88%
3 Edit Types	160,413	46.93%
4 Edit Types	64,385	18.84%
5 Edit Types	4,922	1.44%
Total Examples	341,796	-

3.1.1 Fine-grained Paring and Decomposition of Complex Driving Scenes

Real-world images in Boreas are largely collected from unpaired scenes and camera poses. To align images into paired scenes with aligned camera poses, for each pair of driving sequences, we extract corresponding frames by minimizing camera pose disparities, below a fixed tolerance, employing an equally weighted sum of angular orientation and Euclidean position. We show this in Equation 1 where x is the camera's Euclidean position and ϕ, θ, ψ are its roll, pitch, and yaw respectively.

$$\begin{aligned} I_{target} &= \arg\min_{I \in \mathcal{I}} \ \text{dist}(I, I_{source}) \\ \text{dist}(I_a, I_b) &= \|\vec{x}_a - \vec{x}_b\|_2 + |\phi_a - \phi_b| + |\theta_a - \theta_b| + |\psi_a - \psi_b| \end{aligned} \tag{1}$$

where $\mathcal{I}$ denotes the neighboring frames of I_{source}, and a, b are the indices of two such frames. To collect hierarchical and multimodal scene descriptions, we introduce "Image Descriptor" (Figure 2), a training-free pipeline inspired by (Yao et al., 2025). Real-world driving scenes introduce significant annotation challenges. *First*, most large-scale real-world recordings lack multimodal sensors, leaving RGB-based global and instance-level descriptions less informative. *Second*, the real world typically includes complex scene variations and compositions (see Figure 3 left, where images are captured with the same camera pose at different times).

Our "Image Descriptor" is a **comprehensive annotation system**: it integrates vision-language and depth estimation models that can generate semantic-aligned scene descriptions at two levels:

Multimodal Environments Descriptions. We first extract global information about the scene:

1. We use an image-based vision-language model (VLM) (Chen et al., 2024a) for a global interpretation of extremely fine-grained attributes. We show the VLM prompt in Appendix E.1.
2. To estimate object distances, we apply a metric depth estimation model, Metric3d (Hu et al., 2024), to the full image, producing a depth map whose values correspond to real-world distances.

Instance-Level Semantic Decomposition. After preparing the global description, we then record objects present in the scene:

1. We run a 2D object detector (Owlv2 (Minderer et al., 2024)) that returns, for each detected object, a bounding box, a class label (from the set 'ambulance', 'bicycle', 'traffic light', 'traffic cone', 'person', 'car', 'motorcycle', 'bus', 'building', 'fire truck'), and a unique object ID.
2. For each object, we crop the global depth map (from the 2nd step above) to its bounding box and then refine that region with a binary mask from the Segment Anything Model (SAM (Kirillov et al., 2023)), ensuring we exclude background pixels. The object's distance is taken as the mean depth over this masked area.
3. We invoke the VLM (Chen et al., 2024a) on each object's bounding box to extract additional attributes, such as vehicle color or traffic-light state.

We show an example annotation in Appendix E.

3.1.2 Dense Editing Instructions and Content Preservation

We enable controllable image editing, with uninstructed content preserved, by using two annotation-driven guidances: textual instructions and spatial masks.

Instruction Generation. Using the global scene annotations we prepared in Section 3.1.1, such as weather, time of day, and road conditions, we employ ChatGPT 4o-mini to generate structured editing instructions separately for the input source image and the target driving scene (App. F.2). Our generated instruction only describes what the target image should look like.

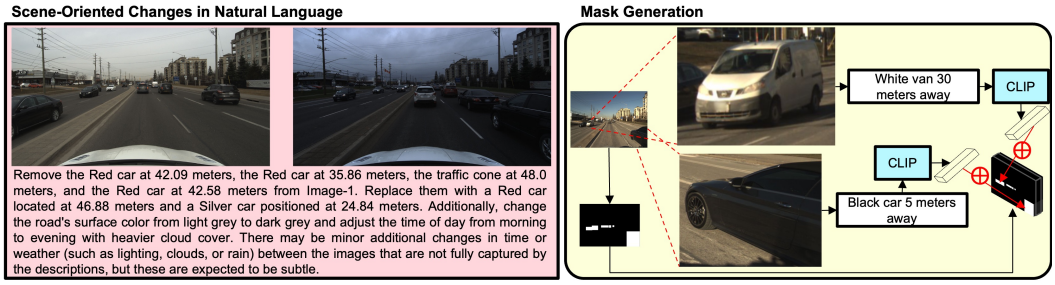

Figure 3: **Driving Scene Features are Dense**. Left: It is difficult to describe rich semantics for multiple object-oriented changes in natural language. Right: To provide fine-grained instructions with rich semantics, we construct both binary removal/addition masks and CLIP text features of instances (Section 3.1.2).

Extracting Semantic-Rich Removal-Addition Masks. To capture nuanced scene edits, each global instruction is paired with two instance-aware masks. Specifically, we generate: (1) a *removal mask* for the source image, which identifies objects to be eliminated, (2) an *addition mask* for the target image, which highlights regions designated for new object placement. Together, they support fine-grained object edits while preserving non-targeted regions. See details in Appendix D.

Expanding Masks with CLIP Text Features. Unlike traditional image editing approaches that primarily target sparse, object-centric modifications (e.g. (Zhao et al., 2024; Brooks et al., 2023a)) driving scenes are inherently much denser and more complex. It is unrealistic to precisely describe rich semantics for multiple instance-specific changes in natural language prompts. To enable the accurate execution of detailed multi-attribute instructions, we use masks expanded with CLIP features (Radford et al., 2021). For each masked object, we encode its text description into CLIP and assign the encoded feature to each masked pixel. This process is shown in Figure 3 right. Objects are processed in descending order of their distances to the ego camera, allowing the CLIP features of closer objects to overwrite those of farther, occluded objects at overlapping pixels.

4 Language-Guided Image Editing

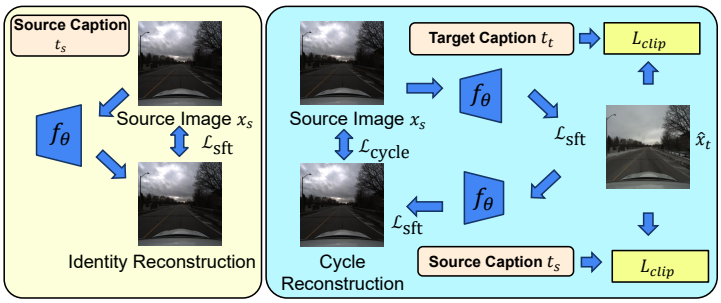

Figure 4: Training language-guided driving scene image editing. Our training pipeline supports both supervised training for paired images and unsupervised training for unpaired ones (e.g. downstream unseen real scenarios). We include three training objectives: supervised fine-tuning $\mathcal{L}_{\text{sft}}$ (Section 4.1), cycle consistency $\mathcal{L}_{\text{cycle}}$ (Section 4.2), and $\mathcal{L}_{\text{clip}}$ (Section 4.3).

In this section, we introduce a suite of training objectives to explicitly encourage **content preservation** and **instruction alignment**. Our training pipeline integrates both supervised and unsupervised objectives, enabling editing models to benefit from paired data for precise editing control when available, while remaining applicable to large-scale, unpaired datasets. In Section 5, we show that this supports the fine-tuning of different image editing models.

4.1 Supervised Fine-Tuning for Instruction Alignment

When paired source–target examples are available, we train image editing models with supervised fine-tuning. Given paired training samples $(x_s, x_t, t_s, t_t, M_r, M_a)$, the generator model f_θ produces an edited image, and we calculate the supervised fine-tuning loss in Eqn. 2.

$$\begin{aligned} \hat{x}_t &= f_\theta(x_s, t_t, M_r, M_a), \\ \mathcal{L}_{\text{sft}} &= \lambda_{\text{sft}} \left\| x_t - \hat{x}_t \right\|_1 + \lambda_{\text{sft-lpips}} \left\| \phi(x_t) - \phi(\hat{x}_t) \right\|_2, \end{aligned} \tag{2}$$

Each training instance includes a source scene (x_s), two natural language instructions describing the transformations from the source to the target (t_t), from target to the source (t_s), a mask of instances to remove and add from the source scene (M_r, M_a), and the resulting edited target (x_t). $\hat{x}_t = f_\theta(x_s, t_t, M_r, M_a)$ is the generator's output. $\phi(\cdot)$ denotes the feature extraction function of a pretrained VGG network (Zhang et al., 2018).

With this design, we can explicitly guide the model toward faithful instruction following. By conditioning the model on instruction-derived pixel masks, we constrain modifications to specified areas, encouraging the model to localize edits. This ensures unedited structures, such as road geometry and lane markings, are unchanged while surrounding vehicles and global features are edited. Masked supervision penalizes deviations in non-edited regions, which also supports content preservation.

As a special case of $\mathcal{L}_{\text{sft}}$, when x_s and x_t are the same image (with blank removal/addition masks), we are essentially asking the editing model to preserve the content:

$$\mathcal{L}_{\text{sft}} = \lambda_{\text{id}} \left\| f_\theta(x_s, t_s, \emptyset, \emptyset) - x_s \right\|_1 + \lambda_{\text{id-lpips}} \left\| \phi(f_\theta(x_s, t_s, \emptyset, \emptyset)) - \phi(x_s) \right\|_2 \quad \text{(identity preservation)} \quad (3)$$

This special case of $\mathcal{L}_{\text{sft}}$, i.e, an identity objective, enforces that when editing instructions correspond to no change (e.g., blank masks or re-adding removed content), the model reproduces the input. This teaches the model to preserve dynamic scene instances that are not specified in the editing instruction.

4.2 Language-Guided Cycle Consistency and Identity Preservation

While supervised fine-tuning provides precise control, acquiring paired data is costly, and the reliance on paired data limits its scalability to driving datasets where explicit ground truth edits are unavailable, especially on unseen driving scenes in the wild. Therefore, we choose to include complementary unsupervised constraints via cycle consistency and identity preservation, such that we can use them in OOD unsupervised cases.

Cycle consistency extends this principle by encouraging reversibility. Without additional constraints, generative editing models may alter portions of the scene in regions unrelated to the instruction. By requiring the original image to be recoverable after a forward–backward editing cycle, the model is penalized for unnecessary deviations from the input. This encourages content preservation by discouraging drift.

$$\begin{aligned} \hat{x}_s &= f_\theta(f_\theta(x_s, t_t, M_r, M_a), t_s, M_a, M_r) \\ \mathcal{L}_{\text{cycle}} &= \lambda_{\text{cycle}} \left\| \hat{x}_s - x_s \right\|_1 + \lambda_{\text{cycle-lpips}} \left\| \phi(\hat{x}_s) - \phi(x_s) \right\|_2. \end{aligned} \quad (4)$$

By combining pixel-level L1 and perceptual LPIPS losses, we enforce structural fidelity while allowing stylistic variation.

4.3 Language-Guided CLIP Loss for Content Preservation

Reconstruction-based supervision is insufficient on its own: the model may collapse to an identity mapping, avoiding all edits to minimize loss. To overcome this degeneracy, we incorporate a complementary alignment signal based on language–image similarity, described in the following subsection.

$$\mathcal{L}_{\text{clip}} = \lambda_{\text{clip}} \left(1 - \text{sim}_{\cos}\left(\text{CLIP}_I(\hat{x}_b),\ \text{CLIP}_T(t_b)\right)\right) + \lambda_{\text{clip}}\ \text{sim}_{\cos}\left(\text{CLIP}_I(\hat{x}_b),\ \text{CLIP}_T(t_a)\right), \quad (5)$$

where CLIP_I indicates CLIP's image feature, CLIP_T is for CLIP's text feature, and $\text{sim}_{\cos}$ for cosine similarity. To counteract degraded generation, in Equation 5, we first incorporate a language-guided CLIP similarity loss (the first term on right-hand side). This loss measures the alignment between the generated output and the provided instruction using CLIP. The aligned CLIP loss encourages outputs to move toward the intended semantic edit, thereby reinforcing instruction alignment. We also introduce a misalignment penalty term (the second term on right-hand side), which discourages similarity to input image description to prevent the model from reproducing the input.

5 Experiments

5.1 Settings

We evaluate our training strategies and pixel-level instructions on two competitive image editing models: UltraEdit (Zhao et al., 2024) and CycleGAN-Turbo (Parmar et al., 2024). Both models

are fine-tuned from powerful diffusion backbones trained on large-scale datasets. Following the evaluation protocol in (Zhang et al., 2024c), we assess editing performance using L1 distance, L2 distance, CLIP image similarity, and DINO similarity. Additional implementation details are provided in App. G.

We structure our experiments to address two core questions in Section 1: 1) The importance of paired driving scene data and training objectives for generating desired edits while maintaining scene integrity (Sec. 5.2). 2) The extent to which fine-grained prompting enhances models' generation to align with instructions and handle the complex interplay of foreground and background modifications in driving scenes (Sec. 5.3). In Sec. 5.4 we extend our editing to out of distribution images and evaluate the downstream performance of a road segmentation model using our edits.

5.2 Controlling Generation via Precise Editing Instructions

We first study the importance of paired driving scene images and training objectives for performing driving scene editing. Our quantitative results in Table 2 show that models trained with our precise editing instructions consistently outperform their baselines. Our trained models ("Ours") trained with Removal-Addition masks (Sec. 3) and training objectives (Sec. 4) show the lowest L1 and L2 scores which suggest high content preservation in unedited regions, and high instruction following in edited regions. Furthermore their high CLIP and DINO scores indicate the strongest instruction alignment while preserving the scene content. Qualitatively, Figure 5 shows that the base CycleGAN-Turbo often fails to modify the images following the pixel guidance (rows 1, 2, 6). Bagel, while it does not support masks, fails to preserve the scene content (rows 3, 4, 6), and UltraEdit, while it only supports binary masks, may fail to follow the text prompt (rows 1, 4, 5). Our models ("Ours") trained with Removal-Addition masks and training objectives generate edits with strong alignment to the text-prompts. Furthermore, with the masks, the models are able to make adjustments to traffic according to instruction (rows 1, 2, 3, 4, 6, 7).

Table 2: Models labeled "ours" are trained using all objectives on the real-world subset of the LangDriveEdit dataset, combined with unsupervised objectives on the NuScenes dataset. UltraEdit and CycleGAN-Turbo refer to pretrained models without any additional fine-tuning. We further compare UltraEdit-Text-SFT and UltraEdit-Mask-SFT, two variants trained with supervised fine-tuning but differing in how object changes are specified. In UltraEdit-Text-SFT, object positions are described exclusively through text, whereas in UltraEdit-Mask-SFT, object positions are conveyed using Removal-Addition masks. More details of their definition can be found in H.2. The best results for each setting are highlighted.

Model	L1 (↓)	L2 (↓)	CLIP (↑)	DINO (↑)
Bagel				
Bagel	0.2245	0.0891	0.8399	0.7261
UltraEdit				
UltraEdit	0.2282	0.0927	0.8475	0.7688
UltraEdit-Text-SFT	0.2336	0.1016	0.8583	0.7319
UltraEdit-Mask-SFT	0.1929	0.0676	0.8798	0.8173
UltraEdit (Ours)	0.1144	0.0296	0.9312	0.9024
CycleGAN-Turbo				
CycleGAN-Turbo	0.1993	0.0649	0.8007	0.6378
CycleGAN-Turbo (Ours)	0.1401	0.0383	0.8800	0.8333

5.3 Fine-grained Instructions for Dense Driving Scenes

In Table 2, we also conduct an ablation study where we train a diffusion model that employ text with binary masks for localized edits ("UltraEdit-Text-SFT"). However, it can only perform on par with the base model. When we switch to using masks augmented with clip features to describe traffic patterns, we see an significant improvement in all scores suggesting higher instruction alignment in edited regions and content preservation in unedited regions. Our UltraEdit and our CycleGAN-Turbo each show a 40 % reduction in L1 and L2 compared to their second best counterparts. Adding unsupervised

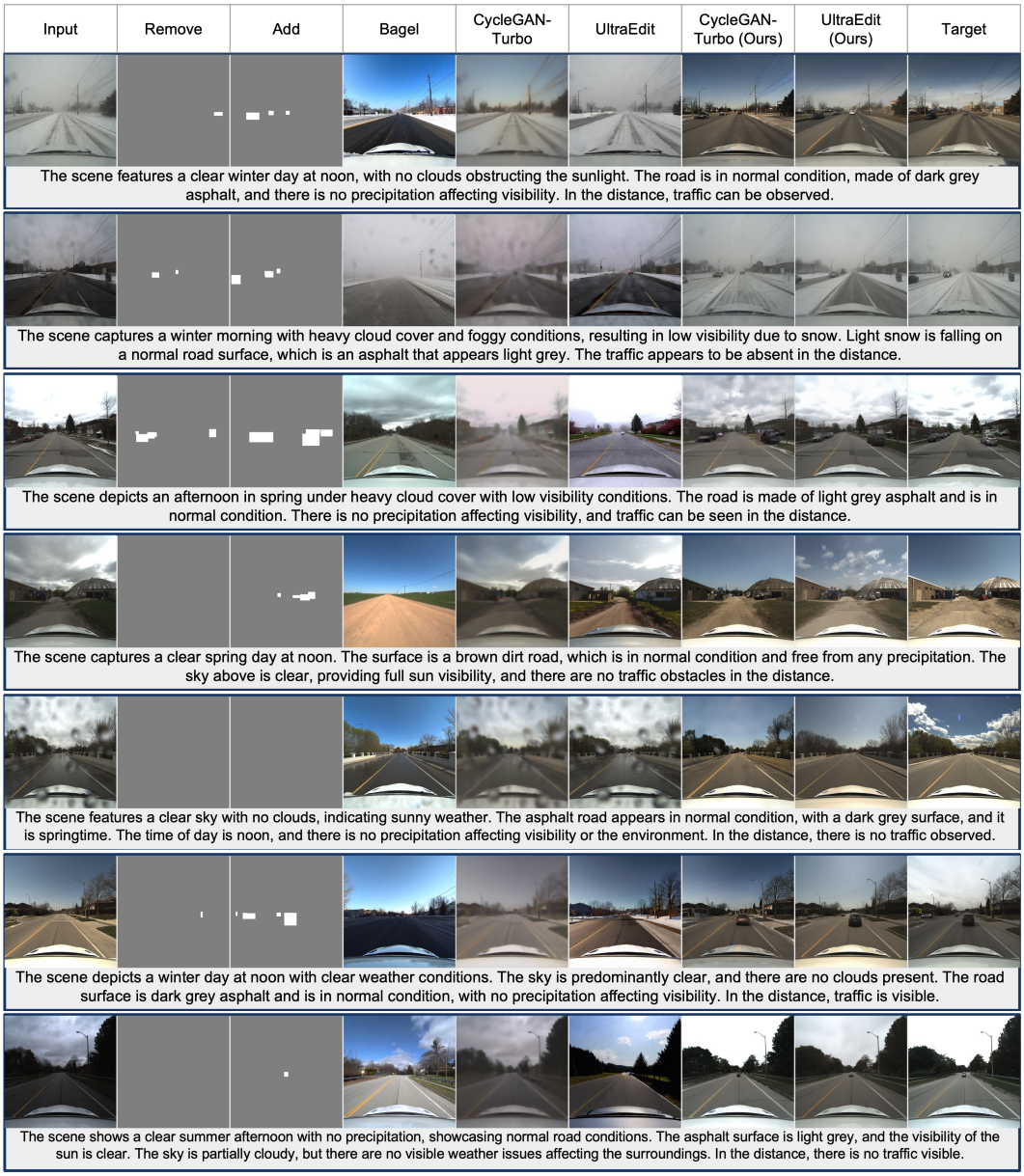

Figure 5: **Visualizations on Boreas**. Results of our models trained on Boreas compared to their baselines. The masks (projected as binary images) enable modifications to traffic while the text-prompt informs the desired global appearance.

training objectives to the Removal-Addition masks ("Ours") leads to a further improvement. An additional qualitative study of our loss functions in App. B.

5.4 Edited Street Scenes for Downstream Driving Tasks

We apply our unsupervised losses to extend the method to NuScenes (Caesar et al., 2020), an out-of-distribution dataset. To this end, we jointly train an editing model on both the real-world Lang-DriveEdit dataset and NuScenes. We use this model to synthesize images from one random quarter of NuScenes under various weather conditions. Then we augment the original NuScenes quarter with these images and train two bird's-eye-view

Table 3: User Study. Overall preference distribution and win rates across 12 questions.

Model	Pref. (%)	Win
CycleGAN-Turbo	11.4	0.0
Bagel	15.9	8.3
Ours	**72.7**	**91.7**

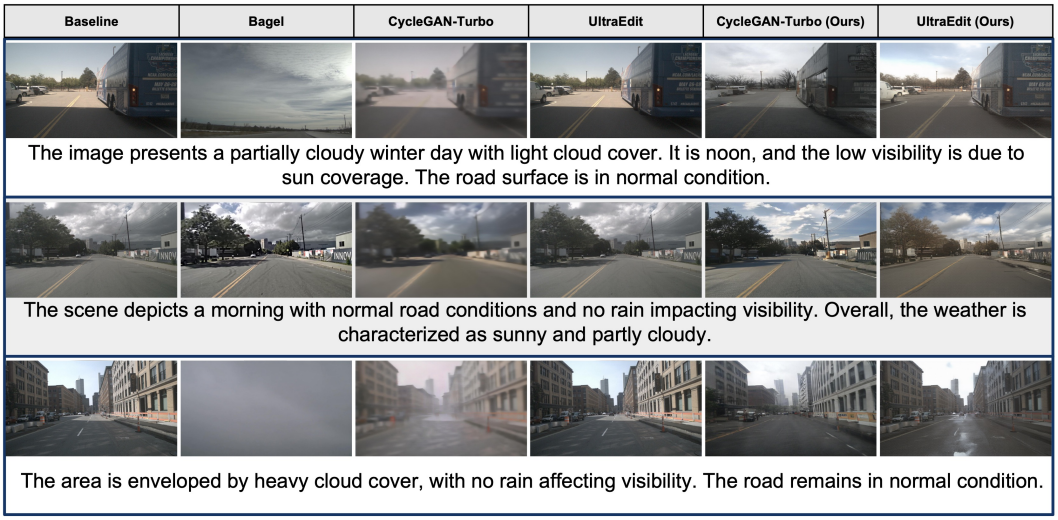

Figure 6: **Out-of-Distribution Generations**. Side by side example of unedited and edited images by Bagel, UltraEdit, CycleGAN-Turbo, and our CycleGAN-Turbo and UltraEdit models jointly finetuned on our real-world dataset and NuScenes. We use the images produced by our model to train a bird-eye-view lane detection model.

(BEV) map segmentation models with and without synthesized images. Following (Liu et al., 2023), we report the highest IoU across different thresholds for each class separately in Table 4. This augmentation leads to an 33% improvement in the average of all classes compared to a baseline trained solely on NuScenes.

Table 4: BEV Map Segmentation. Intersection-over-Union (IoU) across 6 classes and the class-averaged IoU.

Modality	Original	Augmented
Drivable Area	0.5834	0.6448
Ped. Crossing	0.0533	0.1147
Walkway	0.1626	0.2251
Stop Line	0.0609	0.1059
Carpark Area	0.0981	0.1776
Divider	0.1569	0.2180
Mean	**0.1859**	**0.2477**

Figure 6 compares our synthetic NuScenes images with those from CycleGAN-Turbo, UltraEdit and Bagel. Our method shows the superior instruction alignment (rows 1, 2), whereas CycleGAN-Turbo produces blurry and misaligned outputs (rows 1, 3), UltraEdit doesn't edit the image (rows 1, 2, 3), and Bagel sometimes fails to maintain scene consistency (rows 1, 3).

We conducted a user study on the quality of our NuScenes edits produced by Bagel, CyclGAN-Turbo and Our trained CycleGAN-Turbo. We analyzed the responses of 17 participants over 12 images for a total of 176 responses. In each question, we asked participants to "select the better image based on the target scene description. Consider which image better matches the described target scene while maintaining image quality and realism." and "which image better satisfies the editing instruction while preserving the unedited parts of the scene?" As shown in Table 3 preferred our edits for balancing instruction alignment, content preservation and quality.

6 Conclusion

In this work, we present the LangDriveEdit dataset, a significant step forward in the simulation and evaluation of autonomous driving systems through language-driven image editing. By introducing a large-scale, paired dataset with fine-grained visual instructions and precise spatial masks, we enable more controllable, realistic, and diverse scene modifications. Experimental results across state-of-the-art models demonstrate marked improvements in both content preservation and instruction alignment, underscoring the dataset's utility. LangDriveEdit not only fills a critical gap in existing benchmarks but also opens new avenues for research in instruction-guided scene editing for safe and robust autonomous driving.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

LLMs did not play a significant role in either the research ideation or the writing of this paper. Their use was limited to correcting minor grammatical issues and typographical errors.

LIMITATIONS

Our approach relies on language models for generating editing instructions, which may introduce hallucinations or factual inaccuracies. While our precise mask conditioning helps mitigate these issues by constraining modifications to specific regions, some inconsistencies remain. Future work could explore several robustness enhancements: implementing multi-step verification where an LLM validates initial instructions, incorporating human feedback through active learning or implementing automatic consistency checking between generated instructions and source images.

REFERENCES

Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

James Betker, Gabriel Goh, Li Jing, TimBrooks, Jianfeng Wang, Linjie Li, LongOuyang, JuntangZhuang, JoyceLee, YufeiGuo, WesamManassra, PrafullaDhariwal, CaseyChu, Yunxin-Jiao, and Aditya Ramesh. Improving image generation with better captions. URL https://api.semanticscholar.org/CorpusID:264403242.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023a.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023b. URL https://arxiv.org/abs/2211.09800.

Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, Angela P Schoellig, and Timothy D Barfoot. Boreas: A multi-season autonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2):33–42, 2023. doi: 10.1177/02783649231160195.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W. Cohen. Subject-driven text-to-image generation via apprenticeship learning, 2023.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024a.

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

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. VQGAN-CLIP: open domain image generation and editing with natural language guidance. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXVII*, volume 13697 of *Lecture Notes in Computer Science*, pp. 88–105. Springer, 2022. URL https://doi.org/10.1007/978-3-031-19836-6_6.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

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. *arXiv preprint arXiv:2310.02601*, 2023.

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024a.

Zhi Gao, Yuntao Du, Xintong Zhang, Xiaojian Ma, Wenjuan Han, Song-Chun Zhu, and Qing Li. Clova: A closed-loop visual assistant with tool usage and update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13258–13268, June 2024b.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *CoRR*, abs/2208.01626, 2022. URL https://doi.org/10.48550/arXiv.2208.01626.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *CoRR*, abs/2210.09276, 2022. URL https://doi.org/10.48550/arXiv.2210.09276.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Dongxu Li, Junnan Li, and Steven C. H. Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing, 2023.

Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1633–1642, 2019.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5164–5172, 2025.

Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pp. 89–106. Springer, 2020. URL https://doi.org/10.1007/978-3-030-58621-8_6.

Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018.

Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024.

Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. *arXiv preprint arXiv:2307.11410*, 2023.

Marcel Aguirre Mehlhorn, Andreas Richter, and Yuri AW Shardt. Ruling the operational boundaries: A survey on operational design domains of autonomous driving systems. *IFAC-PapersOnLine*, 56 (2):2202–2213, 2023.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection, 2024. URL https://arxiv.org/abs/2306.09683.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.

Alexander Quinn 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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16784–16804. PMLR, 2022. URL https://proceedings.mlr.press/v162/nichol22a.html.

Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models, 2024.

Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models, 2024. URL https://arxiv.org/abs/2403.12036.

Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2482–2491, 2018.

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 *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 2021. URL http://proceedings.mlr.press/v139/ramesh21a.html.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. URL https://doi.org/10.48550/arXiv.2204.06125.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01042.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510, June 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*. URL https://openreview.net/forum?id=08Yk-n5l2Al.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. URL https://arxiv.org/abs/2311.17042.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

Hao Shao, Yuxuan Hu, Letian Wang, Steven L. Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models, 2023.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023a.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks, 2023b. URL https://arxiv.org/abs/2311.10089.

Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024.

Shanlin Sun, Bingbing Zhuang, Ziyu Jiang, Buyu Liu, Xiaohui Xie, and Manmohan Chandraker. Lidarf: Delving into lidar for neural radiance field on street scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19563–19572, 2024.

Alexander Swerdlow, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird's-eye view layout. *IEEE Robotics and Automation Letters*, 2024.

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

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation, 2022.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18359–18369, 2023a.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Kairui Yang, Enhui Ma, Jibin Peng, Qing Guo, Di Lin, and Kaicheng Yu. Bevcontrol: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*, 2023a.

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

Manyi Yao, Bingbing Zhuang, Sparsh Garg, Amit Roy-Chowdhury, Christian Shelton, Manmohan Chandraker, and Abhishek Aich. ifinder: Structured zero-shot vision-based llm grounding for dash-cam video reasoning. *Advances in Neural Information Processing Systems*, 2025.

Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15459–15469, 2024a.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024b.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing, 2024c. URL https://arxiv.org/abs/2306.10012.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023a.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023b.

Bin Zhao, Xuelong Li, Xiaoqiang Lu, and Zhigang Wang. A cnn–rnn architecture for multi-label weather recognition. *Neurocomputing*, 322:47–57, 2018.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale, 2024. URL https://arxiv.org/abs/2407.05282.

A Related Works

Image Editing Dataset. Developing image-editing datasets is more challenging than Text-2-Image (Betker et al.; Ramesh et al., 2021; 2022; Rombach et al., 2022), with data scarcity being a major bottleneck (Wang et al., 2023a; Hui et al., 2024). MagicBrush (Zhang et al., 2024b) uses manual annotation with DALL-E2 (Ramesh et al., 2022), while InstructPix2Pix (Brooks et al., 2023a) generates pairs using the prompt-to-prompt method (Hertz et al., 2022) on LAION-Aesthetics (Schuhmann et al., 2022). SeedEdit (Shi et al., 2024) iteratively refines data and models. Most prior work targets simple object edits, underperforming in complex street scenes. We present *LangDriveEdit*, the first instruction-driven editing dataset for autonomous driving contexts.

Image Editing via Generation. Instruction-based editing of real photos is a key task in image processing (Gao et al., 2024b; Crowson et al., 2022; Liu et al., 2020; Zhang et al., 2023a; Ruiz et al., 2023; Pan et al., 2024). Large-scale diffusion models have greatly enhanced text-driven editing (Kawar et al., 2022; Saharia et al.; Li et al., 2023; Chen et al., 2023; Ma et al., 2023; Meng et al., 2022; Mokady et al., 2023; Tumanyan et al., 2022; Nichol et al., 2022; Sheynin et al., 2023a). Recent models like InstructPix2Pix (Brooks et al., 2023a) and HIVE (Zhang et al., 2023b) allow users to edit images via instructions. MagicBrush (Zhang et al., 2024b) enhances this with manual annotations, and UltraEdit (Zhao et al., 2024) sets a new benchmark using synthetic data. We show our dataset further boosts these methods in street-scene editing.

Image Editing for Autonomous Driving. Rising demand for driving-scene data has led to scene editing methods using NeRF or Gaussian Splatting (Liang et al., 2025; Yang et al., 2023b; Sun et al., 2024; Tonderski et al., 2024; Chen et al., 2024b), though these struggle with diverse scene composition. Meanwhile, multi-condition generation methods are gaining interest (Swerdlow et al., 2024; Yang et al., 2023a; Wang et al., 2023b; Gao et al., 2023; Wen et al., 2024; Alhaija et al., 2025; Gao et al., 2024a; Lu et al., 2024). Yet, instruction-guided image editing for driving remains underexplored due to a lack of datasets. We introduce *LangDriveEdit* to fill this gap.

B Additional Results

We conduct an qualitative ablation study at Figure 7. Specifically, when training solely with SFT (column 2) the model cannot preserve the details of the small cars and merges them together (row 1). Meanwhile, when we add unsupervised training objectives with the exception of the CLIP loss (column 3), the model outputs degenerate to match the input (row 2) especially for Nuscene. Using SFT, with unsupervised objectives, and CLIP similarity loss to prevent degeneration (column 3) shows the highest degree of instruction following (row 2) and content preservation (rows 1, 2).

C Synthetic Dataset Development

C.1 Precise Control of Scene Variations in Synthetic Environments

Our paired instructions and annotations on real images (Section 3.1) enable the learning of fine-grained edits across diverse objects and scenarios. Yet, real-world data remains limited in supporting arbitrary and precise manipulations. In contrast, synthetic environments allow flexible control over textures, semantics, and backgrounds, even for individual objects within a scene.

To provide complementary and precise control over scene variations, we additionally include paired images, annotations, and instructions from synthetic environments, as a supplement to our real-world images. We choose Carla (Dosovitskiy et al., 2017), an open-source simulator designed as a research and development platform for autonomous driving with extensive customization capabilities. With this simulator, we synthesize paired videos from the perspective of an autonomous vehicle across six diverse urban and suburban environments. Recent works demonstrate CARLA's utility in closed-loop end-to-end driving with language models (Shao et al., 2023), safety-critical scenario generation for autonomous vehicle testing (Zhang et al., 2024a), and multi-ability benchmarking of end-to-end driving systems (Jia et al., 2024).

Figure 7: **Ablation on Training Objectives**. Top: Boreas. Bottom: Nuscenes. We use blank masks during the evaluation of Boreas and study the content preservation abilities of the models over the vehicles.

Table 5: Comparison of edits performed by open-source driving scene editing datasets: Snow 100K (Liu et al., 2018), Outdoor-Rain (Out-Rain) (Li et al., 2019), Rain-Drop (Qian et al., 2018), MT Weather (Zhao et al., 2018), and *LangDriveEdit* (Ours) on Carla.

Category	Edit	Snow 100K	Out-Rain	Rain-Drop	MT Weather	Ours (Carla)
Global	Time of Day					✓
	Weather	✓	✓	✓	✓	✓
	Season	✓				✓
Road	Road Condition					✓
	Road Type					✓
Building	Building Appearance					✓
Vehicle	Vehicle Color					✓
	Vehicle Type					✓
	Vehicle Changes					✓
Traffic Signal	Traffic Light Color					✓
Pedestrian	Pedestrian Changes					✓
	Pedestrian Clothing					✓

C.1.1 Structured Scene Edits and Selections for Aligned Instruction

For each video pair, we implement controlled edits between the base and modified simulations, each spanning 30 seconds of driving. We list our edit types in Table 5. Specifically:

- The distribution and quantity of pedestrians and vehicles are randomly sampled to ensure diversity. Pedestrian edits are applied to all pedestrians in a scene, while the specific change (e.g., clothing style or color) is randomly sampled for each individual from the Carla pedestrian catalog. Text

descriptions of cataloged clothing were created through visual inspection. As of the time of this publication, we use the dev-branch of the Unreal Engine 4.26 version of CARLA and follow the pedestrian catalog at https://carla.readthedocs.io/en/latest/catalogue_pedestrians/.

- All vehicles receive the same type of modification, with random variations (e.g., color) applied individually.
- Weather conditions and time of day are selected from prepared Carla profiles to maintain realistic scenes.
- For environmental elements, we assign randomized textures to road surfaces and buildings, with each building receiving a distinct texture to enhance scene variability.

Each simulation captures comprehensive views through six ego-mounted cameras providing 360-degree coverage, all used in the dataset construction. The sensor suite records RGB images, depth maps, semantic segmentation maps, and instance segmentation maps for each frame. Following our workflow on real images (Section 3.1.2), we leverage these semantic maps to prepare precise masks of subjects targeted for editing; meanwhile, global environmental edits use blank masks since these changes apply to the entire scene. Additionally, we document per-frame attributes for all vehicles, pedetrians, and traffic signals visible to the ego vehicle, for precise tracking of object properties and behaviors.

C.1.2 Filtering and Balancing Frame Pairs for Content Preservation

To further improve the quality and diversity of edited images, we apply a multi-stage filtering process:

- Remove frames where edited subjects are too small or absent;
- Eliminate frames with high visual similarity based on SSIM thresholds and color histogram correlation;
- Discard redundant frames captured when the ego vehicle was stationary;
- For global modifications without object-specific edits (such as weather and time-of-day changes), we retain all non-redundant frames that pass our similarity thresholds based on SSIM and color histograms.

To ensure only relevant edits are kept, we filter out frames where the edited subjects are too small or absent. Specifically, a 2D bounding box must cover at least 0.7% of the image area for pedestrians, 1.2% for vehicles, and 0.3% for traffic lights. We also remove frames with high visual similarity to others, defined as SSIM > 0.97 or color histogram correlation > 0.5, to maintain scene diversity. We use a blank mask for global edits, such as changes in weather or time of day, since these modifications can affect all objects in the scene (e.g., through lighting changes). We show our synthetic dataset statistics in Table 6. To address class imbalance in our training data, we oversample underrepresented editing categories.

D Real Dataset Development

We generate: (1) a *removal mask* for the source image, which identifies objects to be eliminated, and (2) an *addition mask* for the target image, which highlights regions designated for new object placement. Together, they support fine-grained object edits while preserving non-targeted regions

These pixel-level editing masks are systematically derived by comparing corresponding frame annotations via three rules:

- Distance-based filtering: Objects beyond 50 meters from the ego vehicle are excluded unless they occupy a significant image area.
- Truncation detection for undersized 2D bounding boxes near image boundaries.
- Occlusion handling: In complex traffic scenarios, overlapping vehicle bounding boxes are each preserved to maintain scene coherence.

Table 6: **Synthetic Dataset Statistics**. Breakdown of samples across object-level and global environment editing types.

Edit Type	Count	Percentage
Object Editing (458,136 samples)		
Road Texture	153,654	7.69%
Building Texture	3,662	0.18%
Walker Color	28,726	1.44%
Walker Replacement	39,166	1.96%
Walker Deletion	34,460	1.72%
Vehicle Replacement	45,806	2.29%
Traffic Light State	24,050	1.20%
Vehicle Color	64,300	3.22%
Vehicle Deletion	64,312	3.22%
Global Environment Editing (1,539,370 samples)		
Weather	527,450	26.41%
Weather + Time of Day	549,612	27.51%
Time of Day	462,308	23.14%
Total Samples	1,997,506	100%

E Annotation Pipeline

E.1 Annotation Pipeline Prompt

The VLM is prompted with the instruction loaded from:

```
You are an expert in autonomous driving, specializing in analyzing traffic scenes. You receive a series of traffic images from the perspective of the ego car. Your task is to describe the driving environment, focusing on weather, lighting, road layout, and environment.

It is essential that you strictly follow the rules and instructions below. Any deviation from the specified structure or format will result in an invalid output.

STRICTLY follow Rules:
 - You must strictly follow the dictionary structure provided above.
 - Only use the specified terms for weather, light, road layout, and environment. Do not create your own terms.
 - No additional information or categories should be added.
 - You should strictly follow these instructions. If an object or element is not visible or does not exist in the scene, set the value to 'None'. Ensure every field is filled with the appropriate value or 'None'.
 - STRICTLY ignore any text written on the image.


Output the result in the following dictionary format:

{
  "surrounding_info": {
    "weather": "[e.g., 'cloudy', 'sunny', 'rainy', 'fog', 'snowy']",
    "road_layout": "[Choose from: 'straight road', 'curved road', 'intersection', 'T-junction', 'ramp']",
    "environment": "[Choose from: 'city street', 'country road', 'highway', 'residential area']",
    "sun_visibility_conditions": "[Choose from: 'clear', 'foggy', 'low visibility', 'hazy']",
    "road_condition": "[Choose from: 'wet', 'icy', 'normal', 'debris', 'potholes']",
```

```
    "surface_type": "[Choose from: 'asphalt', 'gravel', 'dirt', 'concrete']",
    "surface_color": "[Choose from: 'light grey', 'dark grey', 'black', 'brown']",
    "time_of_the_day": "[Choose from: 'morning', 'midday', 'afternoon', 'night', 'dawn', 'dusk'.]",
    "precipitation_intensity": "[Choose from: 'none', 'light', 'moderate', 'heavy', 'torrential'.]",
    "precipitation_visibility_impact": "[Choose from: 'none', 'low', 'moderate', 'high']",
    "cloud_cover": "[Choose from: 'clear', 'light', 'moderate', 'heavy'.]
    }
}
```

E.2 Example Annotation

We show an annotated image and the output caption from the annotation pipeline

Figure 8: Example annotation of the image by the annotation pipeline.

We present a truncated version of the annotation for 8 below:

```
{
    "surrounding_info": {
        "weather": "cloudy",
        "road_layout": "intersection",
        "environment": "city street",
        "sun_visibility_conditions": "low visibility",
        "road_condition": "normal",
```

```
        "surface_type": "asphalt",
        "surface_color": "dark grey",
        "time_of_the_day": "morning",
        "precipitation_intensity": "none",
        "precipitation_visibility_impact": "none",
        "cloud_cover": "heavy"
    },
    "object_info": [
        {
            "class": "building",
            "bbox": [
                1234,
                745,
                2060,
                1051
            ],
            "object_id": 1,
            "distance_from_ego_vehicle": "67.42 meters",
            "attributes": "Brown color, Brick texture, Apartment type"
        },
        {
            "class": "traffic light",
            "bbox": [
                779,
                963,
                790,
                986
            ],
            "object_id": 2,
            "distance_from_ego_vehicle": "78.61 meters",
            "attributes": "Red light"
        },
        {
            "class": "car",
            "bbox": [
                79,
                1025,
                147,
                1062
            ],
            "object_id": 10,
            "distance_from_ego_vehicle": "69.81 meters",
            "attributes": "White color, Not a police car."
        },
        {
            "class": "car",
            "bbox": [
                123,
                1027,
                249,
                1067
            ],
            "object_id": 11,
            "distance_from_ego_vehicle": "61.59 meters",
            "attributes": "White color, Not a police car."
        }
    ]
}
```

F LLM GENERATED EDITING INSTRUCTIONS

F.1 SYNTHETIC

We prompt chatGPT-4o mini with the following instruction to produce editing instructions for synthetic images based on the captions of the target image:

```
You are an expert in autonomous driving, specializing in analyzing traffic scenes. You receive a text description of a traffic image from the perspective of an autonomous vehicle's camera.

Your task is to produce FOUR VERSIONS of the SAME PROMPT, each with DIFFERENT WORDING BUT IDENTICAL CONTENT, that describes the driving scene depicted in the image.

IMPORTANT:
- Each version should describe the SAME SCENE, just phrased differently.
- Use natural, conversational language as if explaining the scene to another person.
- You may paraphrase, use synonyms, and vary sentence structure, but do not invent details not present in the caption.
- It is OK to combine or rephrase information for readability and flow.
- Avoid sounding like a computer or simply listing numbers and attributes. Make the description sound like something a human would say.
- Do not use quantitative descriptions. Only use qualitative natural language to describe the scene.
- Do NOT add your own subjective opinions or emotions (e.g., do not say 'beautiful', 'moody', etc.), but you may use natural transitions and phrasing.


The output should be in the format below:

### Scene Description:

version_1: {{description_1}}
version_2: {{description_2}}
version_3: {{description_3}}
version_4: {{description_4}}

Image Caption: {caption_0}
```

F.2 REAL-WORLD

We prompt chatGPT-4o mini with the following instruction to produce editing instructions for real-world images based on the captions of the target image:

```
You are an expert in autonomous driving, specializing in analyzing traffic scenes. You receive a text description of a traffic image from the perspective of an autonomous vehicle's camera.

Your task is to produce FOUR VERSIONS of the SAME PROMPT, each with DIFFERENT WORDING BUT IDENTICAL CONTENT, that describes the driving scene depicted in the image.

IMPORTANT:
- Each version should contain the EXACT SAME DESCRIPTION, just phrased differently.
- ALL prompts should describe EXACTLY THE SAME SCENE with no variation in what is being described.
- Only use adjectives and descriptors that are explicitly provided in the caption. Do NOT add your own subjective descriptors like "moody," "tranquil," "charming," etc. Stick strictly to the attributes and descriptors that appear in the input caption.
```

```
At the end of each prompt version, append the following line:
    "There may be minor additional changes in time or weather (such as
        lighting, clouds, or rain) between the images that are not fully
        captured by the descriptions, but these are expected to be subtle
        ."

The prompt should be in the format below where each version describes the
     same contents but with different wording.

### Scene Description:

version_1: {{description_1}}
version_2: {{description_2}}
version_3: {{description_3}}
version_4: {{description_4}}

Image Caption: {caption_0}
```

G Model and Evaluation Details

We evaluate our training methods and pixel level instructions on two competitive image editing models: UltraEdit (Zhao et al., 2024)) and CycleGAN-Turbo (Parmar et al., 2024). CycleGAN-Turbo, based on the Stable Diffusion Turbo by Stability AI (Sauer et al., 2023) performs text-instructed image generation in one diffusion step. UltraEdit is based on Stable Diffusion 3 which is fine-tuned on 500K dataset of free-form edits and an additional 100K dataset of precise mask-conditioned edits. All models are evaluated with the dataset described in Sec. 3.

Following image editing benchmarks used in (Zhang et al., 2024c; Zhao et al., 2024; Sheynin et al., 2023b), we consider the following metrics: the L1 and L2 distance, the CLIP image similarity, and the DINO image similarity between the edited image and the ground truth. These metrics measure how well the edited image preserves the original content and reflects the required edit. For each of the real-world and synthetic datasets, we evaluate 2000 images for editing independently in both directions: transforming the source image to match the target, and conversely, modifying the target to reproduce the source.

H Training Details

H.1 CycleGAN-Turbo

We train this model across 2×80GB NVIDIA A100 GPUs with a total batch size of 4 for 10000 steps. Our training parameters are:

$$\begin{aligned} \lambda_{\text{gan}} &= 0.5, \\ \lambda_{\text{id}} &= 0.05, \quad \lambda_{\text{id-lpips}} = 0.05, \\ \lambda_{\text{cycle}} &= 0.05, \quad \lambda_{\text{cycle-lpips}} = 0.05, \\ \lambda_{\text{sft}} &= 0.1, \quad \lambda_{\text{sft-lpips}} = 1.0, \\ \lambda_{\text{clip}} &= 0.5. \end{aligned} \tag{6}$$

To incorporate the guidance from the remove and add masks, we expand the VAE input channels to accept the concatenation of the input image and conditioning masks and train end-to-end. The weights of any existing convolutions are maintained and new weights are initialized as zero. We train at a resolution of 512×512 and a learning rate of 1×10^{-5}.

H.2 UltraEdit

We evaluate the UltraEdit model with the following settings:

- **UltraEdit**. The UltraEdit model supports a single binary mask as conditioning therefore we simplify our remove and add masks into one with the union of their binary projections.
- **UltraEdit-text**. We train an UltraEdit model using only supervised objectives on a modified real-world subset of the LangDriveEdit dataset without CLIP masks. To do this, following the process described in Sec. 3.1.2, we construct the editing prompts by asking chatGPT-4o to describe, in addition to global changes, all objects to remove from left to right, and all objects to add from left to right. The pixel add and remove masks are the binary image equivalents of the full add and remove masks. This model is trained across 4×48 GB NVIDIA A6000 GPUs with a total batch size of 256 for 10000 steps.
- **UltraEdit-clip**. We train an UltraEdit model using only supervised training objectives and the real-world dataset described in Sec. 3. We train this model across 4×80GB NVIDIA A100 GPUs with a total batch size of 4 for 5000 steps.
- **UltraEdit (ours)** We train an UltraEdit model using the objectives described in Sec. 4 on our real-world dataset described in Sec. 3 in addition to unsupervised objectives on NuScenes. To adapt these objectives to multi-step diffusion, we apply gradient checkpointing and perform end-to-end training with our unsupervised losses. We train this model across 4×80GB NVIDIA A100 GPUs with a total batch size of 4 for 5000 steps. Our training parameters are:

$$\begin{aligned} \lambda_{\text{gan}} &= 0.5, & & \\ \lambda_{\text{id}} &= 0.05, & \lambda_{\text{id-lpips}} &= 0.05, \\ \lambda_{\text{cycle}} &= 0.05, & \lambda_{\text{cycle-lpips}} &= 0.05, \\ \lambda_{\text{sft}} &= 3.0, & \lambda_{\text{sft-lpips}} &= 0.5, \\ \lambda_{\text{clip}} &= 0.5. & & \end{aligned} \tag{7}$$

H.3 Road Segmentation Model

We train the base model on a random quarter of the NuScenes dataset across 4×48GB NVIDIA A6000 GPUs with a total batch size of 16 for 10000 steps. We train for 20 epochs.

We train another model on our synthetic NuScenes dataset across 4×48GB NVIDIA A6000 GPUs with a total batch size of 16 for 10000 steps for 20 epochs. Then we finetune on the original quarter of the NuScenes dataset across 4×48GB NVIDIA A6000 GPUs with a total batch size of 16 for 10000 steps for 20 epochs.

I Generation Details

I.1 CycleGAN-Turbo

We maintain the default parameters from (Parmar et al., 2024) and evaluate our trained model with remove and add masks. The base model is evaluated without mask with the default parameters from (Parmar et al., 2024).

I.1.1 UltraEdit

- **UltraEdit**. The UltraEdit model supports a single binary mask as conditioning therefore we simplify our remove and add masks into one with the union of their binary projections. We maintain the default parameters from (Zhao et al., 2024).
- **UltraEdit-text**. We train an UltraEdit model using only supervised objectives on a modified real-world subset of the LangDriveEdit dataset with masks projected to binary images. We maintain the default parameters from (Zhao et al., 2024).
- **UltraEdit-clip**. We evaluate the model with full add and remove masks using 20 diffusion steps and classifier-free guidance scale of 1 and image guidance scale of 1.
- **UltraEdit (ours)** We evaluate the model with full add and remove masks using 8 diffusion steps and classifier-free guidance scale of 1 and image guidance scale of 1.