WeatherEdit: Controllable Weather Editing with 4D Gaussian Field

Chenghao Qian[1], Wenjing Li[1], Yuhu Guo[2], Gustav Markkula[1]
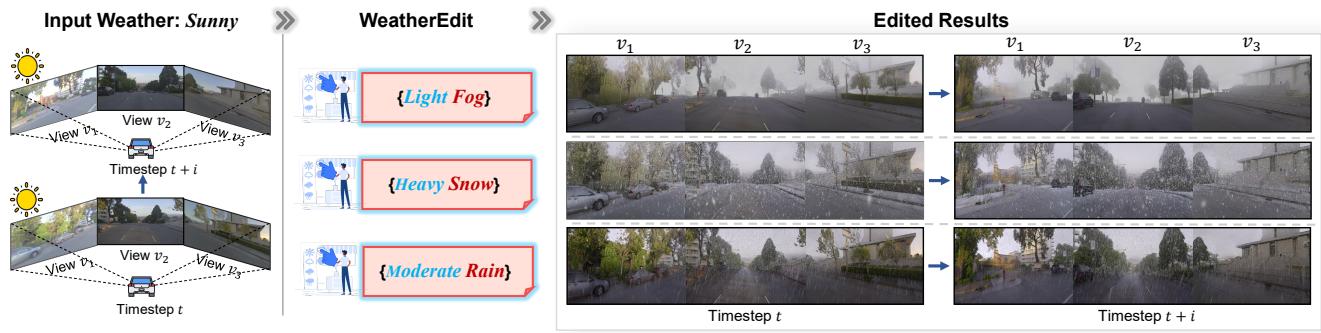[1]University of Leeds,
[2]Carnegie Mellon University

Figure 1. Realistic, Controllable Weather Editing. Given input sunny weather images from multi-view cameras across multiple timesteps, WeatherEdit can generate high-fidelity scenarios under diverse weather conditions (*e.g.*, foggy, snowy, and rainy) with various severity levels (*e.g.*, light, moderate, heavy) based on user instructions. The best view is in zoom.

Abstract

In this work, we present WeatherEdit, a novel weather editing pipeline for generating realistic weather effects with controllable types and severity in 3D scenes. Our approach is structured into two key components: weather background editing and weather particle construction. For weather background editing, we introduce an all-in-one adapter that integrates multiple weather styles into a single pretrained diffusion model, enabling the generation of diverse weather effects in 2D image backgrounds. During inference, we design a Temporal-View (TV-) attention mechanism that follows a specific order to aggregate temporal and spatial information, ensuring consistent editing across multi-frame and multi-view images. To construct the weather particles, we first reconstruct a 3D scene using the edited images and then introduce a dynamic 4D Gaussian field to generate snowflakes, raindrops and fog in the scene. The attributes and dynamics of these particles are precisely controlled through physical-based modelling and simulation, ensuring realistic weather representation and flexible severity adjustments. Finally, we integrate the 4D Gaussian field with the 3D scene to render consistent and highly realistic weather effects. Experiments on multiple driving datasets demonstrate that WeatherEdit can generate diverse weather effects with controllable condition severity, highlighting its potential for autonomous driving simulation in adverse weather. See project page: https://jumponthemoon.github.io/w-edit

1. Introduction

3D weather editing (3D-WE) aims to generate multi-weather conditions in 3D scenes based on user instructions. It is a crucial task in computer vision with applications in autonomous driving[26, 27], augmented reality, and virtual scene synthesis [1]. While existing methods range from 2D-based image techniques [36, 41] to 3D neural rendering [17], developing a unified framework that enables multiple realistic weather effects and controllable severity remains a significant challenge. For instance, style transfer methods can modify weather backgrounds but are typically limited to a single effect. Diffusion models [39] enable multi-weather synthesis; however, they often lack fine-grained control and struggle with spatial and temporal consistency across multiple images [30]. More recently, ClimateNeRF [17] utilized Neural Radiance Fields (NeRF) to model snow accumulation and flooding in 3D space. However, this method is limited to static weather events and leaves the overall weather tone unchanged, lacking full-scene atmospheric realism.

In 3D-WE, achieving both **realism** and **controllability** is important. Realism ensures high-quality weather effects

and we consider it involves three aspects: (R1) *Distortion-free editing*: The editing results should be free from visual distortions and unnatural artifacts. (R2) *Temporal-view consistency*: Weather effects and scene elements should maintain consistency across both time steps and different viewpoints. (R3) *Dynamic weather effects*: Weather particles, such as snowflakes and raindrops, should be realistically simulated to reflect natural weather dynamics. On the other hand, controllability ensures flexible and fine-grained editing of weather effects, enabling users to customize various aspects of the scene, including: (C1) *Type*: The system should support the selection and transition between different weather types (*e.g.*, rainy, snowy, foggy). (C2) *Event*: The system should allow users to specify whether the weather effect is static (e.g., previously fallen snow) or ongoing (e.g., falling snow). (C3) *Severity*: Users should be able to adjust the intensity of weather conditions, such as changing light rain to heavy rain or increasing fog density. Unfortunately, current approaches have not explicitly considered the above aspects, leading to limitations in realism and controllability.

In this paper, we consider the above points and introduce a novel weather editing framework named WeatherEdit that enables realistic, and controllable weather generation in 3D scenes. WeatherEdit follows a progressive 2D-to-4D transformation process, starting with 2D image-based background editing, transitioning to 3D scene reconstruction, and culminating in 4D dynamic weather simulation. This is achieved through a two-step process: **weather background editing** and **weather particle modeling**.

In weather background editing, we first introduce an efficient all-in-one adapter that integrates multiple weather styles into a single model, allowing for the fine-tuning of a diffusion model to achieve distortion-free rendering (R1) and flexible weather type control (C1). Second, to ensure temporal-view consistency (R2), we introduce a parameter-free attention mechanism that aligns different viewpoints simultaneously and the same viewpoint over time. For weather particle modeling, we propose a novel 4D Gaussian field, representing weather particles as 3D Gaussians with attributes such as color and density while simulating their motion. This enables dynamic weather effects (R3) and severity (C3). Rather than simulating an infinite field for full-scene coverage, we optimize efficiency by recycling particles within a compact field and aligning it with the rendering camera, maintaining both realism and performance.

The final weather effect is rendered from the 4D Gaussian field combined with the 3D scene reconstructed using edited weather background images. This flexible combination allows control over weather events, such as a wet road without rainfall (static) or a heavy snowstorm with limited visibility (dynamic), enabling adjustable event (C2).

The main points of this paper are summarized below.

1. Based on our analysis of weather editing characteristics, we introduce WeatherEdit, a comprehensive and efficient framework for realistic and controllable weather generation. Compared with existing methods which focus on either background editing or static weather effects, a progressive 2D-to-4D transformation process in WeatherEdit enhances adaptability across a wider range of scenarios.
2. We introduce an all-in-one adapter to enable a diffusion model for multi-weather (snowy, rainy and fog) synthesis and a Temporal-View attention to ensure consistent editing across multi-frame and multi-view.
3. We design a 4D Gaussian field for weather particle modeling, enabling plausible simulation of raindrops, snowflakes, and fog with controllable severity.
4. We demonstrate WeatherEdit's effectiveness in generating realistic, consistent, and controllable weather effects in 3D driving scenes, showcasing its applicability to real-world scenarios.

2. Related Work

Weather Effect Synthesis

Existing methods can be categorized into 2D-based and 3D-based approaches. 2D-based methods utilize generative models [24, 43] to synthesize weather effects in single images. [14, 33] use CycleGAN [43] to edit weather conditions, while [32] applies GAN-based inpainting to generate flooding with water reflections. [11] uses graphical model to produce fog effects, while [26] finetunes a diffusion model to various weather effects. Conventional 3D-based methods [8, 10, 34, 44] use graphical simulation to model weather in 3D space. More recently, [17] has introduced neural rendering techniques for creating weather backgrounds but it is limited to post-weather event.

Diffusion Model for Image Editing

Diffusion models [12, 21, 29] have significantly advanced image editing, enabling flexible and controllable modifications. Methods such as ControlNet [42] and T2I-Adapter [23] introduce trainable modules to guide the generation process, allowing for fine-grained control over diffusion-based outputs. InstructPix2Pix [4] enables interactive editing with textual instructions by trained on image-text pairs. TurboEdit [7] employs a pseudo-guidance approach to enhance edit magnitude without introducing artifacts.

Although these methods support diverse styles, they are not specifically designed for weather editing, often distorting content or generating overly artistic effects. Moreover, they primarily focus on single-image editing, lacking spatial and temporal consistency across multiple frames and views. These limitations restrict their ability to generate realistic and consistent weather effects within the same scene.
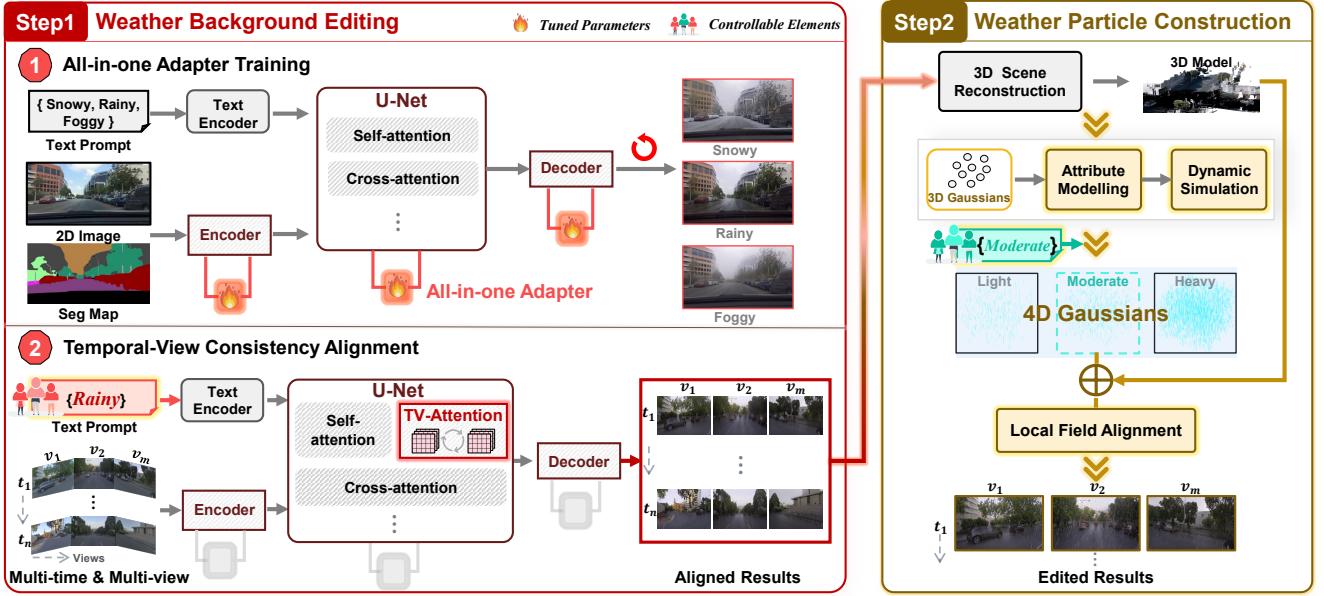
Figure 2. **Overview of the WeatherEdit Framework.** (a) Weather Background Editing: An all-in-one adapter enables the synthesis of diverse weather effects (e.g., snow, rain, fog) from text prompts and segmentation maps. Inference takes multi-frame and multi-view images as input, using temporal-view (TV) attention to ensure consistency.(b) Weather Particle Construction: A 4D Gaussian field, initialized from 3D Gaussians, undergoes attribute modeling and dynamic simulation, allowing user-controlled weather intensity (e.g., light, moderate, heavy). Local field alignment ensures seamless integration of the reconstructed 3D scene with dynamic weather effects.

3D Scene Reconstruction.

Neural Radiance Fields (NeRF) [22]and 3D Gaussian Splatting (3DGS) [15] are leading techniques for 3D scene reconstruction from multi-view images. NeRF encodes scenes implicitly with a neural network, while 3DGS represents them explicitly as 3D Gaussians, enabling faster rendering. Their versatility allows applications in dynamic scene reconstruction [6, 38], content creation [19, 20], and artifact removal [25]. While [17] extends NeRF to model smog, snow, and floods for weather simulation but focuses only on static weather conditions, leaving dynamic weather phenomena unaddressed.

3. Methods

In order to generate realistic and controllable multi-weather scenes in 3D, we propose a novel framework, WeatherEdit, which follows a progressive 2D-to-4D transformation process. As illustrated in Fig. 2, WeatherEdit adopts a two-step weather editing pipeline: weather background editing and weather particle construction. The first step generates distortion-free, spatially and temporally aligned 2D images with edited backgrounds, while the second step dynamically models weather particles with various types and severity levels. The joint association of these two steps enables the flexible generation of multi-weather scenes.

(a) Single Adapter   (b) All-in-one Adapter

Figure 3. **Single Adapter vs. All-in-One Adapter.** (a) Utilizes separate adapters for each weather style (e.g., snow, rain, fog), while (b) employs a unified all-in-one adapter for efficient multi-style adaptation. Blue box highlights adapters to be trained.

3.1. Weather Background Editing

All-in-one Adapter. Existing diffusion-based methods often suffer from scene deformations and excessive stylization, making them unsuitable for precise and realistic weather editing. While [24] achieves effective weather editing by fine-tuning separate models with multiple LoRA [13] adapters (shown in Fig. 3 (a)) for different styles, it requires multiple retraining steps, leading to inefficiencies in storage and training time.

To address this, we propose an all-in-one LoRA adapter that injects multiple weather styles into a single model (as shown in Fig. 3 (b),), eliminating the need for training multiple ones. Let $W_0 \in \mathbb{R}^{d \times k}$ be the pre-trained weights, x the input image, and i the weather style (*e.g.*, snowy, rainy, foggy) specified by text prompt. Each style is associated
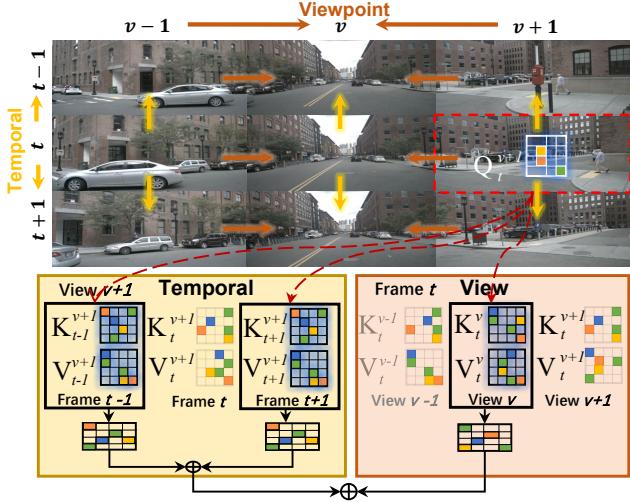
Figure 4. **Temporal-View Attention for Consistent Weather Editing.** The TV-attention aggregates information across both adjacent frames and center viewpoints to ensure multi-frame and multi-view consistency. The top section illustrates frame and viewpoint relationships, indicated by directional arrows, while the bottom section visualizes the attention operations.

with a low-rank matrix L_i. The adapted model's forward pass is then computed as $h = W_0 x + L_i x$.

Moreover, since weather effects depend on scene semantics (e.g., snow on trees, rain on roads), we enhance editing by conditioning the model on semantic segmentation maps. Given an input image x with its corresponding segmentation map M, we encode the latent representation as $\mathbf{z}_0 = \mathcal{E}(x, M)$. The model fine-tuning objective is then formulated as:

$$\mathcal{L}(\Delta\theta) = \mathbb{E}_{\epsilon,t} \left[\|\epsilon - \epsilon_{\theta+\Delta\theta}(\sqrt{a_t}\mathbf{z}_0 + \sqrt{1-a_t}\epsilon, t, p)\|^2 \right], \tag{1}$$

where ϵ is the sampled Gaussian noise, $\epsilon_{\theta+\Delta\theta}$ is the UNet model with LoRA, and p is the text prompt. This formulation ensures editing results are semantically relevant, enabling precise and context-aware weather editing.

Temporal-View Consistency Allignment. After fine-tuning the model, multi-time and multi-view images are processed to achieve background scene editing. However, processing each image independently can lead to inconsistencies across time steps and viewpoints due to the inherent stochasticity of generative models. To address this, we propose a Temporal-View (TV-) attention mechanism to enable our model to integrate information from adjacent frames and key viewpoints.

In multi-view driving datasets, cameras are typically positioned in a structured left-front-right arrangement. The front camera captures overlapping content with both the left and right cameras, providing shared information across views. To leverage this, we design the view attention to al-

low the left and right cameras to query the center (front) view:

$$\text{S\_Attn}_t^v = \text{Softmax}\left(\frac{Q_t^v \cdot K_t^{v,T}}{\sqrt{d}}\right) V_t^v, \tag{2}$$

where Q_t^v is the query from view v, and K_t^v, V_t^v are the key and value derived from the center view v. This ensures that each side viewpoint aggregates information from the center view, maintaining result consistency across different perspectives.

To preserve temporal coherence, each frame t queries its adjacent frames ($t-1$ and $t+1$), allowing it to integrate contextual information from both past and future frames. The temporal attention is defined as:

$$\text{T\_Attn}_t^v = \text{Softmax}\left(\frac{Q_t^v \cdot K_v^{\mathcal{T}(t),T}}{\sqrt{d}}\right) V_v^{\mathcal{T}(t)}, \tag{3}$$

where $\mathcal{T}(t) = \{t-1, t+1\}$, ensuring that each frame incorporates temporal context, preventing inconsistencies across consecutive frames. The final attention is computed as a weighted sum of self-, view-, and temporal-attention, with λ controlling the balance to ensure global coherence across frames and viewpoints:

$$\text{Attn}_t^v = \lambda \cdot \text{SelfAttn}_t^v + (1-\lambda) \cdot (\text{S\_Attn}_t^v + \text{T\_Attn}_t^v). \tag{4}$$

In this way, we can effectively reduce inconsistencies caused by independent image processing, producing a coherent editing effect across both view and temporal domains.

3.2. Weather Particle Construction

Real-world weather is inherently regional, and we incorporate this into weather particle construction. Specifically, we propose a *4D Gaussian field*, initialized with 3D Gaussians carrying physical attributes (*e.g.*, color, quantity) to model raindrops, fog, and snowflakes through attribute modeling. To enhance realism, we apply dynamic simulation to create realistic weather effects.

To generate a consistent falling particle effect, a large number of particles must be simulated, which is computationally expensive. To address this, we model particle behavior within a localized field, dynamically recycling particles as they exit while aligning the field with the rendering camera. This ensures a continuous and realistic weather effect while optimizing computational efficiency.

Attribute Modeling. Different weather particles exhibit unique characteristics. For instance, snowflakes are typically white with irregular shapes, while raindrops are transparent and elongated. To accurately capture these differences, we define a set of characteristic attributes A to
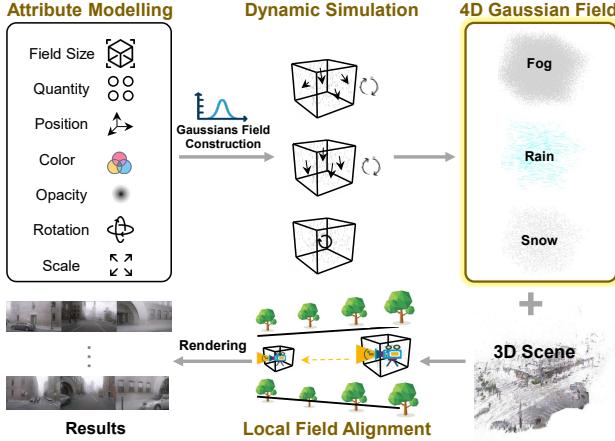
Figure 5. **Weather Particle Construction Process.** The attribute modeling stage defines weather properties to construct a 4D Gaussian field, which then undergoes dynamic simulation to model fog, rain, and snow. Local field alignment ensures seamless integration with the 3D scene, enabling realistic weather rendering.

distinguish different types of weather particles: $A_i = \{C_i, P_i, R_i, S_i, O_i\}$, where, C_i defines color, P_i represents position, R_i and S_i correspond to rotation and scale, and O_i signifies opacity. The index i refers to a specific type of weather, such as snow, rain, or fog. Additionally, we model these parameters using Gaussian distributions, expressed as $\mathbb{G}(A_i)$, to mimic the natural randomness of real-world weather. To compose the 4D Gaussian field, we use B_i to define the particle field size and combine it with quantity q, which controls the number of weather particles to regulate density. The final attributes of the 4D Gaussian field, are denoted as: $W_i = \{\mathbb{G}(A_i), q, B_i\}$.

Dynamic Simluation. Real-world weather is dynamic, with particles falling under gravity and wind. Instead of simulating these forces in detail, we approximate motion using a constant directional velocity, updating each particle's position as follows:

$$P_j(t + \Delta t) = P_j(t) + D_j \cdot \Delta t, \qquad (5)$$

where

$$P_j(t) = \left[x(t), y(t), z(t)\right]^\top, D_j = \left[D_x, D_y, D_z\right]^\top \quad (6)$$

represent the position and velocity of the j-th particle, respectively. Here, $x(t), y(t), z(t)$ denote the spatial coordinates of the particle at time t, while D_x, D_y, D_z are its velocity components along the x, y, and z axes.

To efficiently simulate a consistent falling particle effect, we recycle particles when they exit the predefined simulation field by resetting their positions. The position of the i-th particle, $P_i(t)$, is updated as follows:

$$P_i(t) = \begin{cases} p_{\max} - \delta_p, & \text{if } p_i(t) < p_{\min}, \\ p_{\min} + \delta_p, & \text{if } p_i(t) > p_{\max}, \\ P_i(t), & \text{otherwise.} \end{cases} \qquad (7)$$

Here, $p_{\min}$ and $p_{\max}$ denotes the field's lower and upper bounds, while δ_p represents the offset to the field bounds.

Local Field Alignment. After obtaining the reconstructed 3D scene and simulated 4D Gaussian field, we combine them to render the final weather scene. In large-scale environments, the weather field must scale appropriately to ensure a consistent weather effect. However, as scene size increases, more particles are required, leading to higher computational costs. To mitigate this, we constrain the size of the 4D Gaussian field and limit the number of particles, aligning its motion with the rendering camera. This keeps the field relatively stationary to the camera while preserving internal dynamics and intended weather effects. Given camera poses $\mathbf{T}_0$ at t_0 and $\mathbf{T}_t$ at t, the relative transformation is:

$$\Delta\mathbf{T} = \mathbf{T}_t\mathbf{T}_0^{-1}. \qquad (8)$$

The particle's position, initially $P_j(t_0)$, updates as:

$$P_j(t) = \Delta\mathbf{R}P_j(t_0) + \Delta\mathbf{t}, \qquad (9)$$

where $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$ are the rotation and translation from $\Delta\mathbf{T}$. In this way, we can ensure consistent weather dynamics in large-scale scenes with minimal computational cost.

4. Experiments

4.1. Experimental Setup

Dataset. To develop the weather dataset for all-in-one adapter fine-tuning, we extract 1,237 image pairs from the BDD100K [40], MUSE [3], and ACDC [31] datasets. Each pair contains a normal-weather image and an adverse-weather image, including snowy, rainy, and foggy conditions, along with a corresponding ground-truth semantic segmentation map. For weather editing, we selected eight scenes with multi-view image sequences from the Pandaset [37], Waymo Open Dataset [35], nuScenes [5], and KITTI-360 [18] datasets.

Implementation details. We implement the 2D editing model using CycleGAN-Turbo[24] and train on a single RTX A6000. The 4D Gaussian field and 3D scene reconstruction is based on OmniRe[6] and 3DGS[15], with both training and rendering performed on a single RTX 3090.

Evaluation Metrics. To evaluate edited weather background quality, we use the cosine similarity of CLIP [28] image embeddings (CLIP-S) to measure content preservation and the directional CLIP similarity [9] (CLIP-DS) to assess alignment with text instructions. To assess consistency across temporal and spatial dimensions, we calculate
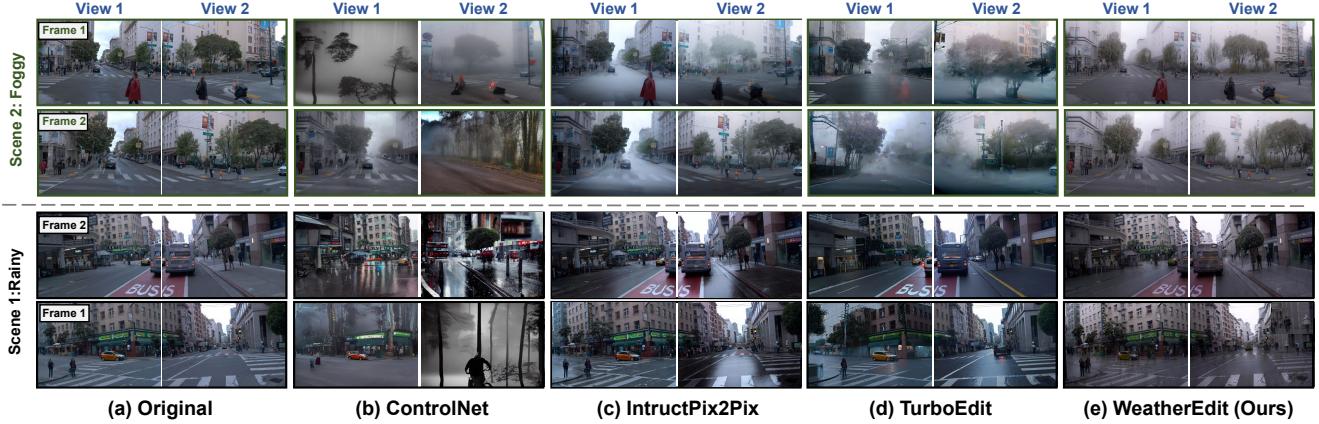
Figure 6. **Comparison with 2D-based Editing Methods.** We show (a) original images with weather editing results from (b) ControlNet, (c) InstructPix2Pix, (d) TurboEdit, and (e) Ours. Existing methods often suffer from over-stylization, content removal, and inconsistencies in spatial and temporal coherence. In contrast, our method preserves scene integrity while ensuring multi-view and multi-frame consistency, producing realistic and temporally coherent weather effects across viewpoints and frames.

Figure 7. **Comparison with 3D-based weather synthesis methods in editing *snowy*.** 3D Stylization (b) produces only a subtle snowy effect on the sidewalk, while (c) ClimateNeRF present unnatural snow cover. In contrast, our method (d) synthesizes a realistic snowy effect while ensuring a perceptually convincing atmosphere.

the warp error to measure temporal coherence between adjacent frames and employ the Bhattacharyya distance [2] to quantify color distribution similarity across viewpoints.

4.2. Experimental Results

We present both qualitative and quantitative comparisons of our weather background editing method against 2D-based approaches, including ControlNet [42], InstructPix2Pix [4], and TurboEdit [7], as well as 3D-based methods, such as 3D stylization [16] and ClimateNeRF [17]. For dynamic weather effects, we apply qualitative analysis, *i.e.,* controllable weather simulation, to demonstrate the results.

Weather Background Editing Comparison. For the comparison of 2D-based methods, as shown in Fig. 6, we observe that ControlNet tends to over-stylize the image, altering its original content. InstructPix2Pix produces a reasonable effect; however, the effect is inconsistent across adjacent frames and viewpoints. While TurboEdit accurately stylizes the image, it significantly modifies the scene content (e.g., turning pedestrians into water reflections and trees in a foggy scene). In contrast, our method achieves a realistic and coherent weather transformation while preserving scene integrity, as also indicated by the highest CLIP-DS and CLIP-S scores shown in Tab. 1.

For 3D-based methods, 3D stylization produces only a

	ControlNet	InstructPix2Pix	TurboEdit	Ours
CLIP-DS Score ↑	0.2905	0.2953	0.2573	**0.3023**
CLIP-S Score ↑	0.5944	0.6908	0.6762	**0.7495**

Table 1. Comparison with 2D-based weather background editing.

	3D Stylization	ClimateNeRF	Ours
CLIP-DS Score ↑	0.2253	0.2674	**0.2834**
CLIP-S Score ↑	**0.8417**	0.7022	0.7513

Table 2. Comparison with 3D-based weather background editing.

barely perceptible snowy effect on the vegetation, aligning with the highest CLIP-S score(shown in Tab. 2), which indicates minimal change in the scene. In contrast, ClimateNeRF generates an unrealistic snowcover on the road surface due inaccurate scene geometry. Furthermore, neither method is capable of modifying the overall scene tone to accurately reflect the intended weather conditions. In contrast, our approach not only synthesizes a realistic snowy effect but also adjusts the scene tone to create a coherent and perceptually convincing snowy atmosphere, as also indicated by the highest CLIP-DS score in Tab. 2.

Controllable Weather Simulation. A key advantage of our method over existing approaches is its ability to generate

Pandaset **Nuscenes** **Waymo**

(a) Original Driving Scene

Snowy *Foggy* *Rainy*

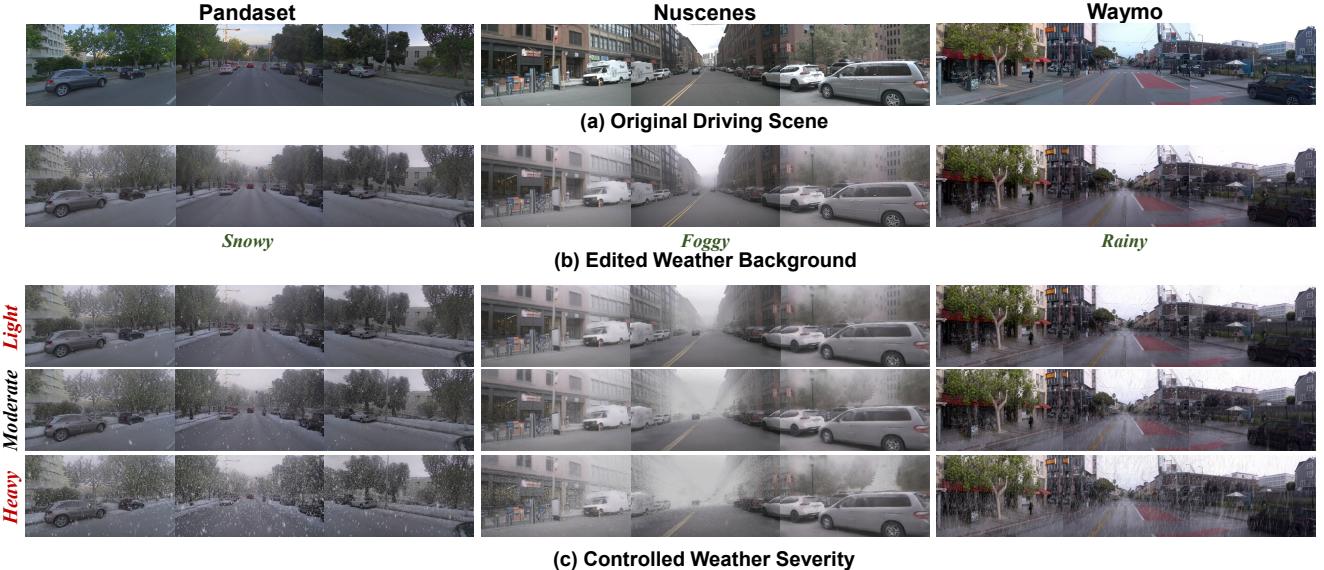**(b) Edited Weather Background**

(c) Controlled Weather Severity

Figure 8. **Controllable Weather Effect and Severity in Driving Scenes.** WeatherEdit can modify (a) original driving scenes from Pandaset, Nuscenes, and Waymo datasets by generating (b) snowy, foggy, and rainy effects while preserving scene structure. Additionally, it enables (c) controllable weather severity, adjusting precipitation intensity from light to heavy for realistic and flexible weather simulation.

dynamic weather effects while precisely controlling both the type and severity of weather conditions. Additionally, we can flexibly choose whether to introduce weather particles and, if included, adjust their intensity with fine granularity. This allows for a range of scenarios, from a static wet road surface without active rainfall to a heavy snowstorm with dense snowfall obscuring visibility. To visualize the results, we render edited weather backgrounds and modulate weather severity. The weather backgrounds, depicted in Fig. 8b, are reconstructed from 2D editing outputs, with the specific weather type determined by the selected adapters during the editing phase. Following this, weather particles are introduced by applying the 4D Gaussian field, as illustrated in Figure Fig. 8c. By adjusting particle attributes such as color and opacity, we can simulate various weather particles, including snowflakes, raindrops and fog. Moreover, by controlling particle quantity, shape, rotation, and velocity, we can achieve different levels of severity—ranging from light to moderate to heavy precipitation. Our approach enables highly flexible and physics-consistent scene manipulation, allowing for the realistic simulation of diverse environmental conditions.

5. Ablation Studies

We conduct ablation studies on the semantic condition input for finetuning, temporal-view attention for image editing and modular design in weather particle construction.

Semantic Conditioned Input. We investigate the effectiveness of semantic-conditioned input by comparing editing results and FID score with and without it. By computing the

	Self-attention only	w View	w Temporal	Full
Warp-error ↓	0.043	0.042	**0.039**	0.041
Bhattacharyya distance ↓	0.272	0.253	0.261	**0.245**

Table 3. Ablation on Temporal-View Attention.

cross-attention map using "snowy" in the text prompt, we observe that semantic conditioning enhances the model's response to sidewalks and trees, aligning weather effects more accurately with scene structures, as shown in Fig. 11. Additionally, the model achieves a lower average FID value across different weather conditions compared to cases without semantic conditioning, demonstrating its effectiveness in capturing weather characteristics during training.

Temporal-View Attention. As shown in Fig. 10, without TV-Attention, view 3 exhibits inconsistent editing effects, with a darker tone across two frames compared to view 2. After incorporating temporal-view attention, view 3 achieves more consistent editing, displaying a similar snowy effect between adjacent views and frames. For the qualitative analysis shown in Tab. 3, temporal attention reduces warp error, indicating improved temporal coherence, while spatial attention results in a lower Bhattacharyya distance, suggesting more consistent editing across viewpoints. By integrating both temporal and view attention, the full implementation achieves the lowest Bhattacharyya distance while maintaining a balanced reduction in warp error, ensuring globally consistent editing results.

Modular Design for 4D Gaussian Field. As shown in Fig. 9, without 2D background editing (b), the scene con-

(a) Original Scene   **(b) w/o Weather Background Editing**   **(c) w/o Attribute Modelling**

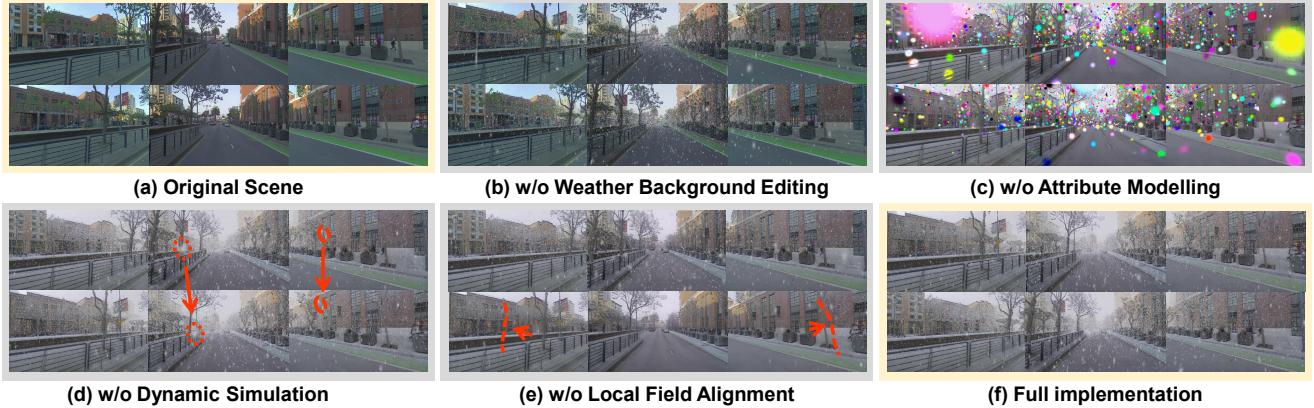**(d) w/o Dynamic Simulation**   **(e) w/o Local Field Alignment**   **(f) Full implementation**

Figure 9. **Ablation Study on WeatherEdit modules.** (b) Without Weather Background Editing, the scene lacks atmospheric tone. (c) Without Attribute Modeling, the scene presents unrealistic, blobs instead of proper weather effects. (d) Without Dynamic Simulation, particles remain static in the camera view (highlighted in red circle), lacking natural motion. (e) Without Local Field Alignment, weather particles stay fixed at their initial position and disappear as the camera moves forward (highlighted in red line). (f) Full Implementation generates a realistic weather background with natural, scene-consistent weather dynamics.

(a) w/o TV-Attention
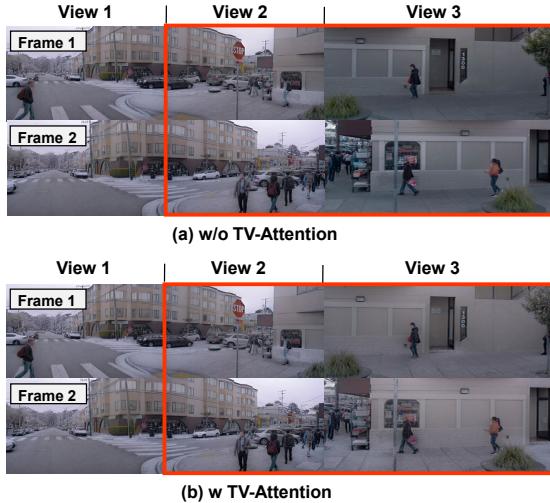

(b) w TV-Attention

Figure 10. **Effect of Temporal-View (TV) Attention on Consistency.** Without TV-Attention (a), inconsistencies appear across frames and viewpoints, as highlighted in the red box. With TV-Attention, weather effects maintain consistent scene appearance across frames and viewpoints.

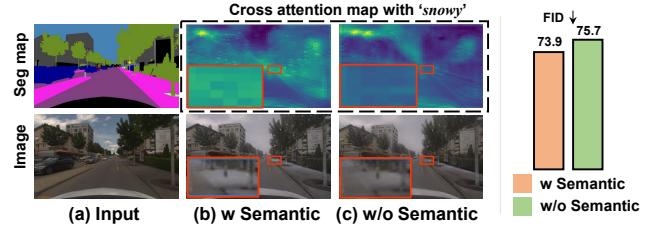**(a) Input**   **(b) w Semantic**   **(c) w/o Semantic**

Figure 11. **Ablation on semantic conditioned input.** Left (a) shows the input image paired with a segmentation map. With semantic conditioning (b), snowfall presents more prominent feature (highlighted in red box) and aligns better with scene structures compared to (c) without it, while cross-attention maps highlight improved view awareness.

tains weather particles but lacks an overall weather tone. During attribute modeling, we randomly initialize 4D Gaussians, resulting in a scene filled with colorful blobs (c) instead of realistic weather effects. Without dynamic simulation (d), the particles remain relatively static in the camera view, lacking the natural falling motion. Without local field alignment (e), weather particles stay fixed at the initial position of the scene. As the camera moves forward, the weather particles gradually disappear instead of persisting throughout the scene. In contrast, when all modules are included, the full implementation (f) produces a scene with a realistic weather background and natural weather dynamics.

6. Conclusion

We introduce WeatherEdit, a novel 4D weather editing framework capable of generating realistic weather effects—including fog, rain, and snow—with precise control over their severity. Our approach leverages an all-in-one adapter that enables a diffusion model to transform normal-weather images into diverse weather styles. To maintain consistency of edited results across multiple frames and viewpoints, we introduce an ordered Temporal-View (TV) Attention mechanism. Additionally, by designing a 4D Gaussian field, we model the attributes and dynamics of weather particles within a 3D scene, enhancing realism and enabling fine-grained control over weather severity. WeatherEdit demonstrates its potential in generating extreme weather conditions from normal scenes, providing a foundation for future research on evaluating the resilience of real-world applications such as autonomous driving.

References

[1] Yasin Almalioglu, Mehmet Turan, Niki Trigoni, and Andrew Markham. Deep learning-based robust positioning for all-weather autonomous driving. *Nature Machine Intelligence*, 4(9):749–760, 2022. 1

[2] Anil Kumar Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943. 6

[3] Tim Brödermann, David Bruggemann, Christos Sakaridis, Kevin Ta, Odysseas Liagouris, Jason Corkill, and Luc Van Gool. Muses: The multi-sensor semantic perception dataset for driving under uncertainty. In *European Conference on Computer Vision (ECCV)*, 2024. 5

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402, 2023. 2, 6

[5] 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 multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[6] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction. In *The Thirteenth International Conference on Learning Representations*, 2025. 3, 5

[7] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models, 2024. 2, 6

[8] Bryan E Feldman and James F O'Brien. Modeling the accumulation of wind-driven snow. In *ACM SIGGRAPH 2002 conference abstracts and applications*, pages 218–218, 2002. 2

[9] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 5

[10] John K Haas. A history of the unity game engine. 2014. 2

[11] Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaech, and Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3675–3681. IEEE, 2019. 2

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[14] Sunhee Hwang, Seogkyu Jeon, Yu-Seung Ma, and Hyeran Byun. Weathergan: Unsupervised multi-weather image-to-image translation via single content-preserving uresnet generator. *Multimedia Tools and Applications*, 81(28):40269–40288, 2022. 2

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 3, 5

[16] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 453–468, 2018. 6

[17] Yuan Li, Zhi-Hao Lin, David Forsyth, Jia-Bin Huang, and Shenlong Wang. Climatenerf: Extreme weather synthesis in neural radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 3, 6

[18] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 5

[19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 300–309, 2023. 3

[20] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6657, 2024. 3

[21] 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. *arXiv preprint arXiv:2108.01073*, 2021. 2

[22] 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 *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, page 405–421, Berlin, Heidelberg, 2020. Springer-Verlag. 3

[23] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2024. 2

[24] Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024. 2, 3, 5

[25] Chenghao Qian, Yuhu Guo, Wenjing Li, and Gustav Markkula. Weathergs: 3d scene reconstruction in adverse weather conditions via gaussian splatting. In *International Conference on Robotics and Automation(ICRA)*, 2025. 3

[26] Chenghao Qian, Yuhu Guo, Yuhong Mo, and Wenjing Li. Weatherdg: Llm-assisted procedural weather generation for domain-generalized semantic segmentation. *IEEE Robotics and Automation Letters*, 2025. 1, 2

[27] Chenghao Qian, Mahdi Rezaei, Saeed Anwar, Wenjing Li, Tanveer Hussain, Mohsen Azarmi, and Wei Wang. Allweather-net: Unified image enhancement for autonomous driving under adverse weather and low-light conditions. In *International Conference on Pattern Recognition*, pages 151–166. Springer, 2025. 1

[28] 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*, pages 8748–8763. PmLR, 2021. 5

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[30] Thomas Rothmeier and Werner Huber. Let it snow: On the synthesis of adverse weather image data. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3300–3306, 2021. 1

[31] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10765–10775, 2021. 5

[32] Victor Schmidt, Alexandra Luccioni, Mélisande Teng, Tianyu Zhang, Alexia Reynaud, Sunand Raghupathi, Gautier Cosne, Adrien Juraver, Vahe Vardanyan, Alex Hernández-García, et al. Climategan: Raising climate change awareness by generating images of floods. In *International Conference on Learning Representations*. 2

[33] Victor Schmidt, Alexandra Luccioni, S Karthik Mukkavilli, Narmada Balasooriya, Kris Sankaran, Jennifer Chayes, and Yoshua Bengio. Visualizing the consequences of climate change using cycle-consistent adversarial networks. *arXiv preprint arXiv:1905.03709*, 2019. 2

[34] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4): 1–10, 2013. 2

[35] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[36] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2353–2363, 2022. 1

[37] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE international intelligent transportation systems conference (ITSC)*, pages 3095–3101. IEEE, 2021. 5

[38] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 3

[39] Yijun Yang, Hongtao Wu, Angelica I. Aviles-Rivero, Yulun Zhang, Jing Qin, and Lei Zhu. Genuine knowledge from practice: Diffusion test-time adaptation for video adverse weather removal. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25606–25616, 2024. 1

[40] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 5

[41] Howard Zhang, Yunhao Ba, Ethan Yang, Varan Mehra, Blake Gella, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Alex Wong, and Achuta Kadambi. Weatherstream: Light transport automation of single image deweathering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13499–13509, 2023. 1

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847, 2023. 2, 6

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

[44] Károly Zsolnai-Fehér. The flow from simulation to reality. *nature physics*, 18(11):1260–1261, 2022. 2