WeatherCity: Urban Scene Reconstruction with Controllable Multi-Weather Transformation

Supplementary Material

Abstract

In the supplementary material, we present additional implementation details (Sec. 1), including training (Sec. 1.1), baselines (Sec. 1.3) and evaluation (Sec. 1.4). We also provide further experimental results and analysis (Sec. 2), including detailed quantitative and qualitative results (Sec. 2.1), temporal consistency comparison (Sec. 2.2), and 3D baseline comparisons (Sec. 2.3)

1. Implementation Details

We build WeatherCity upon a dynamic Gaussian scene graph following the node design in OmniRe [1], containing a sky node, a static background node, and multiple rigid and non-rigid object nodes for vehicles and pedestrians, respectively, each represented by 3D Gaussian primitives with learnable position, scale, rotation, opacity, and a shared appearance feature vector. For each weather condition, a lightweight weather-specific MLP with two fully connected layers, ReLU activation, and a final Sigmoid layer maps the shared feature of every Gaussian to its RGB color, yielding a set of weather-dependent Gaussians while keeping the underlying geometry shared across all conditions.

1.1. Training Details

Parameter setting. We optimize all scene nodes jointly for 30,000 iterations using Adam, while adopting node-specific learning rates to stabilize training for different motion patterns. The rotation parameters of Gaussian nodes are trained with a learning rate of 5×10^{-5} for non-rigid nodes and 1×10^{-5} for all other nodes. All other scalar parameters, including shared features and weather-decoder weights, use a base learning rate of 1×10^{-4}. All Gaussian densification operations are driven by the absolute gradient of the Gaussian parameters [10] with a densification threshold of 3×10^{-4}; and the scaling threshold for pruning is set to 3×10^{-3}. The shared Gaussian feature dimension is fixed to 32 with the hidden layer dimension equal to 64. During training, we randomly sample both raw clear-weather images and edited images of different weather types as supervision, and render corresponding views from the Gaussian representation to compute reconstruction losses, while all Gaussians (scene and weather particles) are rasterized with the standard 3D Gaussian Splatting pipeline.

Weather Particle Simulation. For dynamic weather effects, we instantiate dedicated Gaussian nodes for rain and snow inside a scene-aligned 3D bounding volume, and treat each particle as an elongated or compact Gaussian primitive that is jointly rasterized with the reconstructed scene. In the rainy setting, we sample 40,000 raindrop particles whose base color is fixed to $c_{\text{rain}} = [0.7, 0.7, 0.8]$, with scale initialized to $[0.0025, 0.0025, 0.075]$ and opacity set to 0.13, which produces thin, semi-transparent streaks aligned with the velocity direction. For snow, we use 16,000 particles with a brighter color $c_{\text{snow}} = [0.9, 0.9, 0.95]$, an anisotropic scale of $[0.0064, 0.004, 0.004]$, and opacity 0.2, leading to denser and more softly visible flakes that exhibit fluttering motion under the turbulence term. Fog is modeled as a global depth-dependent medium using the Beer–Lambert formulation, where the transmittance is parameterized by density $d_f = 0.2$ and the global fog color is set to $c_{\text{fog}} = [0.8, 0.8, 0.85]$, enabling continuous control of visibility and color tone by adjusting d_f and c_{fog}.

Prompt design. In all experiments, we use identical text instructions for Qwen-Image and all baselines to ensure a fair comparison. As shown in Table 1, for each target weather condition, we design a structured prompt that separately specifies: (1) strict preservation of the original layout and style, (2) the desired visual properties of the target weather, and (3) prohibited artifacts.

The prompts enforce consistent content preservation, including camera composition, object categories, and spatial arrangement, so the models focus solely on modifying global atmospheric conditions.

For rain, the prompt specifies an overcast sky, wet roads with puddles and reflections, and cool, dim lighting, while forbidding sunlight, dry ground, and hallucinated objects. For snow, it additionally removes all but the designated white and red vehicles, converts foliage to snow-covered bare branches, and requires overcast lighting with falling snow, without introducing new elements or distortions. For fog, it similarly keeps only the white and red vehicles and requests realistic atmospheric haze with depth-dependent visibility reduction and soft, overcast illumination, while prohibiting clear-air or warm-light appearances. These prompts ensure consistent, content-preserving weather editing across rain, snow, and fog for all compared methods.

1.2. Loss Functions

To jointly optimize all learnable parameters of the scene representation and the dynamic nodes model, we employ a weighted combination of image-based reconstruction terms

Table 1. Prompts used for Qwen-Image and all baseline methods under each weather condition.

Weather	Prompt
Rainy	Please strictly maintain the original composition, all scene contents (including ground, buildings, vegetation, cars, background, pedestrians, etc.), their positions, and the original artistic style. Convert the scene to a rainy setting. Requirements: The image should be clear and realistic; the sky must be overcast with dark clouds; the ground should be wet with puddles and reflections; the lighting should be dim, and the overall tone should be cool to create a rainy atmosphere. Do NOT include: sunny weather, blue skies and white clouds, sunlight, dry ground, any elements not present in the original image, trees that are not in the original, distorted visuals, deformed subjects, or incorrect proportions.
Snowy	Please strictly maintain the original composition, all scene contents (including ground, buildings, vegetation, cars, background, pedestrians, etc.), their positions, and the original artistic style. Convert the scene to a snowy setting. Requirements: The image should be clear and realistic; the sky should be overcast with falling snowflakes; the ground should be naturally covered with snow; do not add any extra vegetation; the lighting should be soft, and the overall tone should be cool. If the original image contains green leaves, please turn them into snow-covered bare branches. Do NOT include: sunny weather, warm tones, sunlight, melting snow, elements unrelated to the original image, elements not present in the original image, distorted visuals, deformed subjects, or incorrect proportions.
Foggy	Please strictly maintain the original composition, all scene contents (including ground, buildings, vegetation, cars, background, pedestrians, etc.), their positions, and the original artistic style. Convert the scene to a foggy setting. Requirements: The image should be clear while maintaining realistic atmospheric fog; the sky should appear overcast; the scene should be filled with natural, soft fog that reduces visibility in the distance; lighting should be diffused and soft, with an overall cool tone. Do NOT include: sunny weather, warm tones, sunlight, dry and clear air, elements unrelated to the original image, elements not present in the original image, deformed subjects, or incorrect proportions.
Snowy & vehicle removal	Please remove all vehicles in the image except for the white and red ones, and then transform the scene into snowy weather. Requirements: The image should be clear and realistic; the sky should be overcast with falling snowflakes; the ground should be naturally covered with snow; do not add any extra vegetation; the lighting should be soft, and the overall tone should be cool. If the original image contains green leaves, please turn them into snow-covered bare branches. Do NOT include: sunny weather, warm tones, sunlight, melting snow, elements unrelated to the original image, distorted visuals, deformed subjects, or incorrect proportions.

and regularization losses,

$$\begin{aligned}\mathcal{L}_{total} =& \mathcal{L}_{rgb} + \lambda_{cc}\mathcal{L}_{cc} + \lambda_{depth}\mathcal{L}_{depth} \\ &+ \lambda_{opacity}\mathcal{L}_{opacity} + \mathcal{L}_{reg}.\end{aligned} \tag{1}$$

We set the depth weight to $\lambda_{\text{depth}} = 0.01$, the opacity weight to $\lambda_{\text{opacity}} = 0.05$, and the content consistency weight to $\lambda_{\text{cc}} = 1.0$. The losses $\mathcal{L}_{rgb}$, $\mathcal{L}_{cc}$, and $\mathcal{L}_{depth}$ have been introduced in the main text. Here, we additionally present the details of losses $\mathcal{L}_{opacity}$ and $\mathcal{L}_{reg}$.

Opacity loss. We further constrain the opacities of the Gaussians using a 2D supervision derived from the sky mask. For each view, we render an opacity map O_G from the current Gaussian scene, and use a binary sky mask M_{sky} obtained from semantic segmentation [8]. The opacity loss takes the form

$$\begin{aligned}\mathcal{L}_{\text{opacity}} =& -\sum_{u} O_G(u)\log O_G(u) \\ &-\sum_{u} M_{\text{sky}}(u)\log\big(1-O_G(u)\big).\end{aligned} \tag{2}$$

Regularization loss. The regularization loss comprises sharp shape regularization, voxel deformer regularization, temporal smoothness regularization, and scaling regularization.

1.3. Baselines

ControlNet [11] is an image editing method that introduces spatial conditional control into text-to-image diffusion models. Its core lies in achieving precise guidance of the image

editing process by injecting spatial constraint information, supporting image modification tasks under various conditions. By aligning the intermediate features of pre-trained diffusion models with spatial conditions (such as edges and depth), this method maintains the flexibility of text prompts while enhancing the structural consistency of editing results. In the weather editing task of this study, ControlNet [11] generates target weather effects based on text prompts. However, experimental results indicate that it is prone to scene content distortion (e.g., vehicle deformation, incorrect lane markings), lacks fine-grained control over weather intensity, and has a slow inference speed (only 0.033 FPS), making it difficult to meet the real-time and consistency requirements of 4D scene simulation.

TurboEdit [2] is a text-guided image editing method based on few-step diffusion models. It aims to reduce artifacts generated during diffusion model editing and improve editing efficiency and image quality by optimizing noise scheduling strategies and novel guidance techniques. By adjusting the noise distribution and guidance signals during the diffusion process, this method maintains the visual coherence of editing results while reducing inference steps, making it suitable for fast image content modification tasks. As an image-level editing comparison baseline, TurboEdit [2] can generate weather effects to a certain extent. However, limited by the nature of 2D image editing, it cannot model depth-aware atmospheric effects (e.g., depth attenuation of fog), and exhibits insufficient performance in semantic consistency and scene structure preservation. Meanwhile, its inference speed is still far below real-time requirements (0.097 FPS).

FRESCO [9] is a video editing model for zero-shot video translation. Its core innovation lies in modeling spatial-temporal correspondence to achieve cross-domain editing of video sequences without specialized training for specific tasks. By capturing spatial alignment and temporal coherence between video frames, this method completes style or scene transformation of video content under the guidance of text prompts, suitable for dynamic sequence editing tasks. In this study, FRESCO [9] serves as a video-level editing baseline to verify the weather transformation capability in dynamic scenes. However, experimental results show that it still suffers from significant scene content distortion in multi-weather editing, and has limited ability to simulate dynamic weather effects (e.g., falling rain and snow). Its inference speed (0.142 FPS) is difficult to support the real-time simulation needs.

Qwen-Image [7] is a powerful text-guided image editing foundation model with high-quality image generation and editing capabilities. It can accurately respond to semantic requirements in text prompts and generate realistic and content-consistent editing results. Trained on large-scale data, this model achieves a good balance between image content preservation and target effect generation, supporting flexible editing in various scenarios. However, as a pure 2D image editing model, Qwen-Image [7] lacks modeling of temporal coherence between frames. When used alone, it is prone to temporal flickering and geometric inconsistency issues, and cannot support object-level editing and dynamic weather simulation of 4D scenes.

ClimateNeRF [5] is a 3D-level weather editing method that integrates physical simulation with Neural Radiance Fields (NeRF) to enable the editing of various climate effects in 3D scenes. By leveraging the inherent 3D geometric modeling capability of NeRF, this method achieves more realistic environmental rendering compared to 2D image editing approaches. However, a key limitation is that it is confined to static reconstruction and simulation of static weather phenomena, it lacks the ability to model dynamic vehicles and cannot simulate dynamic weather effects, such as falling rain or snow, which are critical for 4D urban scene simulation. Additionally, it supports only a limited range of weather editing operations and fails to realize flexible text-guided weather control. Furthermore, ClimateNeRF [5] has a rendering speed of 0.032 FPS, which is insufficient to meet the demands of real-time simulation.

1.4. Evaluation Details

CLIP-Score (CLIP-S [4]). CLIP-S measures the visual similarity between the original image I and the edited image $\hat{I}$ using the CLIP image encoder. Let $f_{\text{CLIP}}(\cdot)$ denote the CLIP model, then the metric is computed as:

$$\text{CLIP-S} = \frac{\langle f_{\text{CLIP}}(I),\, f_{\text{CLIP}}(\hat{I})\rangle}{\|f_{\text{CLIP}}(I)\|_2\, \|f_{\text{CLIP}}(\hat{I})\|_2}. \tag{3}$$

CLIP Directional Similarity (CLIP-DS [3]). CLIP-DS evaluates whether the "editing direction" in CLIP space—produced by the edited image relative to the original image—aligns with the target editing direction defined by the text prompt. Given the original image I, edited image $\hat{I}$, and target text instruction T, the metric is:

$$\text{CLIP-DS} = \frac{\left\langle f_{\text{CLIP}}(\hat{I}) - f_{\text{CLIP}}(I),\; f_{\text{CLIP}}(T)\right\rangle}{\|f_{\text{CLIP}}(\hat{I}) - f_{\text{CLIP}}(I)\|_2\, \|f_{\text{CLIP}}(T)\|_2}. \tag{4}$$

Semantic Consistency Score (Sem-CS). Sem-CS measures the semantic consistency between edited and original images. We apply a `ConvNeXt-XL-384×384` [6] model pretrained on ADE20K [12] to perform panoptic segmentation on the original image I and the edited image $\hat{I}$. Let IoU_c denote the IoU of category c, aggregated over all ADE20K classes $\mathcal{C}$. Sem-CS is defined as the frequency-weighted IoU (fwIoU):

$$\text{Sem-CS} = \frac{\sum_{c\in\mathcal{C}} n_c\, \text{IoU}_c}{\sum_{c\in\mathcal{C}} n_c}, \tag{5}$$

Figure 1. **Qualitative comparison of rainy weather on the Waymo Open Dataset.** Our model excels at capturing complex lighting interactions and environmental changes induced by rain, providing a higher level of physical realism.

Figure 2. **Qualitative comparison of snowy weather on the Waymo Open Dataset.** As shown, WeatherCity generates highly realistic snow accumulation on vehicles and roads while avoiding the texture distortions common in other approaches.

where n_c is the number of pixels belonging to class c in the ground-truth segmentation of the original image.

We note that fog synthesis substantially reduces scene visibility, which consequently invalidates metrics designed for content preservation (e.g., CLIP-S, Sem-CS). Thus, for foggy weather, our evaluation is solely based on the CLIP-DS metric.

2. Additional Results and Analysis

2.1. Detailed Quantitative and Qualitative Results

Table 2 and Table 3 present the complete quantitative comparison results for the Waymo and nuScenes datasets, respectively. Our method significantly outperforms all baseline approaches (ControlNet, FRESCO, and TurboEdit) across all metrics. Specifically, higher CLIP-S indicates better content preservation w.r.t. the original scene. Furthermore, the improvements in Sem-CS quantify our method's ability to preserve the original scene content—such as road layout and vehicle geometry—during the weather transformation process, confirming that WeatherCity minimizes the content distortion often observed in image-level editing frameworks.

Rainy: As illustrated in Fig. 1 and Fig. 4, our method successfully renders high-frequency details such as falling raindrops and specular reflections on wet road surfaces. Unlike baselines such as ControlNet and FRESCO, which tend to apply a global style transfer that often blurs the boundary

Figure 3. **Qualitative comparison of foggy weather on the Waymo Open Dataset.** WeatherCity demonstrates superior depth-consistent haze rendering, effectively preserving the semantic layout of the original scene compared to baseline methods.

Figure 4. **Qualitative comparison of rainy weather on the nuScenes Dataset.** Our method accurately simulates realistic wet surface reflections and rain streaks, significantly outperforming ControlNet and FRESCO in terms of visual fidelity.

between the road and the environment, our method leverages 3D scene representations to ensure that reflections are geometrically consistent with the camera view.

Snowy: In snowy weather generation, as shown in Fig. 2 and Fig. 5, our method achieves realistic snow accumulation on distinct surfaces, such as vehicle roofs and vegetation, without altering the underlying object semantics. The visual evidence shows that competitive methods (e.g., ControlNet) frequently hallucinate structures or warp the shape of vehicles when attempting to add snow textures. WeatherCity effectively avoids these artifacts, preserving the clear contours of dynamic objects and lane markings.

Foggy: The foggy scenarios highlight the advantage of our depth-aware approach. As seen in Fig. 3 and Fig. 6, WeatherCity simulates physically plausible depth attenuation, where visibility decreases naturally with distance. In contrast, baselines like TurboEdit and FRESCO often apply a uniform haze layer or introduce artifacts that obscure nearby objects, failing to respect the scene's depth map.

Overall, while baseline methods can produce general weather-like appearances, they suffer from severe content distortion, manifesting as warped vehicles and erroneous scene structures. WeatherCity overcomes these limitations, offering a robust solution for high-fidelity, geometry-preserving weather simulation.

2.2. Temporal Consistency Comparison

Temporal consistency is crucial for 4D urban scene simulation, requiring coherent motion of dynamic objects and continuous evolution of weather effects across frames. Qualitative comparisons on Waymo/nuScenes dynamic sequences (Fig. 7, Fig. 8, Fig. 9) reveal that baseline methods generally suffer from inter-frame inconsistency: scene geometry undergoes deformation and dynamic vehicles exhibit shape changes between consecutive frames, while weather effects display random fluctuations with noticeable inter-frame flickering. In contrast, our approach achieves temporally consistent weather editing throughout the temporal sequence.

2.3. 3D Baseline Comparison

We compare WeatherCity with ClimateNeRF [5], a NeRF-based 3D weather editing method, with quantitative and qualitative results presented in Tab. 4 and Fig. 10. ClimateNeRF exhibits significant limitations due to its static 3D representation: it cannot effectively model dynamic objects (resulting in motion-blurred vehicles) nor simulate evolving weather effects. In contrast, WeatherCity, leveraging dynamic Gaussian modeling, outperforms ClimateNeRF across all metrics while delivering more realistic editing effects. Our approach additionally supports dynamic

Figure 5. **Qualitative comparison of snowy weather on the nuScenes Dataset.** Our approach produces the most visually plausible winter scenes with natural white-out effects, surpassing the consistency and quality of competing editing frameworks.

Figure 6. **Qualitative comparison of foggy weather on the nuScenes Dataset.** WeatherCity achieves a natural atmospheric haze that smoothly obscures distant objects, exhibiting fewer artifacts than TurboEdit or other baselines.

weather particles (such as falling snowflakes) and achieves significantly higher rendering efficiency than ClimateNeRF, conclusively validating its superiority.

References

[1] 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. In *The Thirteenth International Conference on Learning Representations*. 1

[2] Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–12, 2024. 3, 8

[3] 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. 3

[4] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021. 3

[5] 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*, pages 3227–3238, 2023. 3, 5, 8

[6] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 3

[7] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 3, 8

[8] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 2

[9] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Fresco: Spatial-temporal correspondence for zero-shot video translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8703–8712, 2024. 3, 8

[10] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 1053–1061, 2024. 1

[11] 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*, pages 3836–3847, 2023. 2, 3, 8

[12] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through

Figure 7. Qualitative comparison of rainy weather on temporal consistency. The baselines suffer from scene deformation and erratic fluctuations in weather effects, exhibiting noticeable inter-frame flickering. In contrast, our approach enables temporally consistent weather editing.

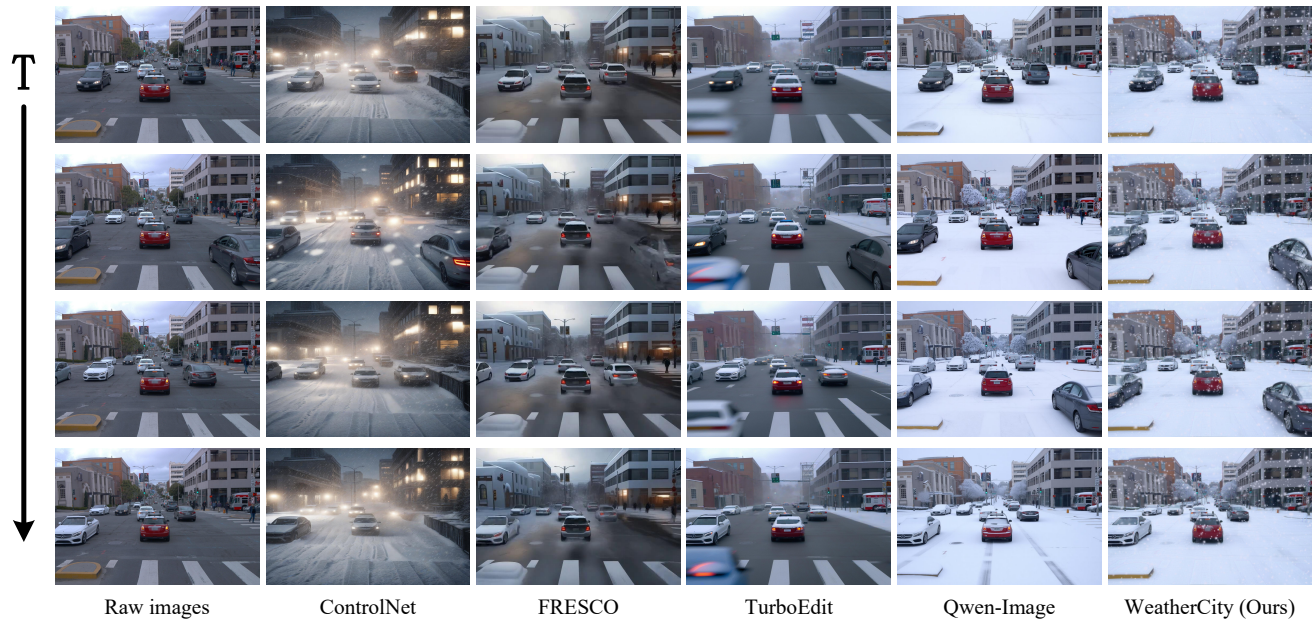

Figure 8. Qualitative comparison of snowy weather on temporal consistency. The baselines suffer from scene deformation and erratic fluctuations in weather effects, exhibiting noticeable inter-frame flickering. In contrast, our approach enables temporally consistent weather editing.

ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

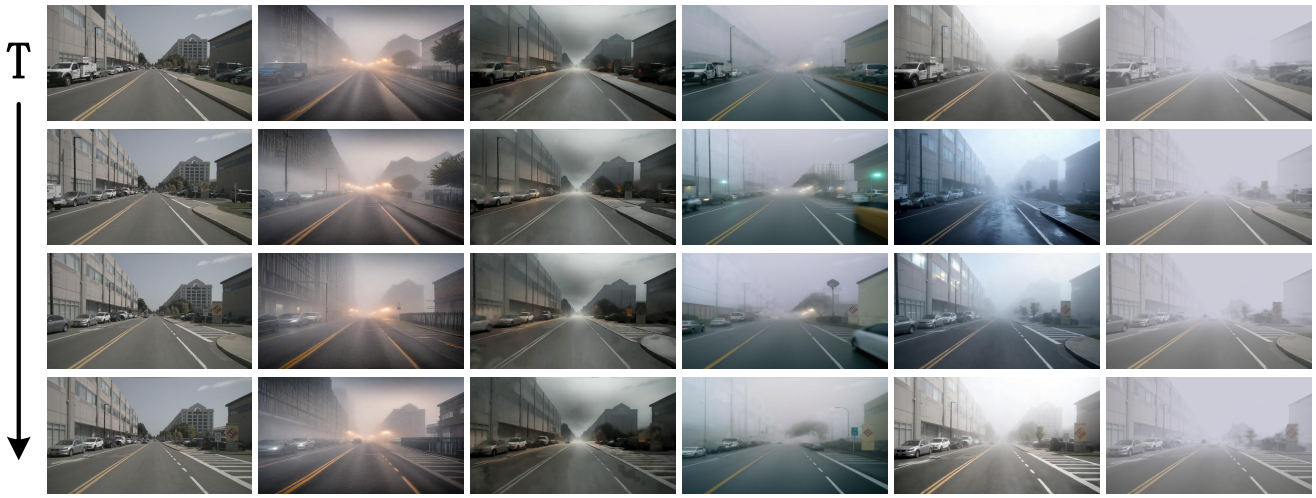

Figure 9. Qualitative comparison of foggy weather on temporal consistency. The baselines suffer from scene deformation and erratic fluctuations in weather effects, exhibiting noticeable inter-frame flickering. In contrast, our approach enables temporally consistent weather editing.

Table 2. Comparison on Waymo Open Dataset. ↑ means higher is better.

Method	**Rainy**			**Snowy**			**Foggy**
	CLIP-S↑	CLIP-DS↑	Sem-CS↑	CLIP-S↑	CLIP-DS↑	Sem-CS↑	CLIP-DS↑
ControlNet [11]	0.654	0.228	0.713	0.615	0.261	0.677	0.225
TurboEdit [2]	0.843	0.233	0.825	0.816	0.228	0.787	0.221
FRESCO [9]	0.721	0.209	0.852	0.719	0.253	0.797	0.177
Qwen-Image [7]	0.813	0.248	0.845	0.757	0.310	0.840	0.251
WeatherCity (Ours)	**0.898**	**0.300**	**0.931**	**0.847**	**0.330**	**0.899**	**0.278**

Table 3. Comparison on nuScenes Dataset. ↑ means higher is better.

Method	**Rainy**			**Snowy**			**Foggy**
	CLIP-S↑	CLIP-DS↑	Sem-CS↑	CLIP-S↑	CLIP-DS↑	Sem-CS↑	CLIP-DS↑
ControlNet [11]	0.703	0.250	0.810	0.609	0.234	0.812	0.201
TurboEdit [2]	0.806	0.225	0.848	0.758	0.261	0.811	0.266
FRESCO [9]	0.726	0.220	0.863	0.694	0.239	0.847	0.213
Qwen-Image [7]	0.823	0.256	0.891	0.785	0.302	0.913	0.264
WeatherCity (Ours)	**0.880**	**0.272**	**0.977**	**0.860**	**0.333**	**0.960**	**0.301**

Table 4. Comparison on Waymo Open Dataset scene 788. ↑ means higher is better.

Method	**Snowy**			**Foggy**	**FPS↑**
	CLIP-S↑	CLIP-DS↑	Sem-CS↑	CLIP-DS↑	
ClimateNeRF [5]	0.807	0.294	0.905	0.269	0.032
WeatherCity (Ours)	**0.847**	**0.341**	**0.941**	**0.280**	**25.67**

Figure 10. Qualitative comparison with ClimateNeRF on Waymo Open Dataset. ClimateNeRF is limited to static modeling and static weather effect editing, while WeatherCity produces more photorealistic results and additionally supports dynamic weather particles, such as falling snowflakes.