

AutoScape: Geometry-Consistent Long-Horizon Scene Generation

Supplementary Material

In the supplementary material, we provide additional content that could not be included in the main paper due to page and format constraints. The supplementary material is organized as follows:

- In § A describes the accompanying video material.
- In § B presents the remaining implementation details.
- In § C presents the architectural and training details of the RGB-D VAE of AutoScape.
- In § D provides additional experimental results.

A. Video Material

In the supplementary material, we include two video files to enhance the understanding of our work. The first, *comparison.mp4*, showcases a comparative analysis of the generated video quality against that produced by other methods. The second, *interpolation_video.mp4*, illustrates the interpolation process in detail, including the rendered interpolation points and their corresponding interpolation generations.

B. Remaining Implementation Details

This section presents the remaining implementation details that are not covered in the main paper due to space limitations. The proposed method is implemented with PyTorch [46] and the Diffuser library.

Optimization Settings. For both RGB-D pretraining and rendering-conditioned training, we utilize the AdamW optimizer to facilitate optimization. The learning rate (lr) and weight decay (wd) are set to 1×10^{-4} and 1×10^{-2} , respectively, with a learning rate warmup applied over the first 3000 iterations. Gradient clipping with a maximum norm of 1 is implemented to enhance training stability. Additionally, both training and inference are conducted using bfloat16 (brain floating-point 16-bit) precision to ensure computational efficiency and optimization effectiveness.

HD Map and Bbox Condition. To enable more flexible controllability, we augment our RGB-D diffusion model with a ControlNet [86] branch to encode HD maps and object bounding boxes. Figure 9 provides a visualization of these conditioning inputs. Specifically, for the map condition, we extract the layers (*i.e.*, lane boundary, lane divider, and pedestrian crossings) from the vector HD maps [8, 32, 84] and then project them onto the image plane. To specify the location and orientation of objects precisely, we utilize two types of box control images: semantic box control and orientation box control. Both box controls are derived by projecting 3D bounding boxes onto the image plane with the camera parameters. For the semantic box control, different colors are used to distinguish vehicles, pedestrians, roadblocks, etc. For the

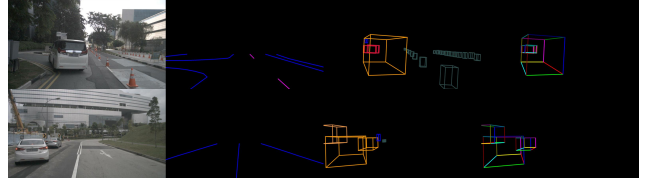


Figure 9. The control signals with the corresponding images. From left to right are ground-truth RGB images, projected maps, semantic box control, and orientation box control.

orientation box control, the orientation of each vehicle is indicated by assigning unique colors to each edge of the box. Figure 8 in the main paper demonstrates the controlled generation through these protocols. Note that our conditioning strategies for HD Maps and objects are different from those in MagicDrive [16] or DriverDreamer [68].

Training with ControlNet. ControlNet is only incorporated during the rendering-conditioned training stage, as the HD maps and object boxes conditions are not available for the RGB-D pre-training stage, where we use datasets beyond driving. The ControlNet is initialized using the U-Net model from the RGB-D pretraining stage, following those outlined in the original ControlNet [86]. During the rendering-conditioned training stage, we fine-tune both ControlNet and U-Net to facilitate convergence.

Inference Settings. For diffusion model inference, we utilize DPM-Solver [40] with 50 steps. Additionally, classifier-free guidance [21] is employed to enhance the quality of conditioned generation, using a guidance strength of 7.5 in accordance with the default settings of the diffuser library.

C. Details of the RGB-D VAE

Similar to LDM3D [61], we modify the VAE to support depth encoding and decoding to accommodate depth generation, while preserving the latent code shape. Specifically, we first normalize the depth to 0-1, with a maximum depth of 300 meters, to align with the scale of the RGB channels. Then, the normalized depth (1 channel) is concatenated with RGB (3 channels) to create a 4-channel RGB-D input for the VAE. Architecturally, we extend the first and last convolutions in both the encoder and decoder to accommodate this 4-channel input and output, ensuring compatibility with RGB-D data. As the default 8-bit choice for RGB channel leads to significant precision loss for depth channel [61], we employ 16-bit precision for RGB-D inputs and outputs to retain depth details accurately. Since the latent feature shape remains unchanged, we apply the existing U-Net architecture directly for latent diffusion.

The RGB-D VAE is initialized with the pretrained RGB VAE from Stable Diffusion models [51]. The added parameters are initialized to zero to preserve pretrained knowledge. The optimization target is defined as

$$\begin{aligned}\mathcal{L}_{\text{VAE}} = & \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log p_{\theta}(\mathbf{x}_{\text{rgb}} | \mathbf{z})] \\ & + \lambda_{\text{depth}} \cdot \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log p_{\theta}(\mathbf{x}_{\text{depth}} | \mathbf{z})] \quad (8) \\ & + D_{\text{KL}}(q_{\phi}(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z}))\end{aligned}$$

where \mathbf{x}_{rgb} represents the RGB image data. $\mathbf{x}_{\text{depth}}$ represents the depth map data. \mathbf{x} is the combination of \mathbf{x}_{rgb} and $\mathbf{x}_{\text{depth}}$. $q_{\phi}(\mathbf{z} | \mathbf{x}_{\text{rgb}}, \mathbf{x}_{\text{depth}})$ is the encoder network with parameters ϕ , encoding both RGB and depth inputs. $p_{\theta}(\mathbf{x}_{\text{rgb}} | \mathbf{z})$ and $p_{\theta}(\mathbf{x}_{\text{depth}} | \mathbf{z})$ are the decoder networks reconstructing RGB images and depth maps from the latent variable \mathbf{z} . D_{KL} is the Kullback-Leibler divergence between the approximate posterior $q_{\phi}(\mathbf{z} | \mathbf{x})$ and the prior $p(\mathbf{z})$.

The first and second term, $\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log p_{\theta}(\mathbf{x}_{\text{rgb}} | \mathbf{z})]$ and $\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log p_{\theta}(\mathbf{x}_{\text{depth}} | \mathbf{z})]$, minimize the reconstruction errors for the RGB images and depth maps, respectively. The third term $D_{\text{KL}}(q_{\phi}(\mathbf{z} | \mathbf{x}) \| p(\mathbf{z}))$, regularizes the latent space by enforcing alignment with a predefined prior distribution, thereby promoting smoothness and continuity in the latent space \mathbf{z} .

Given that depth maps tend to contain less high-frequency information than RGB images due to the inherently smooth nature of geometric data, the reconstruction loss for depth is generally smaller than for RGB. To address this imbalance, we introduce a weighting factor, λ_{depth} , to amplify the depth reconstruction loss. In practice, we set $\lambda_{\text{depth}} = 10$.

To train the RGB-D diffusion model, we implement a two-stage training strategy, as outlined in § 5.1.

D. Additional Experimental Results

More baseline results. To further evaluate the quality of keyframes generated by the proposed AutoScape in comparison to the baselines, we apply ViewCrafter to interpolate the keyframes produced by WonderJourney[†]. This results in FID and FVD scores of 59.1 and 858.9, respectively, which are significantly higher than those achieved by AutoScape (35.1 and 359.0). These findings highlight the superior visual quality of the keyframes generated by our method.

Compare with single-stage video diffusion models. To further assess the performance of our proposed two-stage method against the state-of-the-art one-stage approach, we fine-tune COSMOS-Transfer [1] with the HD map from nuScenes and perform autoregressive generation to produce long videos. COSMOS-Transfer achieved an FID of 44.2 and an FVD of 436.1, whereas our method attained 35.1 and 359.0, respectively, demonstrating its clear superiority.