# *MagicDrive-V2*: High-Resolution Long Video Generation for Autonomous Driving with Adaptive Control

## Supplementary Material (Appendix)

Please find the videos on our project website: https://flymin.github.io/magicdrive-v2/

**Note**: Our model is capable of generating videos at a resolution of 848×1600 for 241 frames, which is the highest resolution and frame count in the nuScenes dataset. However, the inference cost is currently substantial. Therefore, the primary numerical results in our paper do not utilize this maximum setting. We have included some generated results in Appendix L and on our project website for reference. Future work may focus on further reducing the inference cost.

## A. Sequence Parallel Training

Inspired by Zheng et al. [47], we employ sequence parallelism to train DiT models with large sequence lengths. As illustrated in Figure I, we partition each input across the spatial dimension onto different GPUs. Most operations can be executed within a single GPU; however, the attention blocks necessitate communication. On the right side of Figure I, we demonstrate the communication process, where the full sequence is gathered, but the attention heads are distributed across different GPUs. This approach allows for peer-to-peer communication between GPUs while maintaining a roughly equal load.

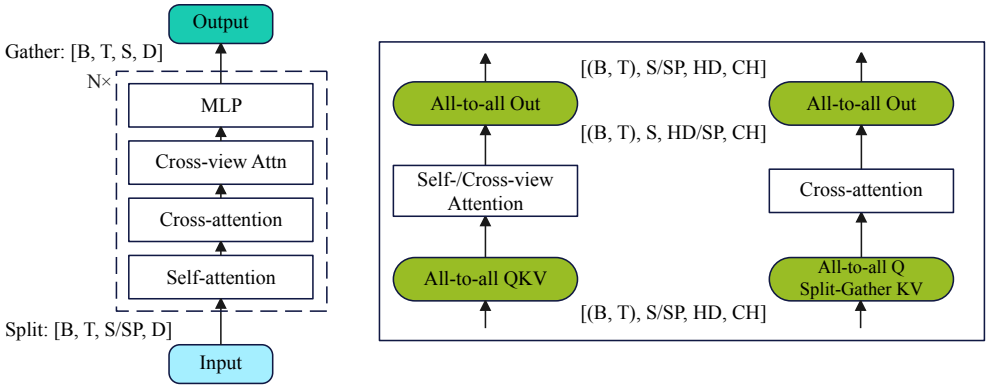



Figure I. **Diagram for Sequence Parallel**. *Left*: We split the spatial dimension before the first block and gather them after the last block. *Right*: For each attention module, we use all-to-all communication, changing the splitting dimension to attention heads. B: batch; T: temporal dimension; S: spatial dimension; D: latent dimension; HD: number of heads; CH: per-head dimension; SP: sequence parallel size.

Additionally, for VAE encoding and decoding, we partition based on batch size and the number of camera views, leveraging multiple GPUs to accelerate processing.

## B. More Details for Mixed Resolution and Frames Training

*MagicDrive-V2* is trained through a progressive training approach with variable length and resolution data configurations (see Section 4.5). Consequently, our method of data mixing corresponds to the three training stages, as detailed in Table I.

Inspired by [47], to maximize the utilization of GPU resources, we employed a bucket-like approach to adjust the data composition. Specifically, each GPU process (or sequence parallel communication group) loads only one type of data to ensure alignment of the batch dimension. Using the training time of the video format with the longest iteration time at batch size = 1 as a benchmark, we adjusted the batch sizes of other data formats so that each type runs at approximately the same speed. Notably, during stage 3 training, due to the limited number of full video clips, we repeat this type of data within an epoch. This ensures that different types of data have a similar magnitude of batch numbers within an epoch.

## C. Efficiency of Progressive Bootstrap Training

The three-stage progressive training approach markedly improves model training efficiency relative to direct Stage 3 training. Table II indicates that, over 4 days, for example, Stage 1 executes approximately 60 times more iterations than Stage 3, and

| Stage | Resolution | Frame(s) | Sequence Parallel | Training Step |
|---|---|---|---|---|
| Stage 1 | 224×400 | Img | - | 80000 |
| Stage 2 | 224×400<br>424×800 | Img, 9, 17, 33, 65<br>Img, 9, 17, 33 | - | 40000 |
| Stage 3 | 224×400<br>424×800<br>848×1600 | Img, 17, full<br>Img, 17, 33, 65, 129<br>Img, 9, 17, 33 | 4 | 30000 |

Table I. **Configuration for Variable Length and Resolution Training**. The mixing configuration aligns with our progressive bootstrap training with 3 stages, from low-resolution images to high-resolution long videos.

| Stages | Seconds/Iter. | Iter. for 4 days |
|---|---|---|
| stage 1 | 4.32 | 80k |
| stage 2 | 39.84 | 8.7k |
| stage 3 | *264.96 | 1.3k |

Table II. **Speed for Each Training Stage of *MagicDrive-V2***, measured on NVIDA A800 GPUs. Over a 4-day period (for example), Stage 1 training yields nearly 60 times more iterations than Stage 3, and Stage 2 offers about 7 times more. *This value is calculated by multiplication with sequence parallel (SP) size (in practice, we use SP size of 4 for the stage 3, with 66.24s/it).

Stage 2 achieves about 7 times more iterations. The progressive training is vital for controlled generative models, which require extensive iterations for effective convergence, as discussed in Section 4.5. The progressive strategy enables the rapid acquisition of high-quality video generation capabilities, utilizing faster iterations in the early stages to enhance convergence and expedite learning.

## D. Video Generation Speed

Table III shows the breakdown of computation and inference costs, together with comparisons with others. By adopting the sequence parallel, our inference speed is on par with the performance of NVIDIA's Cosmos-transfer1 [30]. We open-source our implementation and welcome future efforts on optimization.

| Method | resolution | # views | # frames | Diff. Steps (sec/it) | Latent Dec. (sec) | Total (min) | Device |
|---|---|---|---|---|---|---|---|
| *MagicDrive-V2* | 848x1600 | 3 | 193 | 18.03 | 82.83<br>248.24 (1 GPU) | 11.68 | H20 |
| | 848×1600 | 6 | 241 | 53.74 | 103.36 | 28.92 | |
| | 848×1600 | 6 | 121 | 28.18 | 51.94 | 8.27 | |
| Cosmos-transfer1 [30] | 704×1280 | †6 | 121 | 20.88 | 54 | 19.92 | A100-SXM4 |
| | 704×1280 | 1 | 121 | 3.48 | 9 | 3.32 | |

Table III. **Inference speed of *MagicDrive-V2***. Unless otherwise specified, we use 8 GPUs for testing. As compared with Cosmos-transfer1, our implementation of *MagicDrive-V2* offers reasonable inference speeds. We also implemented a parallel decoding strategy for latent decoding, offering 3× speedup over 1 GPU. Total time contains overhead from CPU offloading. †There is no such a model; this row is estimated by 6× the single-view time.

## E. More Experimental Details

The nuScenes [4] dataset includes 12Hz unannotated data and 2Hz annotated data. According to our experiments, high-frame-rate videos are more beneficial for generative model learning. Therefore, we follow [12] and interpolated the 2Hz annotations to 12Hz annotations with ASAP [37]. Although the interpolation results are not entirely accurate, they do not affect the training for video generation. The Waymo Open Dataset [34] includes 10Hz annotated data. We follow the official

splitting, which has 798 clips for training and 202 clips for validation. The original dataset contains 5 views. However, the dimensions of the left and right side views are smaller than those of the three front views, and their field of view is limited. Therefore, we only retain three front views for training and validation.

**Semantic Classes for Generation**. We follow [12] in data setup on nuScenes. Specifically, for objects, ten categories include car, bus, truck, trailer, motorcycle, bicycle, construction vehicle, pedestrian, barrier, and traffic cone. For the road map, eight categories include drivable area, pedestrian crossing, walkway, stop line, car parking area, road divider, lane divider, and roadblock. For Waymo, the semantics for objects include pedestrian, car, and cyclist, while the semantics for road maps include drivable area, crosswalk, road line (yellow), and road line (white).

## F. More Training Details

**Optimization**. We train our diffusion models using Adam optimizer and a constant learning rate at $8e^{-5}$ with a 3000-step linear warm-up in the last two stages. We primarily use 32 NVIDA A800 GPUs (80G) for training. Our model can also be trained with Ascend 910B (64G). The batch size for each stage is set according to the iteration speed, following the bucket strategy as [47]. For example, in stage 2, we set the batch size for 33×424×800 to 1, which takes about 30s/it. Then we set the batch size to other video types to achieve about 30s/it. This strategy can ensure the load balance between different GPU processes.

**Inference**. By default, images/videos are sampled using Rectified Flow [10] with 30 steps and the classifier-free-guidance (CFG) scale at 2.0. To support CFG, we randomly drop different conditions at a rate of 15%, including embeddings for text, camera, ego trajectory, and boxes. We follow Gao et al. [12] to use $\{\mathbf{0}\}$ as the *null* condition for maps in CFG inference. When inferring high-resolution long videos, we also use sequence parallel (Appendix A) to fit in the limited memory of a single GPU.

## G. Human Evaluation for Multi-frame & Multi-view Consistency

Evaluating multi-frame and multi-view consistency in video generation has long been a challenging issue [1], as the academic community lacks a unified standard to accurately assess such consistency. To address this, we employed human evaluation to measure these two aspects of consistency. Specifically, we invited participants with diverse backgrounds to compare videos generated by *MagicDrive-V2* and Gao et al. [12] under identical conditions. Participants were asked to select the video with better consistency, and the winning probability of each model was statistically analyzed. As illustrated in Figure II, *MagicDrive-V2* demonstrated significantly superior consistency, indicating a substantial improvement over Gao et al. [12].

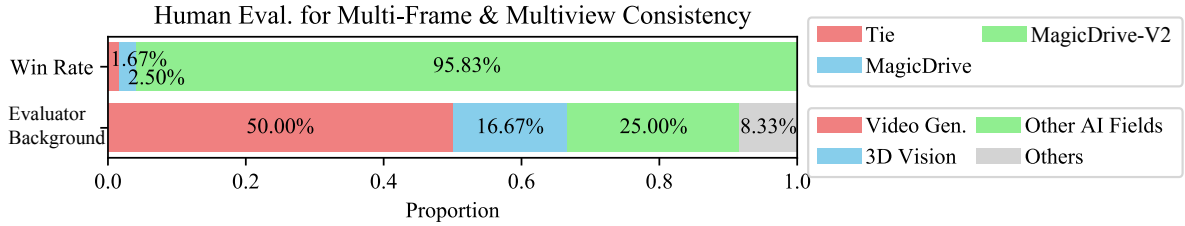



Figure II. Experts evaluated content consistency against Gao et al. [12], showing *MagicDrive-V2* generates videos with superior consistency.

Besides, Figure III provides an ablation comparison between with and without the MVDiT block, showing that it is crucial to include such a block for multi-view consistency.

## H. Human Evaluation on Text Control

We validate text control by generating videos under six weather conditions and asking humans to judge alignment with text prompts. The confusion matrix in Figure IV shows high recognition accuracy ($> 70\%$), confirming effective text control.

## I. More Comparison among VAEs

To quantitatively compare the performance of the VAE, we randomly selected two 6-view videos from the nuScenes dataset and used the PSNR metric to evaluate the VAE's reconstruction ability. Table IV presents the results, averaged across the six


w/o Multi-view Block

w/ Multi-view Block (ours)

Figure III. **Ablation on the MVDiT block**. MVDiT block is the key to enabling multi-view consistency in driving video generation.

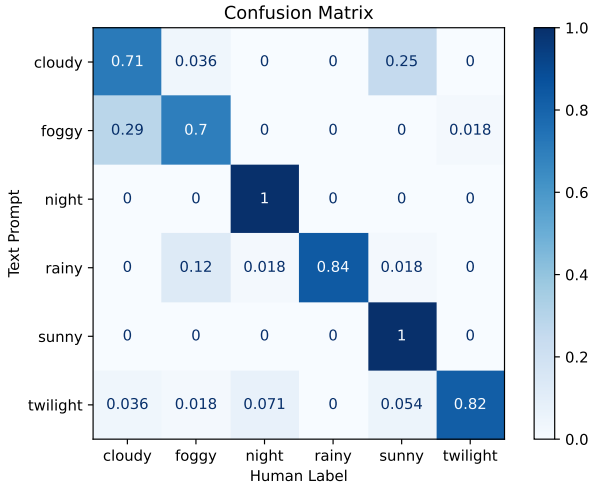



Figure IV. **Confusion matrix to show the text control ability in *MagicDrive-V2***. *MagicDrive-V2* supports the control of 6+ weather conditions.

| Resolution | Model | Image | 17 fr. | 33/34 fr. |
|---|---|---|---|---|
| 224×400 | CogVAE | **34.4261** | **31.0900** | **30.5986** |
| | Open-Sora | 30.4127 | 27.9238 | 27.5245 |
| | SD VAE | 27.7131 | 27.7593 | 27.9404 |
| 424×800 | CogVAE | **38.4786** | **33.5852** | **32.9202** |
| | Open-Sora | 33.6114 | 30.2779 | 29.8426 |
| | SD VAE | 30.9704 | 31.0789 | 31.3408 |
| 848×1600 | CogVAE | **41.5023** | **36.0011** | **35.1049** |
| | Open-Sora | 37.0590 | 33.2856 | 32.8690 |
| | SD VAE | 37.0504 | 33.2846 | 32.8680 |

Table IV. **VAE Comparison for Street Views**. CogVAE [43] and Open-Sora [47] (1.2) are 3D VAEs; SD VAE [31] is 2D VAE, which is also widely adopted by previous street view generation (*e.g.*, [12]). Results are PSNRs calculated through videos from the nuScenes validation set. *MagicDrive-V2* adopts CogVAE.

views. From these results, we observe that CogVAE [43] demonstrates the best reconstruction ability, even surpassing the 2D VAE [31]. Comparing the results from different settings, we find that the current 3D VAEs exhibit good generalization ability for long videos, primarily due to the window-based downsampling techniques [43, 47]. Additionally, we observe that high-resolution content retains a relatively high PSNR after VAE reconstruction, indicating that the current VAEs are more favorable for high-resolution data. This observation also supports our motivation for high-resolution generation.

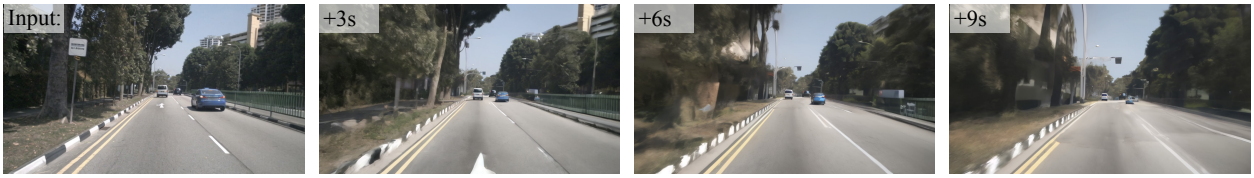
## J. Reason for Using CogVAE without the Pre-trained Diffusion Model

Informed by the work of MagicDrive [12] and others [13], fine-tuning from a well-performing diffusion model can effectively accelerate model convergence. Consequently, in the initial implementation of the DiT architecture, we experimented with Open-Sora 1.2 [47]'s VAE and diffusion models. However, the results were suboptimal, with image generation and video controllability falling short of MagicDrive's performance. We attribute this primarily to the limited generalization capability of text-to-video diffusion and, more critically, to the inadequate reconstruction ability of the VAE.

We conducted a comparative analysis of VAEs, as detailed in Section 5.3 and Appendix I, and found CogVAE [43] to perform well. Given that the VAE determines the upper limit of generation quality, we opted to use CogVAE for video encoding. Notably, CogVideoX [43] employs a novel DiT structure, where each layer's latent space integrates both video and text condition information. This approach may complicate the design of geometry-related conditions. Furthermore, CogVideoX was not trained in a driving scenario. To eliminate these potential confounding factors, we decided to train the diffusion model from scratch using CogVAE. This strategy allows us to move beyond the constraints of pre-trained models, enabling more flexible modifications to the model architecture to achieve multi-view consistency and spatiotemporal encoding of geometry conditions.

Our experience directly demonstrates that high-resolution, long street-view video generation does not necessarily require pre-trained image-text or video-text models. Even so, this is beyond the primary focus of our paper, and we leave related questions for future work.

## K. Single Inference v.s. Rollout Inference




(a) **Generation from Vista**. It takes the first frame as input and generates the following (only supporting the front view).


(b) **Generation from *MagicDrive-V2***. We take conditions as inputs and generate the full video (only show the first 9s for comparison).

Figure V. **Comparison between Rollout for Long Videos (Vista [13]) and Single Inference (our *MagicDrive-V2*)**. Although rollout can handle long videos, the quality is significantly degraded. In contrast, our extrapolation maintains high quality in long video generation.

To achieve long video generation, previous work typically employs a method of future frame prediction combined with rollout. This involves, after the $n$-th inference, taking the last $l$ frames from this inference as the first $l$ frames for the $n+1$-th inference, thus enabling long video generation. However, since the model does not directly capture long-term dependencies and accumulates errors with each inference, such rollouts often fail to support sufficiently long videos. Among rollout methods, Vista [13] currently achieves relatively good results. We compared a 9-second video generated by performing 4 rollouts with Vista (the paper claims it can support 6 rollouts) to a 9-second segment produced by our method, *MagicDrive-V2*. It is evident that our method maintains consistent video quality over long sequences, whereas Vista's results show a

noticeable decline. Therefore, we believe that the hybrid training and length extrapolation approach adopted by *MagicDrive-V2* can achieve higher quality in long video generation.

## L. More Visualization

As said in the "Note", *MagicDrive-V2* is capable of generating $6 \times 848 \times 1600 \times 241$ videos (20s at 12 fps). We include more generated samples in Figure VI-VII. Please see the videos on our project page.


Figure VI. We show some frames from the generated $6 \times 848 \times 1600 \times 241$ videos with the same scene configuration (*i.e.*, boxes, maps, cameras, and ego trajectory) but under different weather conditions. Conditions are from the nuScenes validation set.

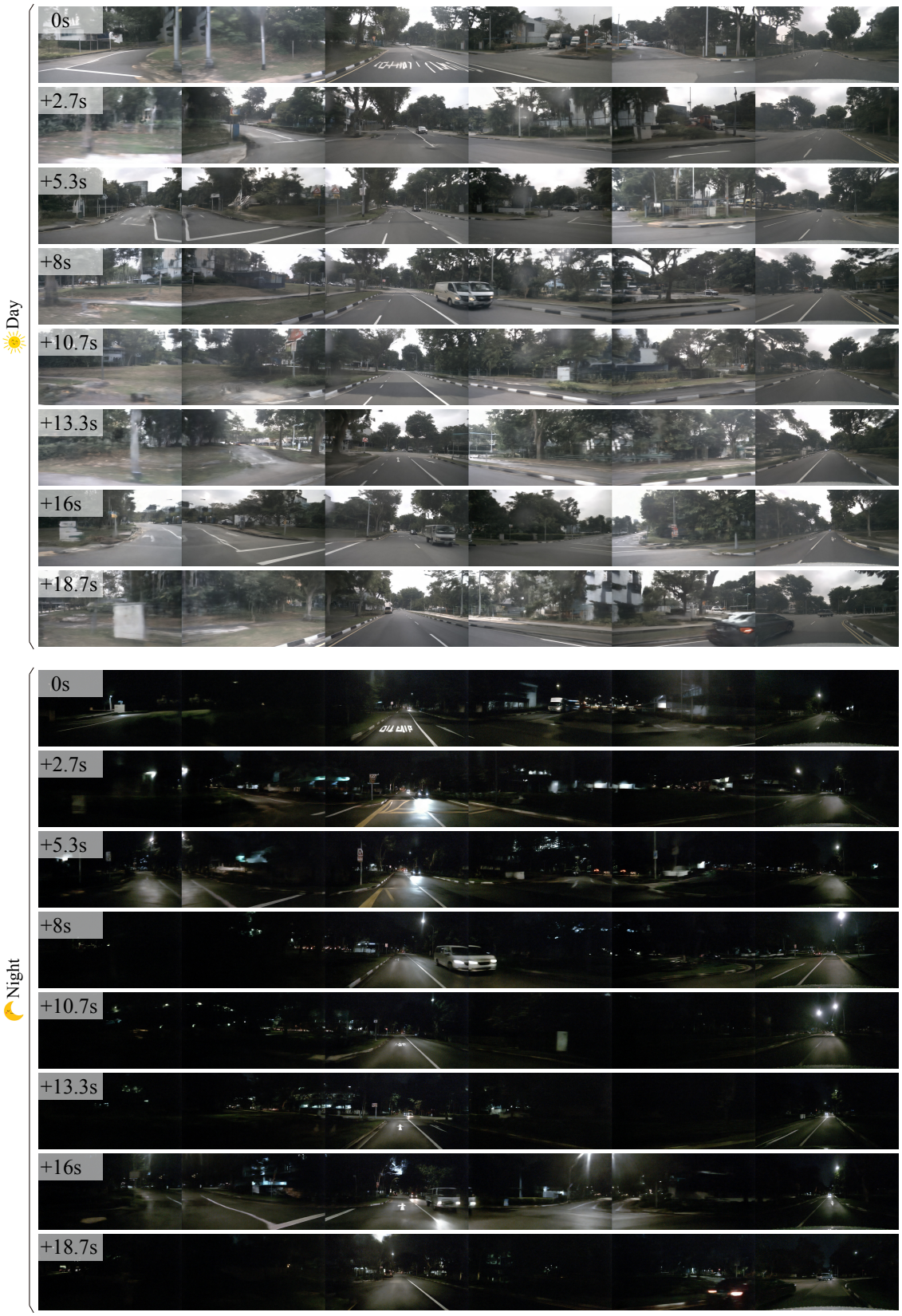



Figure VII. We show some frames from the generated $6 \times 848 \times 1600 \times 241$ videos with the same scene configuration (*i.e.*, boxes, maps, cameras, and ego trajectory) but under different time-of-day conditions. Conditions are from the nuScenes validation set.