Cosmos-Drive-Dreams: Scalable Synthetic Driving Data Generation with World Foundation Models

Xuanchi Ren*, Yifan Lu*, Tianshi Cao*, Ruiyuan Gao*, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixe, Mike Chen, Sanja Fidler, Huan Ling [1]
NVIDIA

Abstract

Collecting and annotating real-world data for safety-critical physical AI systems, such as Autonomous Vehicle (AV), is time-consuming and costly. It is especially challenging to capture rare edge cases, which play a critical role in training and testing of an AV system. To address this challenge, we introduce the *Cosmos-Drive-Dreams* - a synthetic data generation (SDG) pipeline that aims to generate challenging scenarios to facilitate downstream tasks such as perception and driving policy training. Powering this pipeline is *Cosmos-Drive*, a suite of models specialized from NVIDIA Cosmos-1 world foundation model [1] for the driving domain and are capable of controllable, high-fidelity, multi-view, and spatiotemporally consistent driving video generation. We showcase the utility of these models by applying *Cosmos-Drive-Dreams* to scale the quantity and diversity of driving datasets with high-fidelity and challenging scenarios. Experimentally, we demonstrate that our generated data helps in mitigating long-tail distribution problems and enhances generalization in downstream tasks such as 3D lane detection, 3D object detection and driving policy learning. We open source our pipeline toolkit, dataset and model weights through the NVIDIA's Cosmos platform. Project page: https://research.nvidia.com/labs/toronto-ai/cosmos_drive_dreams.

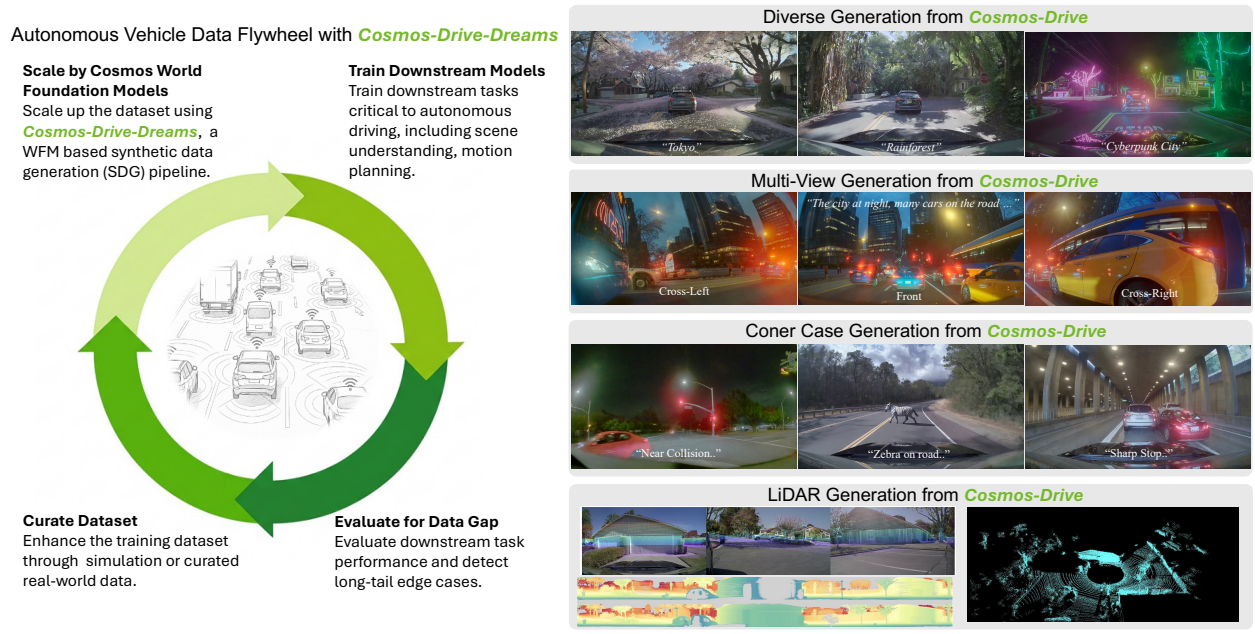


Figure 1: **Left:** Autonomous Vehicle Data Flywheel enabled by Cosmos-Drive-Dreams. The cycle illustrates a continuous feedback loop for improving autonomous driving models with synthetic data generation. **Right:** *Cosmos-Drive* generates high-quality and diverse synthetic videos with multi-view and LiDAR modality support.

[1]*: Equal contribution. Only the core contributors are listed. The full list of contributors can be found in Appendix A of this paper.

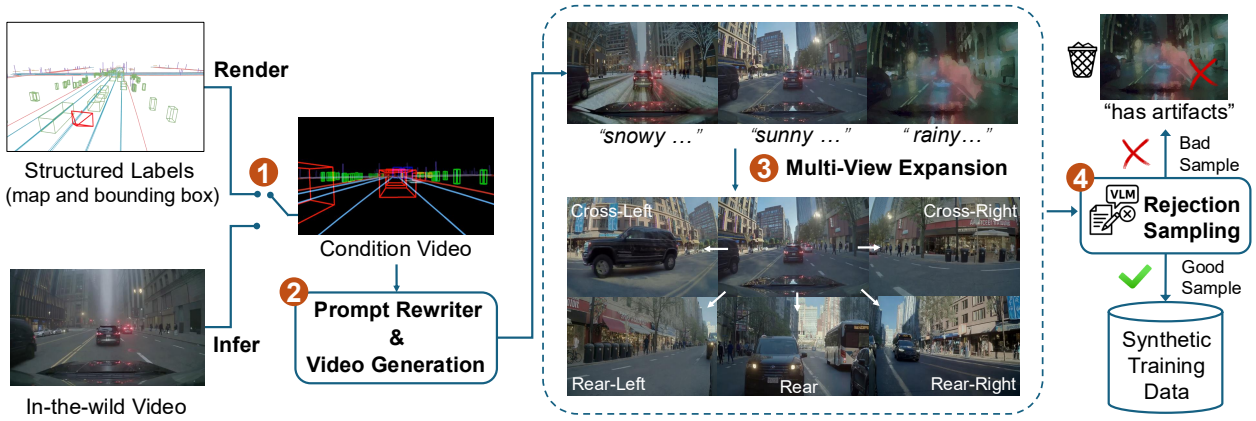


Figure 2: **Overview of our *Cosmos-Drive-Dreams* pipeline.** Starting from either structured labels or in-the-wild video, we generated pixel-aligned HDMap condition video (Step ❶). Then we leverage a prompt rewriter to generate diverse prompts and synthesize single-view videos (Step ❷). Each single-view video is then expanded into multiple views (Step ❸). Finally, a Vision-Language Model (VLM) filter performs rejection sampling to automatically discard low-quality samples, yielding a high-quality, diverse SDG dataset (Step ❹).

1. Introduction

The advancement of physical AI—such as autonomous driving platforms and embodied AI agents— is fundamentally driven by the ability of deep learning models to interpret, reason about, and act within complex real-world environments. Autonomous driving in particular stands out as one of the most data-hungry domains. Yet, creating large-scale, multi-view, temporally consistent driving datasets with fine-grained semantic annotations is expensive. The challenge is further compounded by the need to model long-tail and safety-critical scenarios [4, 6, 18, 24, 45, 53], such as sudden pedestrian crossings, erratic vehicle behavior, unusual road layouts, or extreme weather conditions. The rarity of these events in the real world makes it difficult to collect them at scale, while they are essential for training robust downstream AV perception or policy models: models trained purely on naturally occurring data often struggle to generalize to these corner cases [24, 29, 40], undermining the reliability of autonomous systems in high-stakes, real-world deployment.

Synthetic data represents a promising solution to the problem of corner-case data scarcity. Crafting 3D digital scenes using physical engines [10, 11, 26, 46, 49] offers high flexibility and customizability, but it is both costly and challenging to scale. Recent reconstruction-based methods [9, 43, 44, 48, 51, 57, 59, 67] can render photo-realistic images for novel views; however, their outputs are heavily tied to the original observations and cannot create entirely new scenes. While these methods can produce certain corner cases by modifying traffic scenarios, their non-generative nature limits their ability to model novel and challenging cases, such as extreme weather conditions or illumination changes.

We introduce *Cosmos-Drive*, a suite of video generation models that specialize NVIDIA Cosmos-1 world foundation model [1] to the driving domain, and *Cosmos-Drive-Dreams* — a scalable data pipeline that leverages these models to generate synthetic data for downstream AV task learning.

We build *Cosmos-Drive* on top of the generalist Cosmos-1 models [1] by adapting and post-training to the driving domain, and make them available through the Cosmos platform under the naming of *Cosmos-[ModelName]-Sample-AV*. *Cosmos-Drive* models unlock four key capabilities (See Fig. 3 and § 2): (1) Ability to generate videos that accurately match scene layouts and desired ego car trajectories by conditioning video generation on structured inputs (HDMap, 3D cuboids, text and optional LiDAR depth), ensuring precise, geometry-aware control. (2) Ability to generate multi-view videos via a new single-view-to-multi-view video generation model that can expand up to six view-consistent videos simultaneously; (3) Ability to autolabel HDMap layouts, cuboids and LiDAR-style depth from in-the-wild driving videos, opening up the possibility of re-simulating a recorded scenario for which maps, cuboids or LiDAR depth are hard to obtain, such as videos from the internet; (4) To further enhance Cosmos WFM as a comprehensive neural simulator for physical AI, we have extended its capabilities to support high-quality LiDAR generation.

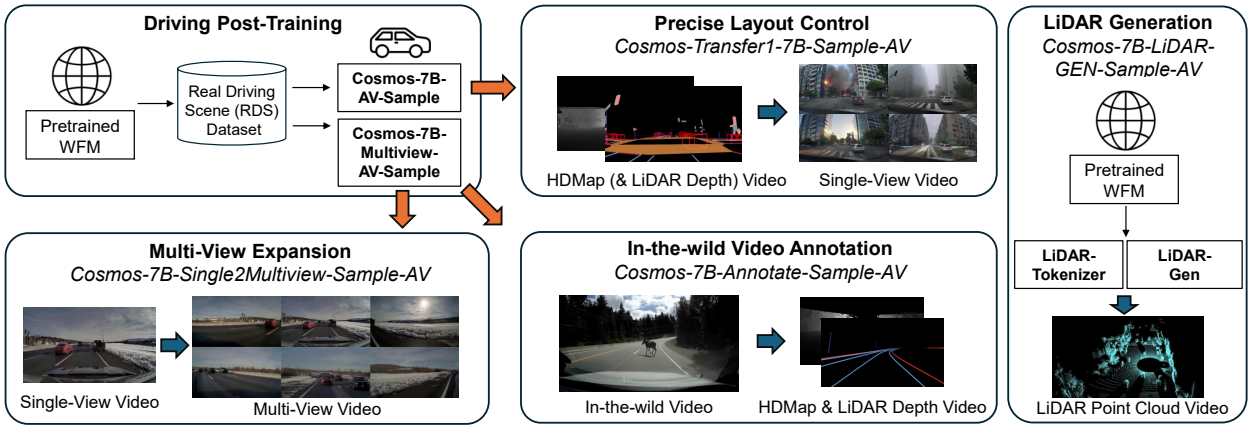


Figure 3: ***Cosmos-Drive*'s model suite. Top Left:** We begin with a pretrained world foundation model (WFM) and post-train it on RDS dataset to obtain driving-specific WFMs. This model is further post-trained into three models, which constitute *Cosmos-Drive*. **Top Right:** Precise layout control model (*Cosmos-Transfer1-7B-Sample-AV*), which generates single-view driving videos from HDMap and optional LiDAR depth videos; **Bottom Left:** Multi-view expansion model (*Cosmos-7B-Single2Multiview-Sample-AV*), which synthesizes consistent multi-view videos from a single view; **Bottom Right:** In-the-wild video annotation model (*Cosmos-7B-Annotate-Sample-AV*), which predicts HDMap and depth from in-the-wild driving videos. **Right:** LiDAR generation model (*Cosmos-7B-LiDAR-GEN-Sample-AV*), which synthesizes high-quality LiDAR points conditioned on HDMap or RGB images.

In *Cosmos-Drive-Dreams* (Fig. 2), we leverage the ability of *Cosmos-Drive* to amplify driving scenes and create variations of recorded or authored scenarios by modifying text descriptions. The generated results are gathered into high-quality synthetic datasets. We demonstrate that our synthetic data improves the performance of downstream tasks such as 3D lane detection, 3D object detection and policy learning, especially in challenging scenarios, and continues to provide measurable gains even when augmenting a large-scale real-world dataset.

In Summary: *(1)*. We present *Cosmos-Drive-Dreams*, a pipeline that leverages foundation world models to generate synthetic data for autonomous driving, addressing the long-tail challenge by leveraging world pretrained knowledge. *(2)* We release *Cosmos-Drive*, a suite of post-trained models for autonomous driving, along with post-training and inference code, a toolkit for customization, and a diverse synthetic driving dataset. *(3)* We demonstrate that synthetic data generated by *Cosmos-Drive-Dreams* enhances AV perception and policy learning, especially on challenging scenarios.

2. *Cosmos-Drive*: Cosmos Model Suite for AV

We post-train Cosmos-1 World Foundation Models (WFM) [1] for driving scenarios. This includes Cosmos text-to-video models *Cosmos-7B-Sample-AV* and *Cosmos-7B-Multiview-Sample-AV*, and dense conditional models: *Cosmos-Transfer1-7B-Sample-AV*, *Cosmos-7B-Single2Multiview-Sample-AV*, and *Cosmos-7B-Annotate-Sample-AV*. As shown in Fig. 3, *Cosmos-Transfer1-7B-Sample-AV* synthesizes single-view driving videos based on HDMap and prompts (Step ❷ in Fig. 2) and *Cosmos-7B-Single2Multiview-Sample-AV* extends single-view videos into multi-view videos (Step ❸ in Fig. 2). The *Cosmos-7B-Annotate-Sample-AV* model supports annotating in-the-wild videos with HDMap and LiDAR depth (Step ❶ in Fig. 2). In addition, the *Cosmos-7B-LiDAR-GEN-Sample-AV* model can generate high-quality LiDAR points from RGB images or directly from the HDMap conditions. We introduce these models in detail in the following subsections.

2.1. Background: Cosmos WFM and RDS Dataset Series

Cosmos WFM. Cosmos-1 [1] is a collection of generalist WFMs for Physical AI that can be post-trained into customized world models for downstream applications. In particular, *Cosmos-7B-Text2World* is a 7B

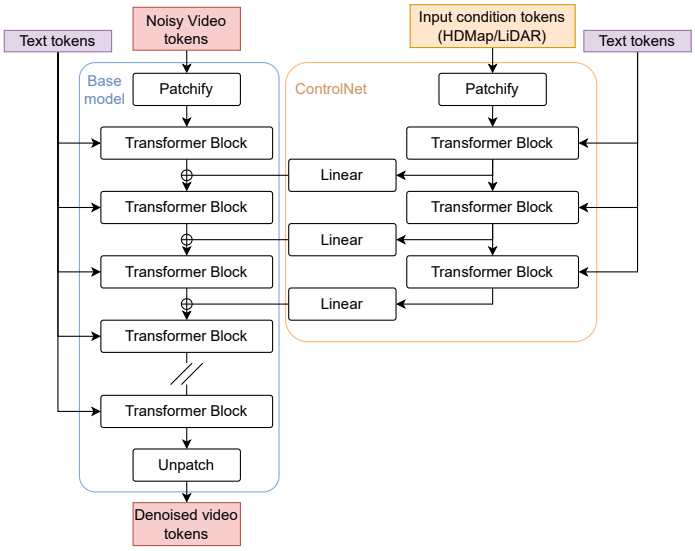


Figure 4: **Architecture diagram of *Cosmos-Transfer1-7B-Sample-AV*.** We adopt DiT architecture [39] for the generation model.

text-to-video diffusion model based on the DiT architecture [39], trained on tens of millions of physically grounded video clips. Cosmos-Transfer1 [38] extends Cosmos video diffusion models by incorporating multiple ControlNets [68], including segmentation map control, Canny edge control, depth control, and blur video control.

Driving Datasets for Post-Training. We utilize the following driving-specific datasets: **Real Driving Scene (RDS)** [1] dataset comprises approximately 3.6 million 20-second 6-view video clips (equivalent to approximately 20,000 hours of data) captured using an NVIDIA internal driving platform. Each clip is recorded from six camera views. This dataset was selected from a large labeled data corpus to match a target distribution of data attributes, including vehicle density, weather, illumination, ego car speed, and behavior. **Real Driving Scene HQ (RDS-HQ)** [38] dataset comprises 750 hours of high-quality 30-FPS 6-view driving video clips, complete with HDMap and 3D cuboid annotations and timestamp-aligned LiDAR point clouds.

2.2. Driving-Specific Post-Training from Cosmos WFM.

To effectively leverage the capabilities of WFMs for driving video generation, we post-train *Cosmos-7B-Text2World* on the ego-centric driving dataset, **RDS**. For frontal-view video generation, we retain the original architecture and fine-tune *Cosmos-7B-Text2World* on front-view videos from the **RDS** dataset using a batch size of 64 for 140k steps with a small learning rate of 1.5×10^{-5}. For multi-view video generation, we extend *Cosmos-7B-Text2World* into *Cosmos-7B-Multiview-Sample-AV* by post-training on six-view videos. For a detailed description of *Cosmos-7B-Multiview-Sample-AV*, we refer the reader to Section 6.3 of Cosmos-1 [1].

This post-training enhances the model's ability to understand and synthesize driving scenes across diverse environments and conditions. From this foundation, we introduce a suite of mechanisms for precise layout control, extensibility and annotation in *Cosmos-Transfer1-7B-Sample-AV* (§ 2.3), *Cosmos-7B-Single2Multiview-Sample-AV* (§ 2.4) and *Cosmos-7B-Annotate-Sample-AV* (§ 2.5), and train on the fine-grained annotations and high-quality videos of **RDS-HQ**, as described in the following sections.

Figure 5: **Precise layout control model (*Cosmos-Transfer1-7B-Sample-AV*)** generates diverse and rare scenarios with the same HDMap but different text prompts, such as: *The video captures a street scene during the day with a steady rain falling...; The scene unfolds in a chaotic environment as a fire engulfs the houses on either side of the street...; The scene is beautifully lined with blossoming sakura...*

2.3. *Cosmos-Transfer1-7B-Sample-AV*: Driving Video Generative with Precise Layout Control

We show the overall architecture of *Cosmos-Transfer1-7B-Sample-AV* in Fig. 4. To make generated videos useful for training layout-related perception models such as 3D lane detection and 3D bounding box detection, the generated content must align precisely with the underlying scene layout and motion. To enable this, we train ControlNet [68] models based on *Cosmos-7B-Sample-AV* conditioned on structured inputs. Specifically, we draw 3D cuboids, lane lines, road boundaries, poles, crosswalks, road markings, traffic lights, and traffic signs in a city-scale 3D space, then render them into 2D video frames using the ego-camera's intrinsic and extrinsic parameters. We refer to this rendered input as the HDMap Video (First Column of Fig. 5), and the corresponding ControlNet model as ***Cosmos-Transfer1-7B-Sample-AV* [HDMap]**. HDMap provides precise information about the relative pose between the ego-camera and the road geometry, enabling the HDMap ControlNet to simulate different ego-vehicle trajectories and to generate rare or safety-critical driving scenarios, such as emergency vehicle encounters, sharp turns, and complex merging behaviors. Additionally, to enable more fine-grained control over the generated videos, we train a LiDAR-based video ControlNet (Second Row of Fig. 8), referred to as ***Cosmos-Transfer1-7B-Sample-AV* [LiDAR]**. To maintain flexibility, we train separate branches for HDMap and LiDAR conditioning, which can be optionally fused at inference time [38].

As illustrated in Fig. 5, our *Cosmos-Transfer1-7B-Sample-AV* model can generate diverse driving videos conditioned on HDMaps and text prompts, including realistic scenes under heavy rain and surreal scenarios such as a car driving through a street engulfed in fire.

Training Details. Since *Cosmos-7B-Sample-AV* base model is already post-trained on driving, during training time, the base diffusion model is kept frozen. We train the control branches using a batch size of 64, a learning rate of 5e-5, and a total of 25K steps.

Figure 6: **Multi-view expansion model (*Cosmos-7B-Single2Multiview-Sample-AV*)** generates multi-view videos with different weather prompts such as "*...sunny day...*" or "*...snow storm...*".

2.4. *Cosmos-7B-Single2Multiview-Sample-AV*: Single-View to Multi-View Expansion

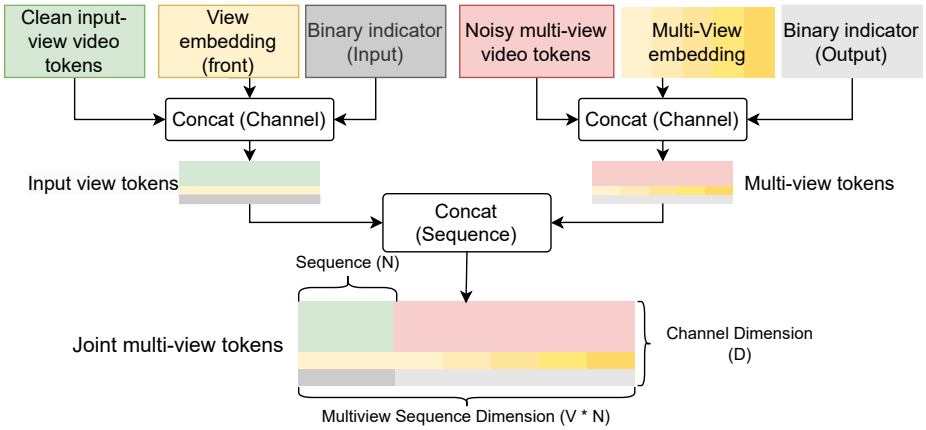
We build an extended video generation model that predicts fixed-rig multi-view videos from frontal-view video, multi-view prompts, and (optionally) multi-view HDMap and/or LiDAR inputs. Fig. 6 demonstrates the ability of our *Cosmos-7B-Single2Multiview-Sample-AV* to synthesize multi-view videos.

To benefit from the knowledge present in WFMs, we fine-tune this *Cosmos-7B-Single2Multiview-Sample-AV* model from *Cosmos-7B-Multiview-Sample-AV*. To convert *Cosmos-7B-Multiview-Sample-AV* into a view-extension model, we use clean video tokens rather than noisy tokens as input for the conditioning view, and append an indicator to each view for whether it is an input or an output (see Fig. 7a). We sequentially concatenate tokens from the input view with the tokens being denoised, such that the self-attention layers within all DiT blocks jointly attend over all views. Cross-attention with per-view text embedding is performed independently for each view, as illustrated in Fig. 7b. This design, together with the per-view view-embedding, allows us to flexibly control which views to condition on and which views are generated. We change the resolution of this model to 576×1024 and train on video clips of 57 frames. To accommodate this increase in context length during training, we select the input view and any 3 of the 5 output views, for a total of 4 views to form a batch. During inference, all 5 output views can be generated simultaneously. To achieve precise scene-layout control in multi-view generation, we also train a multi-view ControlNet based on HDMap or LiDAR inputs. This ControlNet also uses the multi-view architecture described above.

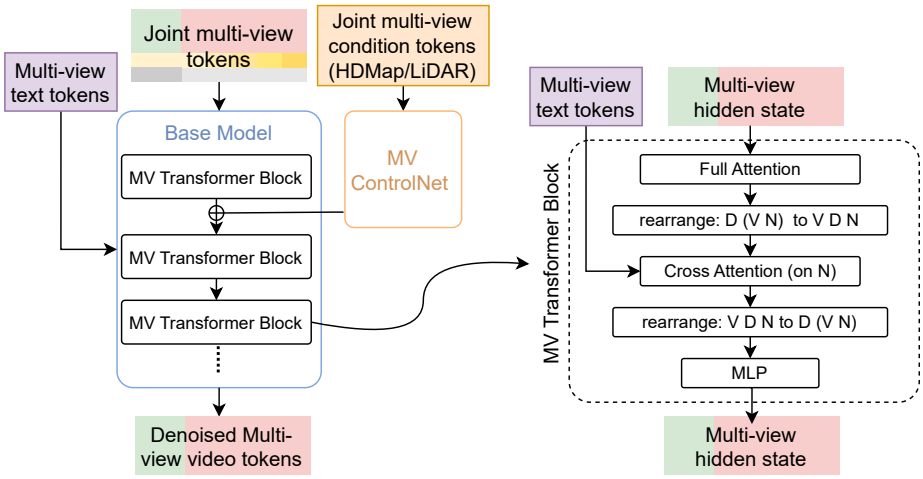
Training Details. We train this model on 2,000 hours of internal driving footage, with batch size 32 and learning rate 5×10^{-5} for 30K steps. Then, we train the HDMap and LiDAR ControlNets on multi-view videos in the RDS-HQ dataset while freezing the weights of the base diffusion model, using also batch size 32 and a learning rate 5×10^{-5} for 20K steps.

2.5. *Cosmos-7B-Annotate-Sample-AV*: In-the-Wild Video Annotation Model

To further expand the utility of our generative driving model, we introduce a novel annotation pipeline within *Cosmos-7B-Annotate-Sample-AV*, as illustrated in Fig. 3 and Fig. 8. Instead of relying on manually annotated datasets or complex sensor setups, *Cosmos-7B-Annotate-Sample-AV* leverages readily available in-the-wild driving videos, significantly enhancing data accessibility and diversity. Our video annotation process converts Internet-sourced single-view RGB videos into corresponding HDMap and LiDAR information. This conversion is crucial for scenarios where direct HDMap and LiDAR data acquisition is impractical, enabling the generation



(a) ***Cosmos-7B-Single2Multiview-Sample-AV* input structure.** We combine clean input-view tokens, noisy multi-view tokens, global per-view embedding tokens, and a binary indicator for input/output together as input to the DiT.



(b) **Architecture of *Cosmos-7B-Single2Multiview-Sample-AV*.** Adapted from *Cosmos-Transfer1-7B-Sample-AV* with transformer blocks replaced by MV transformer blocks.

Figure 7: **Overview of *Cosmos-7B-Single2Multiview-Sample-AV*:** input structure and model architecture.

of rich semantic representations of driving scenes from minimal inputs. Specifically, *Cosmos-7B-Annotate-Sample-AV* takes as input in-the-wild video frames and predicts the spatial layout, including accurate HD maps with road geometries, lane lines, and 3D cuboid for dynamic objects. Concurrently, it infers depth maps to reconstruct the scene's 3D structure, which closely resembles genuine LiDAR scans. In practice, the automated video annotation provided by *Cosmos-7B-Annotate-Sample-AV* dramatically reduces the dependence on costly sensor suites and labor-intensive manual labeling. It empowers scalable generation and augmentation of multi-modal datasets using vast quantities of publicly available driving footage, effectively addressing data scarcity in autonomous driving development.

The architecture of *Cosmos-7B-Annotate-Sample-AV* is shown in Fig. 9. The model is conditioned on the input video by concatenating the video tokens with the denoising tokens along the channel dimension. To support multiple output modalities (e.g. HDMap, LiDAR), we introduce a trainable context embedding for each modality, which replaces the text tokens in the cross-attention layers. Since inferring HDMaps and LiDAR given an input video no longer requires a text prompt, this substitution allows the model to focus purely

Figure 8: **In-the-wild video annotation model (*Cosmos-7B-Annotate-Sample-AV*)** enables corner-case data generation from unlabeled videos. **1st row:** In-the-wild videos from the Nexar Dashcam Collision Prediction Dataset [36] ; **2nd row:** HDMap and LiDAR depth generated by *Cosmos-7B-Annotate-Sample-AV*; **3rd-4th rows:** Videos generated by *Cosmos-Transfer1-7B-Sample-AV* using the same condition videos but different text prompts, such as: *A residential street during golden hour...; A snowy street with a moose crossing the road...; A car turning right into a rainbow-painted clown car...*

on the modality-specific conditioning. This design enables flexible extension to new modalities by adding corresponding embeddings and allows training mixed datasets that may lack certain outputs.

Fig. 8 highlights the generalizability of our pipeline to Internet-scale, in-the-wild videos without relying on curated or annotated datasets. **Training Details.** Similar to before, we fine-tune this model from *Cosmos-7B-Sample-AV* to leverage the knowledge from pretrained WFM. We train the model on the RDS-HQ dataset using a batch size of 32, a learning rate of 2×10^{-5}, for a total of 30k training steps.

2.6. *Cosmos-7B-LiDAR-GEN-Sample-AV*: Specializing Cosmos WFM for LiDAR Generation

To further enhance Cosmos WFM as a comprehensive neural simulator for physical AI, we have extended its capabilities to generate high-quality LiDAR data, a critical component for safety-critical autonomous driving. This advancement positions Cosmos as a more complete and versatile tool for simulating complex real-world driving scenarios. Our contributions include careful LiDAR preprocessing, specialized tokenizer fine-tuning, and the development of HDMap or images-conditioned LiDAR generative model.

LiDAR Data Representation. We represent LiDAR data as a range map. This conversion essentially transforms points (x, y, z) from a Euclidean coordinate system into a spherical one (r, θ, ϕ) by calculating the

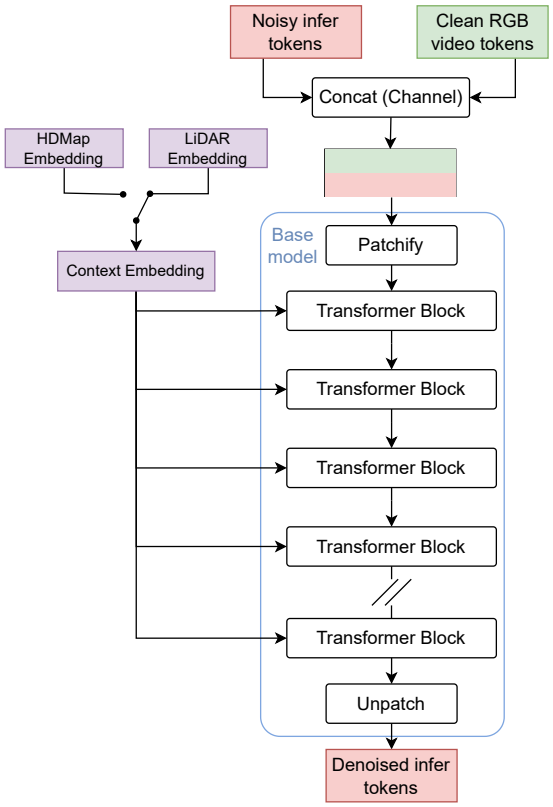


Figure 9: **Architecture of *Cosmos-7B-Annotate-Sample-AV* is adapted from *Cosmos-7B-Sample-AV*.** Text embeddings are replaced with an output embedding.

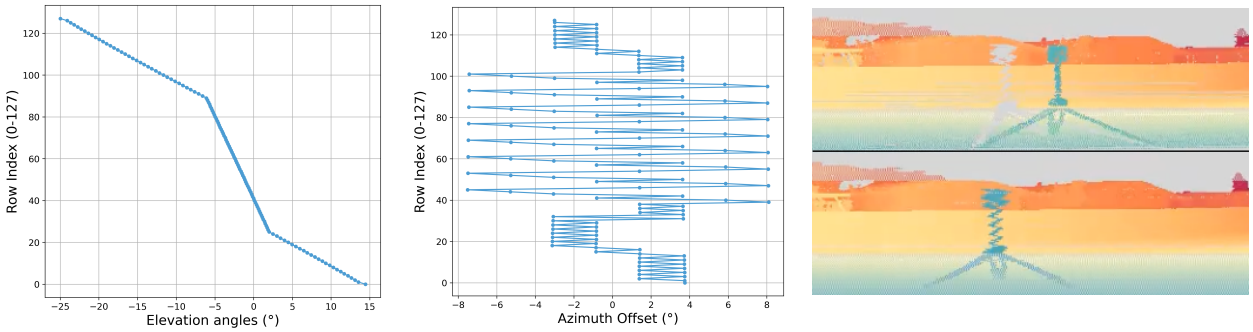
radial distance r, elevation angle θ, and azimuth angle ϕ as:

$$r = \sqrt{x^2 + y^2 + z^2}, \quad \phi = \arctan 2(y, x), \quad \theta = \arcsin(z/r). \tag{1}$$

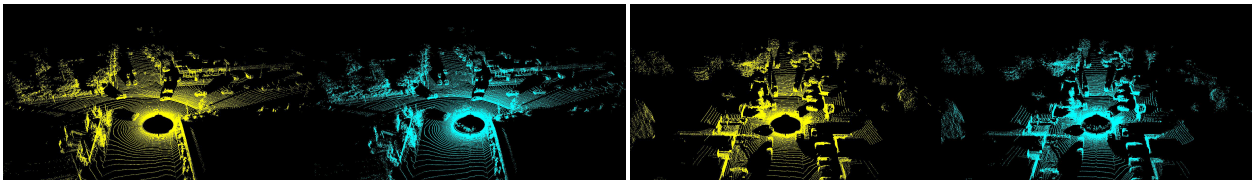
The standard representation assumes that all LiDAR points are captured simultaneously from a fixed origin at $(0, 0, 0)$, which is inaccurate in dynamic driving scenarios. In practice, LiDAR sensors take approximately 0.1 seconds to complete a full 360-degree scan while the vehicle is in motion. This motion-compensated data are typically captured at 10 Hz. Therefore, to accurately project the LiDAR point cloud onto a range map, the motion compensation must be reversed, which in turn requires estimating the timestamp of each point—this depends on a detailed understanding of the LiDAR sensor model (see Fig. 10a). Naively projecting the LiDAR point cloud without accounting for non-uniform elevation angles and detailed azimuth angle profiles results in artifacts such as ghost pixels (gray pixels) as shown in the top row of the right panel in Fig. 10a. To address this, two sensor components are considered:

(1) As shown in the left panel, the 128 LiDAR beams are vertically distributed in a non-uniform manner—sparse at the extremes and dense in the middle. We use this **elevation angle profile** to assign each point to the appropriate row in the range image by matching its vertical angle θ to the closest sensor elevation angle.

(2) At each emission timestamp, the 128 beams have slightly different azimuth angles, creating a zig-zag pattern (see middle panel). We employ this **azimuth angle profile** to correct each point's horizontal angle ϕ, enabling accurate assignment to the appropriate column in the range map. This column index subsequently serves as an estimate of the point's measurement timestamp.



(a) **Nvidia LiDAR sensor model.** The left panel shows the 128 elevation angles are sparse at extremes ($-25°$ and $15°$) and dense in the middle ($-6°$ to $2°$). The right panel shows the range maps obtained from incorrect (top row) and correct (bottom row) sensor models.



(b) **Qualitative results of Cosmos LiDAR tokenizer.** Comparison between LiDAR input and tokenizer reconstruction output visualized as point clouds.

Figure 10: **LiDAR Sensor Model and Tokenizer Results.**

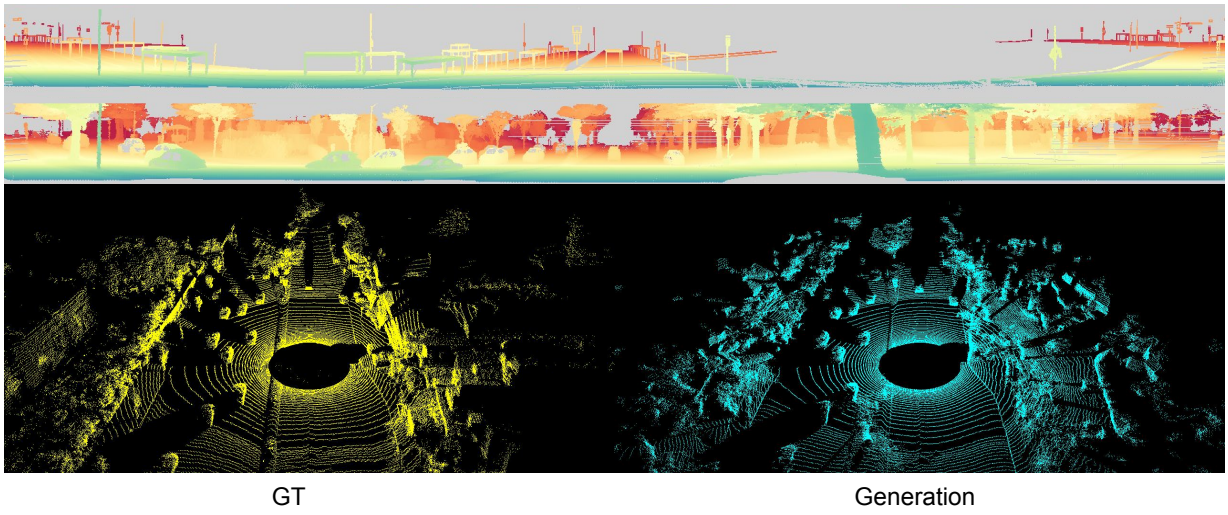
With these corrections, accurate modeling of the sensor produces clean, artifact-free range maps, as illustrated in the bottom row of the right panel in Fig. 10a.

Moreover, in practice, we post-process the LiDAR range map to better accommodate its unique modality. Due to the inherent sensitivity of depth values to the precision, we use the *float32* data type in the tokenizer and defer casting to *bfloat16* for diffusion. To address the spatial-temporal sparsity caused by LiDAR's scanning patterns and low frame rate, we introduce redundancy by repeating each row of the range map four times. Finally, we clip the range values to the 2nd and 98th percentiles and normalize them to the range $[-1, 1]$.

LiDAR Generative Model. We first fine-tune the Cosmos image and video tokenizers to range map modality without altering the network architecture. The qualitative results of our fine-tuned Cosmos LiDAR tokenizer are present in Fig. 10b.

We developed two primary models for LiDAR generation: (1) **HDMap-conditioned LiDAR generation**: This model guides LiDAR generation by channel-wise concatenating the clean latent representation of HDMap data with the noisy latent representation of LiDAR data. This ensures precise alignment with the scene's layout from HDMaps as well as high controllability of the moving agents through the bounding box control. To obtain HDmap and bounding box control for LiDAR, we first sample 3D points from the parametric representation of HDmap and bounding boxes, then undo the motion compensation before projecting them to the range map. (2) **RGB-conditioned LiDAR generation** : This is formulated as a multi-view generation task akin to *Cosmos-7B-Single2Multiview-Sample-AV* model, this model utilizes RGB images from front, rear-left, and rear-right cameras as conditioning inputs to synthesize corresponding LiDAR data. Qualitative results can be found in Fig. 11a and Fig. 11b.

Building upon our HDMap-conditioned LiDAR generation model, we incorporate a Prompt Rewriter (see § 3.1) to adapt text captions based on weather conditions. This enables the model to synthesize LiDAR data under varying weather scenarios, even when the HDMap input remains unchanged. This capability



(a) **HD Map-Conditioned LiDAR Generation. First Row:** HD map conditions visualized in range map view. **Second Row:** Generated LiDAR range map. The range map is color-coded far near, with missing pixel values filled in gray. **Third Row Left:** Ground truth LiDAR point cloud in bird's-eye view. **Third Row Right:** Generated LiDAR visualized as point cloud.

(b) **RGB image-conditioned LiDAR generation. First Row:** Generated LiDAR range map. **Second Row:** Conditional images from the front, and rear-right cameras, overlaid with the projected generated LiDAR point cloud. Generated LiDAR visualized as point cloud.

Figure 11: **LiDAR generation results under different conditions.** We visualize generation quality from HDMAP, RGB image, and generated LiDAR point cloud.

is particularly important, as LiDAR responses are highly sensitive to weather—e.g., rain can reduce point returns due to energy absorption by wet surfaces. As illustrated in Fig. 12, our model effectively captures these weather-dependent characteristics.

3. *Cosmos-Drive-Dreams*: Scalable Synthetic Data Generation Pipeline for AV

In this section, we introduce *Cosmos-Drive-Dreams*, a synthetic data generation pipeline utilizing Cosmos model suite for AV from § 2 to create high-quality, multi-view videos. As illustrated in Fig. 2, our pipeline involves rendering pixel-aligned HDMap projections from structured labels like HDMaps and 3D cuboids or in-the-wild driving videos (Step ❶). These serve as control inputs for guiding video generation. To increase scenario diversity, an LLM-based prompt rewriter generates varied textual descriptions from the initial video caption; and a HDMap conditioned video generation model is applied to synthesize high-quality single-view videos (Step ❷). The resulting videos are expanded into a multi-view format (Step ❸), capturing consistent perspectives like front, rear, and side views, which are essential for autonomous driving applications. Finally, a Vision-Language Model (VLM) filter conducts automated rejection sampling (Step ❹) to eliminate poor or unrealistic outputs. The final product is a high-fidelity, diverse synthetic driving dataset tailored to rare and safety-critical scenarios. The raw inputs (3D cuboids, maps, and trajectories) combined with the generated videos can be directly

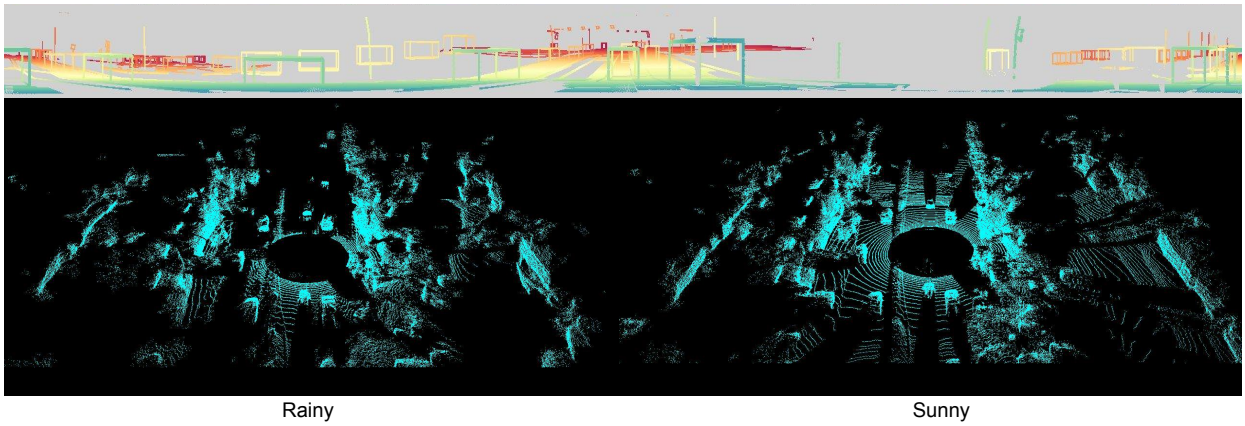


Figure 12: **HD map-conditioned LiDAR generation under different weather prompts. Top:** HD map conditioning input. **Bottom:** Generated LiDAR visualized as point cloud. The LiDAR generated with the rainy prompt shows increased ray drop due to adverse weather conditions.

Figure 13: **Discarded samples during rejection sampling** on synthetic Waymo Open dataset.

utilized as augmented data for downstream tasks.

3.1. Prompt Rewriter

We utilize a Large Language Model, *i.e.*, `Qwen2.5-7B-Instruct` [65], to automatically rewrite the initial video caption into diverse prompts conditioned on various environmental contexts, including different weather conditions (e.g., foggy, rainy, snowy) and times of day (e.g., morning, golden hour, nighttime). Note that given an original video caption, the model is instructed to modify only the environment-related attributes while preserving the core content and spatial layout of the source video. This ensures the layout-level description in the new prompt remains consistent with HDMap video condition. The system prompt for `Qwen2.5-7B-Instruct` to do the prompt rewriting is as follows:

> You are a prompt optimization specialist. Your task is to rewrite user-provided input prompts into high-quality English descriptions by modifying specific *temporal* or *environmental* details, while preserving the core content and actions of the original scene.
>
> There are two types of rewrites:
>
> 1. Time of Day: Change the time setting in the caption, including Golden hour (with long shadows), Morning, and Night.
> 2. Environment/Weather: Change the weather condition or environment (style) in the caption.

Figure 14: **Discarded samples during rejection sampling** on synthetic RDS-HQ dataset.

Requirements:

- Keep the scene and actions the same (e.g., a car driving down a highway should still be a car driving down a highway).
- Change only the details related to time or environment as instructed.
- Ensure the rewrite matches the new condition (e.g., no mention of sun glare in a foggy or snowy version).

We provide a prompt rewriter example below, with the given original prompt: "*The video captures a nighttime drive through a suburban area. The road is illuminated by streetlights and vehicle headlights, creating a well-lit path. Palm trees line both sides of the street, adding a tropical feel to the environment. Several cars are parked along the right side of the road, and a few are driving in the opposite direction. The sky is overcast, with no visible stars or moon, indicating it is likely late evening or night. The overall scene is quiet and peaceful, with no pedestrians or significant traffic.*"

Rainy: The video captures a nighttime drive through a suburban area during a light rain. The road is illuminated by the glow of streetlights and vehicle headlights, creating a soft, shimmering path. Palm trees line both sides of the street, their leaves gently swaying in the light breeze and casting intermittent shadows on the wet pavement. Several cars are parked along the right side of the road, their reflections dancing on the puddles. A few cars are driving in the opposite direction, their lights cutting through the gentle rainfall. The sky is overcast, with no visible stars or moon, suggesting it is late evening or night. The overall scene is quiet and peaceful, with no pedestrians and only minimal traffic, as the raindrops create a soothing background sound.

Snowy: The video captures a nighttime drive through a suburban area blanketed in snow. The road is illuminated by streetlights and vehicle headlights, casting soft, glowing halos in the gentle snowfall. Tall palm trees line both sides of the street, their fronds lightly dusted with snow, adding a serene and almost surreal tropical feel to the wintry environment. Several cars are parked along the right side of the road, their outlines softened by the falling snow, and a few others are driving cautiously in the opposite direction, their taillights twinkling through the flakes. The sky is overcast, with no visible stars or moon, creating a hushed and tranquil atmosphere. The overall scene is quiet and peaceful, with no pedestrians or significant traffic, as the snow muffles any sound and casts a calming blanket over the neighborhood.

Foggy: The video captures a nighttime drive through a suburban area enveloped in a thick fog. The road is illuminated by dim streetlights and vehicle headlights, creating a soft, ethereal glow that barely penetrates the misty veil. Palm trees line both sides of the street, their silhouettes barely discernible through the fog, lending an eerie yet tranquil atmosphere. Several cars are parked along the right side of the road, their outlines barely visible, and a few others are driving slowly in the opposite direction, their lights cutting through the fog like ghostly apparitions. The sky is overcast, with no stars or moon visible, adding to the sense of isolation and calm. The overall scene is quiet and peaceful, with no pedestrians and only minimal traffic, as if the world has paused for this hazy moment.

3.2. Rejection Sampling

To ensure visual realism and consistency in the generated videos, we introduce a rejection sampling module as part of the SDG scaling process (see Fig. 2). This module filters out artifacted or implausible outputs using a VLM-based visual assessment agent. We adopt `Qwen2.5-VL-7B-Instruct` [2], a vision-language model capable of evaluating short video clips, as an automated filter. The model is prompted with task-specific instructions to detect visual anomalies that break continuity or realism, such as object disappearance, shape distortion, and rendering glitches. Videos flagged by the VLM are discarded. This automated process enables scalable, human-free quality control while maintaining dataset integrity. The system prompt for `Qwen2.5-VL-7B-Instruct` to do the rejection sampling is as follows:

> You are responsible for judging the visual quality and consistency of driving videos generated by a computer program. You will be presented with a short video, and your task is to identify whether there are any obvious visual artifacts that break realism or continuity.
>
> The artifacts you are looking for may include:
>
> (1) Object disappearance
> (2) Shape distortion
> (3) Texture or rendering glitches
> (4) Object consistency issues
> (5) Background artifacts
> (6) Temporal discontinuity
> (7) Other major visual errors
>
> Instructions:
>
> - Think out loud, noting your observations and rationale.
> - Then, classify the video as either Clean or Artifacted.
> - Minor imperfections are acceptable as long as realism is preserved.

We show the rejected samples in Fig. 13 and Fig. 14.

3.3. *Cosmos-Drive-Dreams* on the RDS-HQ and Waymo Open Datasets

With our *Cosmos-Drive-Dreams*, we can generate diverse driving video variants for real-world autonomous driving datasets. Here we apply *Cosmos-Drive-Dreams* to RDS-HQ and Waymo Open [50] for synthetic data generation. Specifically, we render the HDMap inputs using structured labels from RDS-HQ and Waymo Open [50] datasets and employ the prompt rewriter module to modify the original video captions, conditioning the generation process on a diverse set of attributes: `golden-hour`, `morning`, `night`, `rainy`, `snowy`, `sunny`, and `foggy`. Each synthesized video has a resolution of 704×1280 and a duration of 121 frames at 30 FPS. From these frontal-view examples, we further extend the first 57 frames of them with *Cosmos-7B-Single2Multiview-Sample-AV* into multi-view videos, each with a resolution of 576×1024. We apply the rejection sampling module and discard 3% of the generated samples, while the remaining high-quality videos will serve as augmented training data for downstream tasks.

4. *Cosmos-Drive-Dreams* Pipeline Evaluation

To evaluate the effectiveness of *Cosmos-Drive-Dreams*, we conducted experiments across a range of downstream autonomous driving tasks, including 3D lane detection, 3D object detection, and driving policy learning. As described in § 3.3, we generate 7× synthetic clips for each real clip. However, not all synthetic clips are used in every training epoch. In all experiments, we define R_{s2r} as the ratio of synthetic clips used per training epoch,

Table 1: **3D lane detection performance with *Cosmos-Drive-Dreams*.** Our pipeline significantly improves the 3D lane detection performance over baseline and Albumentations [5]. "Cate. Acc." means category accuracy.

Dataset	Waymo Open Dataset						RDS-HQ (2k)							
Metric	All		Extreme Weather		Night		All		Rainy		Foggy		Night	
	F1	Cate. Acc.	F1	Cate. Acc.	F1	Cate. Acc.	F1	Cate. Acc.	F1	Cate. Acc.	F1	Cate. Acc.	F1	Cate. Acc.
Original w/o SDG	0.428	0.847	0.378	0.858	0.402	0.842	0.532	0.852	0.458	0.821	0.524	0.844	0.547	0.867
Albumentations $R_{s2r} = 0.5$	0.446	0.846	0.389	0.829	0.412	0.825	0.548	0.860	0.483	0.834	0.548	0.854	0.563	0.871
Albumentations $R_{s2r} = 1$	0.444	0.848	0.369	0.840	0.405	0.840	0.546	0.857	0.483	0.833	0.551	0.854	0.558	0.867
Cosmos-Drive-Dreams $R_{s2r} = 0.5$	0.448	0.853	**0.417**	0.860	0.431	0.862	0.564	**0.876**	0.500	0.848	**0.573**	**0.871**	0.575	0.883
Cosmos-Drive-Dreams $R_{s2r} = 1$	**0.451**	**0.855**	0.404	**0.875**	**0.455**	**0.878**	**0.566**	0.871	**0.506**	**0.851**	0.572	0.867	**0.581**	**0.885**

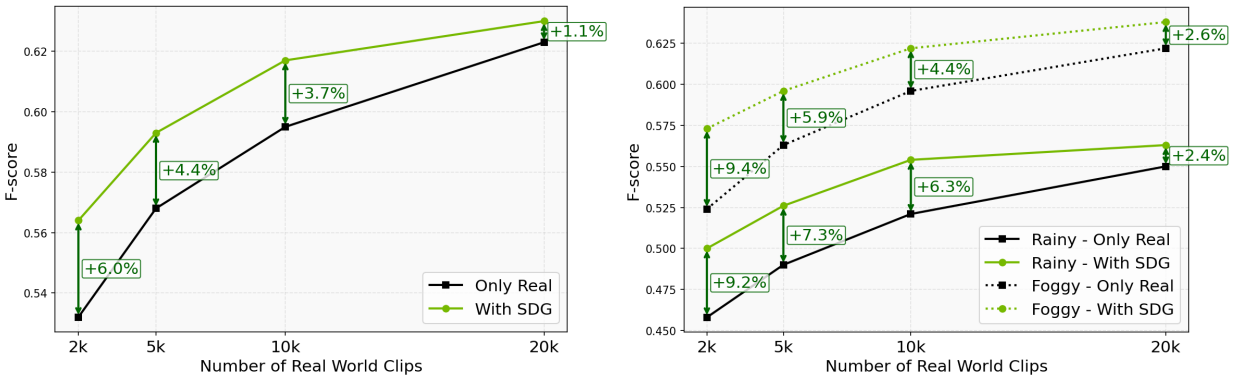


Figure 15: ***Cosmos-Drive-Dreams* improves F-score across varying amounts of real-world training data on 3D Lane Detection task**. SDG clips are mixed with real clips using a ratio of $R_{s2r} = 0.5$. **Left:** Results on testing dataset. Under all weather conditions, SDG consistently improves detection performance across varying amounts of real-world training data, with the most significant gain (+6.0%) observed in the low-data regime (2k clips). **Right:** Results on the extreme weather subset of the testing dataset. In more challenging settings (Rainy and Foggy), the benefits of SDG are even more pronounced—showing gains of up to +9.4% under foggy conditions with only 2k real clips. This highlights SDG's effectiveness in enhancing model robustness, particularly under adverse or underrepresented conditions.

randomly sampled from *Cosmos-Drive-Dreams* 's output (§ 3.3). For example, $R_{s2r} = 1$ indicates that an equal number of synthetic and real clips are used in each training epoch, while "w/o SDG" refers to training with real data only. Our findings show a consistent increase in performance across these tasks when synthetic data was incorporated into the training process, especially for corner cases like extreme weather and nighttime scenarios. We detail three downstream tasks in the following sections.

4.1. 3D Lane Detection

Dataset Setup. We use RDS-HQ subsets of different sizes for lane detection model training. For example, RDS-HQ (2k) indicates 2k clips are used for baseline training. All experiments share a disjoint testing set, containing 2.8k clips that feature diverse and challenging weather conditions. We also conduct 3D lane detection experiments on the Waymo Open Dataset. We filter out clips with poor HDMap alignment, resulting in 504 clips for training and 144 clips for validation. To test the corner-case performance, we follow the corner-case split for the validation set as defined by OpenLane [7], which selects clips with extreme weather or nighttime conditions to better evaluate model performance under challenging scenarios. To bridge the distribution gap for SDG on the Waymo dataset, we additionally finetune another version of *Cosmos-Transfer1-7B-Sample-AV* (§ 2.3) on it. For a fair comparison, in *Cosmos-Drive-Dreams*'s experiments, we randomly sample SDG data generated using HDMap annotations that correspond to the same ground-truth clips used in baseline training.

We use a transformer-based monocular 3D lane detector, LATR [32], for evaluating 3D lane detection tasks.

Table 2: **3D object detection performance with *Cosmos-Drive-Dreams*.** When applied to augment training set (setting Ⓐ), *Cosmos-Drive-Dreams* improves the detection performance in general and extreme weather conditions.

Dataset	**Waymo**			**RDS-HQ (2k)**				**RDS-HQ (20k)**				**RDS-HQ (2k, MV)**			
LET-APL ↑	All	Ext. Wea.	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night
w/o SDG	0.446	0.468	0.399	0.190	0.176	0.167	0.178	0.307	0.289	0.272	0.299	0.221	0.211	0.212	0.214
$R_{s2r} = 0.5$	**0.459**	0.477	**0.418**	0.210	0.199	0.193	0.196	0.320	0.305	0.287	0.307	**0.228**	**0.217**	**0.217**	**0.217**
$R_{s2r} = 1$	0.439	**0.478**	0.410	**0.213**	**0.203**	**0.195**	**0.202**	**0.328**	**0.315**	**0.289**	**0.319**	0.219	0.213	0.208	0.213
LET-APH ↑	All	Ext. Wea.	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night
w/o SDG	0.613	0.632	0.572	0.285	0.262	0.256	0.271	0.446	0.416	0.408	0.434	0.322	0.308	0.311	0.312
$R_{s2r} = 0.5$	**0.622**	**0.640**	**0.591**	0.319	0.302	0.295	0.293	0.463	0.441	0.425	0.443	**0.334**	**0.318**	**0.319**	**0.319**
$R_{s2r} = 1$	0.614	**0.640**	0.589	**0.325**	**0.308**	**0.298**	**0.303**	**0.475**	**0.452**	**0.432**	**0.461**	0.324	0.313	0.309	0.317
LET-AP ↑	All	Ext. Wea.	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night	All	Rainy	Foggy	Night
w/o SDG	0.625	0.642	0.580	0.299	0.280	0.265	0.285	0.459	0.432	0.418	0.448	0.333	0.319	0.320	0.324
$R_{s2r} = 0.5$	**0.634**	**0.650**	**0.599**	0.332	0.316	0.305	0.311	0.477	0.456	0.435	0.460	**0.345**	**0.330**	**0.328**	**0.330**
$R_{s2r} = 1$	0.627	**0.650**	0.598	**0.337**	**0.323**	**0.308**	**0.321**	**0.489**	**0.468**	**0.442**	**0.478**	0.335	0.324	0.317	**0.330**

For the Waymo Open dataset, images are resized to 960 $\times$ 720 resolution. Training spans 24 epochs with a batch size of 32, using a cosine-scheduled learning rate decaying from 2×10^{-4} to 1×10^{-5}. For the RDS-HQ dataset series, images are rectified to a 960 $\times$ 540 pinhole camera, with the model trained for 15 epochs using the same batch size and learning rate.

For the baseline comparison, we employ Albumentations [5], an image augmentation library widely used in other works, such as YOLO v11/v12 [23, 52]. This library simulates conditions such as rain, fog, and nighttime using pixel-level transformations. We report the F1-score and category accuracy of 3D lane detection trained on the Waymo Open Dataset and RDS-HQ (2k) in Tab. 1. The results show that *Cosmos-Drive-Dreams* significantly improves detection performance in cases where Albumentations provides limited benefit. We also provide detailed statistics for the different splits (different weather and time) in the test set. Notably, *Cosmos-Drive-Dreams* achieves F1 score improvements of 10.4% and 9.4% on the rainy and foggy test splits, respectively—scenarios that are underrepresented in the training data—highlighting the effectiveness of our SDG pipeline. Furthermore, *Cosmos-Drive-Dreams* delivers a 6.4% gain on the overall test set, demonstrating its general dataset augmentation capability.

We further analyze *Cosmos-Drive-Dreams* with different scales of real-world training clips. As shown in Fig. 15, we conduct experiments with varying real clip counts (2k, 5k, 10k, 20k) while fixing synthetic data and real data ratio to be 0.5 at each training epoch ($R_{s2r} = 0.5$). *Cosmos-Drive-Dreams* improves over real-clips only baselines, particularly under extreme weather conditions—precisely where our SDG prompt rewriter is designed to target (§ 3.1). In the foggy and rainy settings (right panel), models trained with SDG significantly outperform those using only real data. Even at 20k real-world clips, *Cosmos-Drive-Dreams* gains +2.4% improvement on F1-score in the foggy condition. Notably, while the performance gap between models with and without SDG narrows as the amount of real data increases, this convergence reflects a saturation effect rather than a limitation of SDG. There remains substantial room to scale SDG further—both by expanding the number of generated clips and by exploring more diverse prompt rewrites to enrich scenario variability.

We show visualizations of 3D lane detection on the left of Fig. 17. In these rainy, foggy, and nighttime cases, adding synthetic data significantly helps the model perceive the scene more accurately.

4.2. 3D Object Detection

Apart from annotating from real video clips, HDMap condition used by *Cosmos-Transfer1-7B-Sample-AV* can be generated through traffic models, authored using platforms such as the NVIDIA Omniverse, or acquired from third-party providers. These 3D annotations can be rendered in the camera configuration of real vehicles.

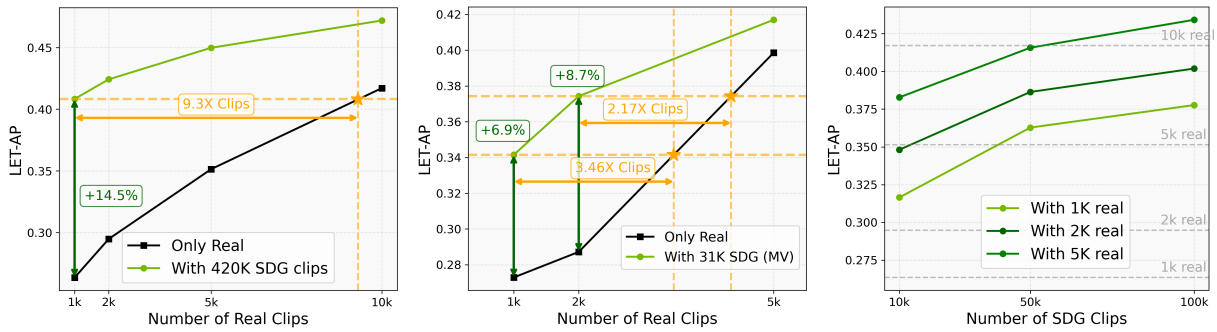


Figure 16: ***Cosmos-Drive-Dreams* generated data improves detection performance at various scales on RDS-HQ-HL** (setting ❷). **Left:** Adding *Cosmos-Drive-Dreams* generated data from RDS-HQ to RDS-HQ-HL train-set significantly improves performance on RDS-HQ-HL test-set. Performance improvement from the addition of SDG data to 1K real clips ($+14.5\%$) is comparable to $9.3\times$ increase in real data. **Center:** *Cosmos-Drive-Dreams* generated multiview data significantly improves performance on RDS-HQ-HL in the multi-view setting. **Right:** Performance increases correlate steadily with the amount of SDG data added for all tested amounts of real data.

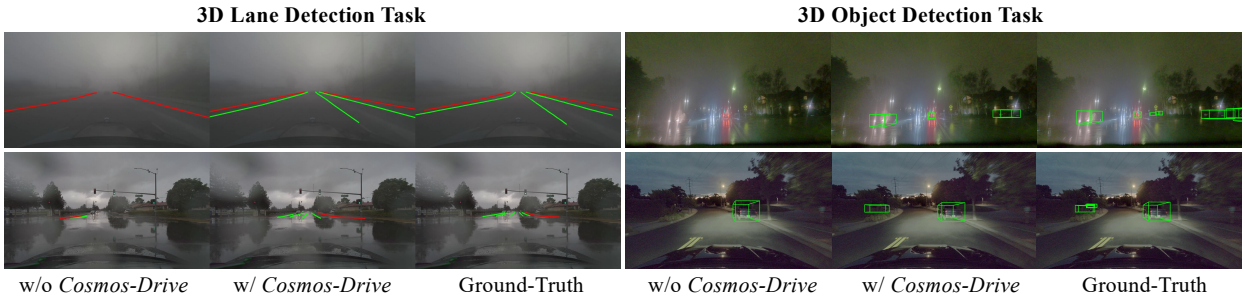


Figure 17: **Visualization of 3D lane detection and 3D object detection** on RDS-HQ (2k) dataset after adding *Cosmos-Drive-Dreams*. In lane detection, green is the laneline and red is the road boundary.

Cosmos-Drive-Dreams serves as a bridge between various sources and the real world, enabling transfer from synthetic labels to realistic driving clips. To demonstrate this transferability and flexibility, we conduct experiments in two settings: ❶ *Cosmos-Drive-Dreams* augments real clips in the training set (following the setting in § 4.1); ❷ *Cosmos-Drive-Dreams* performs SDG using a set of HDMap labels disjoint from real clips. Furthermore, we introduce a multi-view setting, which shares the same clips as above, while utilizing all six cameras for full field of view (FOV) coverage around the ego vehicle.

Dataset Setup. Following § 4.1, we apply *Cosmos-Drive-Dreams* on RDS-HQ and Waymo Open Dataset to generate synthetic clips. When performing SDG for RDS-HQ, we use three subsets of HDMaps of increasing scales (2k, 20k, 52k) to produce synthetic data subsets (16k, 160k, 420k). The first two subsets correspond to the training sets of RDS-HQ as used in § 4.1 and will be used in setting ❶, while the 52k HDMap, 420k generated data set is used for setting ❷. For the multi-view setting, we use two subsets of HDMaps (2k, 11k), corresponding to 10k and 31k synthetic data, for setting ❶ and ❷ respectively. Videos are rectified to 960×640 pixels for Waymo Open and 960×540 for RDS-HQ. For the purposes of setting ❷, we also curate a dataset (10k clips for training, 2k clips for testing) with human-labeled bounding boxes from RDS-HQ that are **disjoint** from the auto-labeled clips of RDS-HQ used above for SDG. We call this dataset RDS-HQ-HL (human-labeled).

We choose BEVFormer [27], a temporal, transformer-based 3D object detector, to evaluate the effectiveness of SDG in camera-based 3D object detection. The model is trained for 12 epochs using a cosine learning rate schedule with a batch size of 16.

For experiment ❶, we report LET-AP, LET-APH, and LET-APL metrics [21] for the vehicle category in Tab. 2. Consistent with the findings of the 3D lane detection task, the results demonstrate that incorporating synthetic

Table 3: **LiDAR-based 3D object detection performance with *Cosmos-Drive-Dreams*.** *Cosmos-Drive-Dreams* improves the overall detection performance.

Dataset	RDS-HQ (1k)				RDS-HQ (2k)			
	mAP	Car	Bus	Truck	mAP	Car	Bus	Truck
w/o SDG	0.240	**0.371**	0.155	0.195	0.289	**0.402**	0.225	0.240
w/ SDG	**0.250**	0.366	**0.181**	**0.203**	**0.297**	0.399	**0.248**	**0.246**

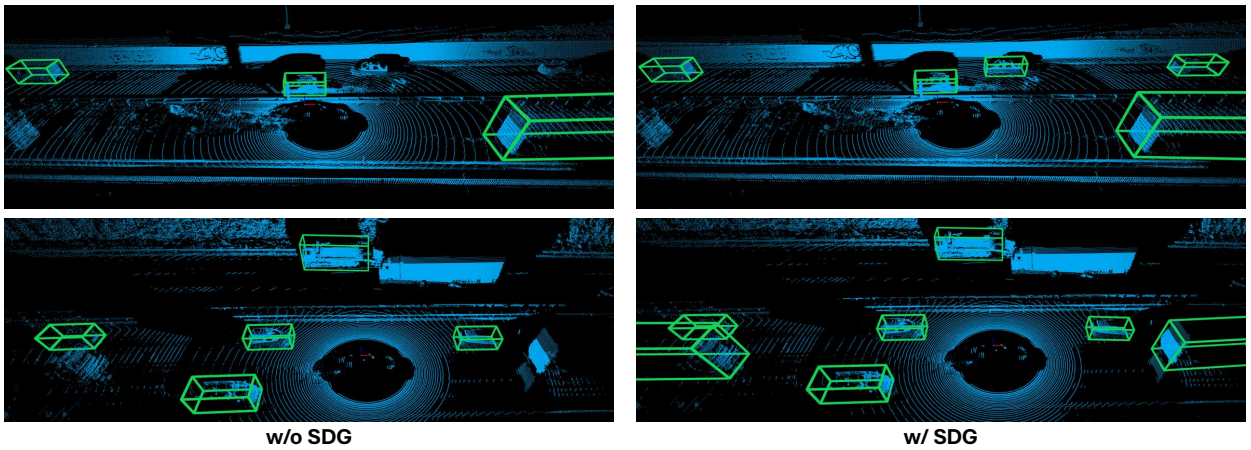


Figure 18: **Visualization of LiDAR-based 3D object detection** on RDS-HQ (2k) dataset after adding *Cosmos-Drive-Dreams*.

data generated by *Cosmos-Drive-Dreams* significantly improves object detection performance at all dataset scales, both for monocular and for multi-view settings. The most notable gains are observed in corner-case scenarios like rainy and foggy scenes, while improvement on the general scenes remains substantial, further highlighting the augmentation capability of *Cosmos-Drive*.

For experiment ⓫, we find that incorporating the SDG data generated from RDS-HQ significantly improves performance compared to using only RDS-HQ-HL. In Fig. 16-Left and Fig. 16-Center, we show experiments of using all available SDG data while varying the amount of RDS-HQ-HL clips in the training set, both for single view and for multi views. *Cosmos-Drive-Dreams* SDG data bridges the gap between the auto-labeled RDS-HQ and the human-labeld RDS-HQ-HL, leveraging the former as input to improve performance on the later. For example, the performance of 1K RDS-HQ-HL clips + *Cosmos-Drive-Dreams* SDG data is approximately comparable in performance to 93K RDS-HQ-HL clips when interpolated linearly. In Fig. 16-Right, we vary the amount of single-view SDG data and combine it with varying amounts of real clips. The results show that scaling up SDG via *Cosmos-Drive-Dreams* correlates positively with performance, and that further scaling up SDG can likely offer even more benefits.

Fig. 17-right visualize the object detection results. In challenging conditions like rain, fog, and nighttime, incorporating synthetic data significantly enhances the model's object detection accuracy.

4.3. LiDAR-based 3D Object Detection

Dataset Setup. We use RDS-HQ (1k) and RDS-HQ (2k) for training. From the HDMap annotations corresponding to each training subset, we generate LiDAR datasets (150k frames in total) with the pipeline introduced in § 2.6, and use it for joint training with RDS-HQ (1k) and RDS-HQ (2k). In each clip, we uniformly sample 20 frames of LiDAR point clouds. For testing, we use the same test set as § 4.1.

We choose Transfusion-LiDAR [3] as the LiDAR-based 3D object detector. Transfusion-LiDAR uses sparse 3D convolution, BEV convolution and transformer-based detection head for 3D object detection. For joint training

RDS-HQ (135k) subsets					
minADE ↓	**30k**	**50k**	**70k**	**100k**	**135k**
$R_{s2r} = 0.0$	1.307	1.182	1.116	1.060	1.026
$R_{s2r} = 0.5$	1.226	1.146	1.077	1.027	0.986
$R_{s2r} = 1.0$	1.202	1.124	1.065	1.008	0.980
$R_{s2r} = 2.0$	1.189	1.109	1.046	1.007	0.979
$R_{s2r} = 3.0$	**1.183**	**1.089**	**1.041**	**1.003**	**0.976**

RDS-HQ (135k) scenario evals			
minADE ↓	**All**	**VRUs**	**left-turn**
$R_{s2r} = 0.000$	0.920	0.774	1.269
$R_{s2r} = 0.007$	**0.917**	0.750	1.265
$R_{s2r} = 0.013$	0.919	0.697	1.203
$R_{s2r} = 0.020$	0.923	**0.692**	**1.186**

Table 4: **Policy learning performance**. Left: *Cosmos-Drive-Dreams* improves the trajectory prediction accuracy on RDS-Bench[Policy]. Right: Small amount of targeted SDG data can improve predictions in corner cases (RDS-Bench[VRU/left]).

with the SDG dataset, we apply JiSAM [8] to better incorporate SDG LiDAR dataset and train the same 3D detector [3]. All models are trained for 20 epochs with a cosine learning rate schedule. For evaluation metric, we use average precisions (APs) for different categories and mAP for overall results.

Results are shown in Tab. 3. We find that the addition of SDG dataset generally improves detection performance for different real data subsets over the baseline. We visualize detection results in Fig. 18, which shows that adding SDG data helps improve model recall for larger and more distant objects.

4.4. Policy Learning

For the policy learning task, we create training subsets of between 30k and 135k clips of 4s length from the RDS-HQ dataset. For evaluation, we use a separate, non-overlapping dataset, named RDS-Bench[Policy]. The test set consists of 3k RDS-HQ clips balanced for different driving scenarios. All training sets are augmented using a variable amount of SDG data from § 3.3, which was created by running *Cosmos-Drive-Dreams* on 50k RDS-HQ clips and rewritten with 7 different weather conditions. For each dataset, we sample a number of SDG clips corresponding to up to 3 times ($R_{s2r} = 3.0$) of the dataset size. We ensure to only select SDG clips which are based on RDS-HQ clips contained in the respective training sets.

Next, we train a transformer-based policy model [37] on all training subsets. We use a simplified policy that is not goal/route conditioned. This model takes as input 9 historical frames of frontal-view camera and predicts the vehicle's trajectory. We evaluate the policies on RDS-Bench[Policy] using a minADE metric with a 5s future horizon. As shown in Tab. 4 and Fig. 20 (left), adding synthetic data consistently improves predictive accuracy for any given amount of real-world clips. The effect is particularly large for smaller training set sizes. However, even the largest size (135k) still benefits from SDG data, even though the SDG data was created based on a much smaller subset of 50k RDS-HQ clips. This suggests that the added diversity from SDG helps to reduce overfitting and boosts generalization, even when using the same real-world source data. Conversely, Fig. 20 (center) shows data that SDG can lower the data demand for achieving the target performance metric. To e.g. achieve minADE = 1.35, we can train on a dataset of 60k real-world clips, or 50k real-world clips with $R_{s2r} = 0.5$, or 39k clips with $R_{s2r} = 3.0$.

Beyond random augmentation, SDG can also be used for more targeted improvements. We noted that due to how we curated our training dataset, the policy would benefit from improvements on specific scenarios, such as those involving vulnerable road users (VRUs) or sharp left turns. As these events are underrepresented in our RDS-HQ (135k) source dataset we curated, we sourced 1000 clips for VRUs and sharp left turns from unlabeled RDS data. We use 400 clips to generate the targeted RDS-Bench[VRU/left] eval sets, and the rest as training data (RDS[VRU/left]). As these clips lacked HDMaps, we first employed *Cosmos-7B-Annotate-Sample-AV* to generate them, and then used *Cosmos-Transfer1-7B-Sample-AV* to generate 3k SDG clips with weather augmentation (SDG[VRU/left]). We train policies on a training set consisting of RDS-HQ(135k), RDS[VRU/left], and subsets of SDG[VRU/left]. As illustrated in Fig. 20 (right), we observe a significant improvement in performance on the targeted RDS-Bench[VRU/Left] evaluation set even though SDG data only constitutes a small percentage ($\leq$ 2 %) of the training set. Importantly, this targeted augmentation does not negatively impact the general

Figure 19: **Policy learning visualization.** Policies trained with synthetic data (yellow) can perform more accurate predictions than the baseline (green), particularly in scenarios with poor visibility or uncommon objects. The ground truth trajectory is shown in white.

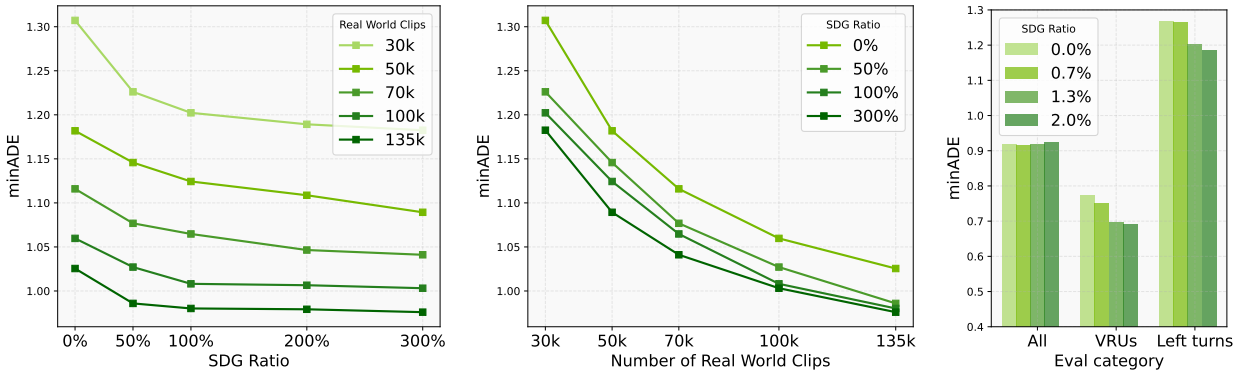


Figure 20: **Policy learning.** Left: Given an amount of real-world clips, adding SDG data improves trajectory prediction accuracy (minADE on RDS-Bench[Policy], lower is better). Center: Less real-world data is needed to reach a target minADE. Right: Adding a small amount of targeted SDG data can improve performance for certain corner cases (RDS-Bench[VRU/left], without hurting overall driving performance.

driving performance, as measured on the RDS-Bench[Policy] set.

Together, these results demonstrate that SDG can be used to extract more performance from an existing set of real-world clips, while also allowing to address targeted weaknesses by leveraging small amounts of unlabeled data with *Cosmos-7B-Annotate-Sample-AV*.

5. Related Work

Synthetic data generation for autonomous driving has undergone significant evolution, with methodologies increasingly focused on improving realism, controllability, and scalability. Broadly, these efforts can be grouped into three categories: physics-based simulation, reconstruction-based rendering, and generative approaches. Each category offers unique trade-offs and capabilities. In this section, we review representative works in each category and highlight key challenges that continue to motivate further research.

5.1. Physics-based Simulators

Physics-based simulators [10, 11, 26, 46, 49] rely on graphics engines and physical models to create synthetic environments and simulate sensor data, including RGB, LiDAR, and depth streams. These platforms offer structured control over environment layout, weather conditions, traffic flow, and agent behavior, which makes them especially useful for benchmarking perception and planning algorithms under repeatable conditions.

CARLA [10] has become a widely adopted open-source platform, offering a rich library of urban environments, sensor modalities, and autopilot agents. MetaDrive [26], on the other hand, emphasizes flexibility and computational efficiency, making it suitable for reinforcement learning (RL) scenarios with tight performance constraints. AirSim [49], built atop Unreal Engine [11], delivers high-fidelity sensor modeling for both aerial

and terrestrial vehicles, enabling pixel-accurate simulations for photorealistic vision tasks. [16, 17] are two representative works that use physics-based simulation to augment existing Lidar data under foggy and snowy conditions, using attenuation and back-scattering effects.

Despite their flexibility and fine-grained control, physics-based simulators often suffer from a lack of photorealism and require extensive engineering effort to support long-tail scenario diversity. Moreover, scaling these environments to cover rare or edge-case conditions (e.g., snowy nights, multi-agent collisions) remains labor-intensive. These limitations have motivated interest in data-driven alternatives capable of achieving broader visual diversity and richer interactions with less manual intervention.

5.2. Reconstruction-based Simulators

Reconstruction-based approaches attempt to bridge the reality gap by converting real-world driving logs into renderable neural scenes [9, 42, 43, 51, 57, 59, 63, 66, 67, 69]. These methods leverage neural reconstruction architectures such as Neural Radiance Fields (NeRF) [34] or 3D Gaussian Splatting [22] to reconstruct photorealistic environments from sensor logs. The resulting scenes enable high-fidelity, closed-loop simulations without requiring extensive manual design.

UniSim [30, 67] exemplifies this approach by transforming a single real-world driving log into a dynamic simulation environment, complete with multi-view camera and LiDAR simulation. OmniRe [9] represents dynamic actors—such as pedestrians and vehicles—through neural scene graphs based on Gaussian splats. 3DGRUT [35, 63] add ray-tracing capabilities which are important for modeling distorted cameras and secondary lighting effects. On the LiDAR front, NFL [20] and DyNFL [61] combine the rendering power of neural fields with a detailed, physically motivated model of the LiDAR sensing process, enabling it to accurately reproduce key sensor behaviors.

While reconstruction-based simulators offer high realism, reusability and closed-loop performance, they remain constrained by the original viewpoint distribution captured in the logs. Out-of-distribution rendering—especially from novel perspectives—often results in visible artifacts or degraded quality. Methods such as Difix3D [62] and UniSim [67] address this limitation using image-based generative priors or discriminators. The reconstruction methods remain limited in their ability to modify scene attributes like lighting, texture, or weather, which hinders their effectiveness for generating diverse training scenarios. Works such as [28, 41, 58] are examples of new methods that address the relighting capabilities specifically.

5.3. Generation-based Simulators

Recent advances in generative modeling have unlocked new possibilities for scalable and diverse simulation [15, 25, 31, 54, 55, 64]. By learning from large video datasets, generative models—particularly those based on diffusion architectures—can synthesize realistic driving scenes that generalize beyond the limits of handcrafted or reconstructed environments.

Notable systems include Drive-WM [56], which integrates driving dynamics into a generative framework; Panacea [60], which focuses on structured video generation; and MagicDrive [12, 13, 14], which enables both 3D scene generation and targeted editing. Delphi [33] and GAIA [19] further advance this frontier by conditioning generation on trajectory and semantic inputs, enabling goal-directed synthesis.

A key limitation of these methods is their reliance on the underlying training distribution. Since rare scenarios (e.g., extreme weather, accidents) are poorly represented in real-world data, generative models often struggle to produce such events with high fidelity or frequency. To overcome this, GAIA-2 [47] significantly scales the training dataset to 14,000 hours of driving video, improving generative coverage and robustness.

Our work extends this line of research by building on the Cosmos World Foundation Model [1], a large-scale

Table 5: ***Cosmos-Drive*** Open-source summary.

	Type	Link
Cosmos-7B-Sample-AV (Sec. 2.1)	model	`base_model.pt` in Huggingface Link
Cosmos-7B-Multiview-Sample-AV (Sec. 2.1)	model	Huggingface Link
Cosmos-Transfer1-7B-Sample-AV (Sec. 2.3)	model	Huggingface Link
Cosmos-7B-Single2Multiview-Sample-AV (Sec. 2.4)	model	Huggingface Link
Cosmos-7B-Annotate-Sample-AV (Sec. 2.5)	model	Under review
Cosmos-7B-LiDAR-GEN-Sample-AV (Sec. 2.6)	model	Under review
Cosmos-Drive-Dreams Toolkit	SDG tool	Cosmos-drive-dreams-toolkits
RDS-HQ Dataset Subset	dataset	PhysicalAI-Autonomous-Vehicle-Cosmos-Drive-Dreams
Cosmos-Drive-Dreams Synthetic Dataset	dataset	PhysicalAI-Autonomous-Vehicle-Cosmos-Drive-Dreams

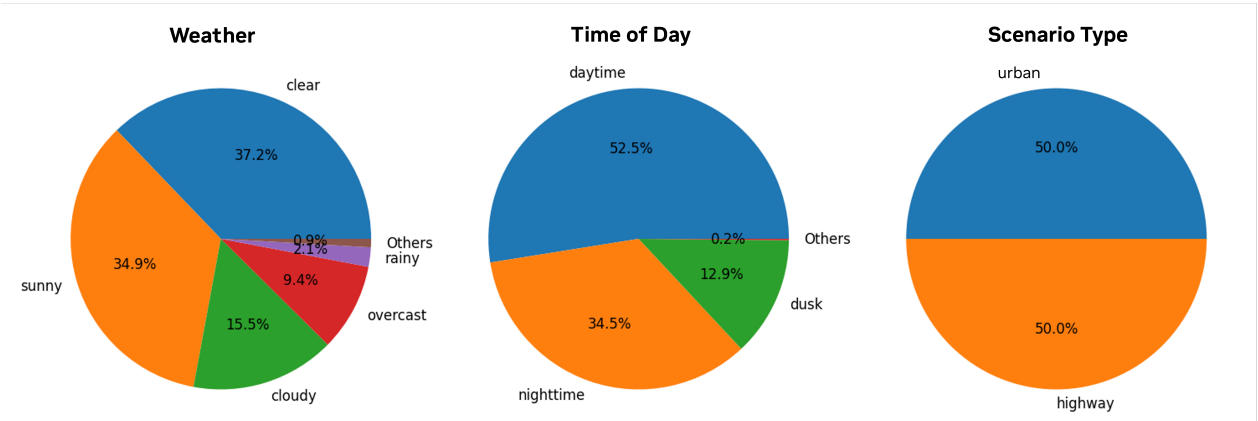


Figure 21: **Distribution for weather, time of day, and scenario of our RDS-HQ subset.**

generative video model pretrained on tens of millions of physical and AI-related videos. We post-train Cosmos-1 on 20,000 hours of carefully balanced driving footage to improve representation of long-tail conditions and region-specific phenomena. This approach enables realistic and diverse driving video generation at scale, offering a promising direction for simulation-driven development of autonomous systems.

6. *Cosmos-Drive-Dreams* Open-Source Summary

We summarize *Cosmos-Drive-Dreams* released models, dataset and toolkit in Tab. 5 with more details as follows.

6.1. Complementary Dataset

A Subset of RDS-HQ Dataset

We release 5,843 10-second clips from RDS-HQ Dataset and the corresponding labels (HD map, 3D objects, camera intrinsics and camera poses) in RDS-HQ format, where each attribute is stored in a tar file. The labels are stored in Hugging Face repository. The HD map label includes crosswalks, lane lines, lanes, poles, road boundaries, road markings, wait lines, traffic lights, and traffic signs. Here the lane lines, lanes, road boundaries, poles, and wait lines are represented as polylines; crosswalks and road markings are represented as polygons; and traffic lights and traffic signs are represented as 3D cuboids. We also provide the annotation for 3D objects in the scene. The category for 3D objects includes automobile, heavy truck, bus, train or tram car, trolley bus, other vehicle, trailer, person, stroller, rider, animal, and protruding object. The distribution for weather, time of day and scenario of this subset can be found in Fig. 21.

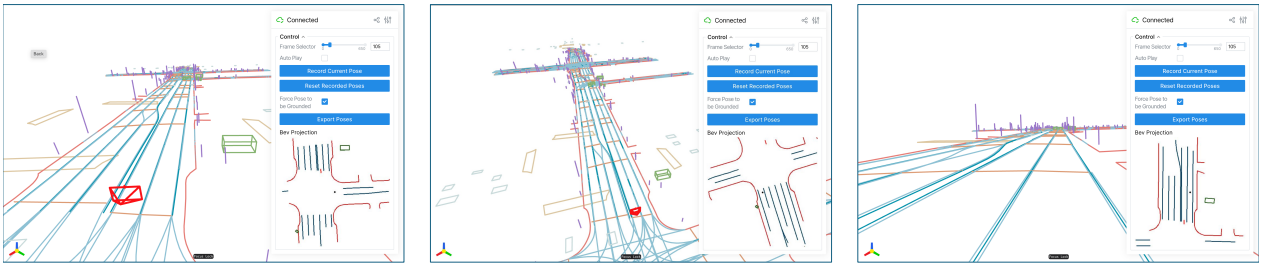


Figure 22: **Novel trajectory customization tool interface** in our toolkit. You can record your current camera pose at any frame and export the full trajectory. We also provide a Bird-eye-view (BEV) visualization to help you determine your exact position. The red frustum represents the original camera pose in the driving log.

Synthetic Dataset

We release a synthetic dataset built upon this subset in Hugging Face repository, consisting of 81,802 121-frame front-view video clips with `golden-hour`, `morning`, `night`, `rainy`, `snowy`, `sunny`, and `foggy` variants. We render their HDMap videos from structured label inputs and divide them into 121-frame chunks. Since real clips have approximately 300 frames, we will have two chunks for each real clip. We use `{clip_id}_{chunk_id}_{weather}` for the synthetic video naming. The first chunk (`chunk_id=0`) corresponds to 1-st to 121-st frame in the original clip; and the second chunk (`chunk_id=1`) corresponds to 122-nd to 242-nd frame in the original clip.

6.2. *Cosmos-Drive-Dreams* Toolkits

We provide a toolkit at Cosmos-drive-dreams-toolkits, offering multiple features to facilitate synthetic data generation. We highlight features of our toolkits as follows

1. **HDMap Video Rendering.** Our toolkit supports projecting 3D polylines, 3D polygons, and 3D cuboids onto the image plane using either a pinhole camera model or an f-theta camera model with data parallelism. The rendering results serve as condition videos for our model suite.
2. **Novel Trajectory Customization.** We provide a GUI tool for users to roam in 3D scenes. By recording camera pose at several key frames, the GUI tool will automatically interpolate a driving trajectory, which can be seamlessly used in **HDMap Video Rendering.** It enables users to design corner cases with fine-grained controls over the ego trajectory. We illustrate the GUI tool interface and how they create novel trajectories given an existing scene in Fig. 22.
3. **Third-party Dataset Support.** We provide an exemplary script to convert a third-party dataset (*e.g.* Waymo Open Dataset) into RDS-HQ format, which can be used to generate training data for Cosmos finetuning. It enables users to build their own *Cosmos-Drive-Dreams* pipeline using our pre-trained models or fine-tune them on their own datasets.

7. Conclusion

We introduced *Cosmos-Drive-Dreams*, a scalable synthetic data pipeline built upon *Cosmos-Drive* video generative models post-trained from Cosmos WFM. Our results show that the generated data enhances perception and policy learning, especially in long-tail cases. However, the reliance on computationally heavy diffusion models makes large-scale generation time- and resource-intensive, which we leave to future optimization efforts. These limitations will be addressed in the future work.

A. Contributors and Acknowledgements

A.1. Core Contributors

*: Equal Contribution, ◇: Corresponding Authors

- ***Cosmos-Drive* World Foundation Model Post-training**:
  Xuanchi Ren*, Tianshi Cao*, Amirmojtaba Sabour*, Tianchang Shen*, Jun Gao
- ***Cosmos-Drive-Dreams* Pipeline Development**:
  Xuanchi Ren, Yifan Lu, Tianshi Cao, Jay Zhangjie Wu
- ***Cosmos-Drive-Dreams* Downstream Tasks Evaluation**:
  Yifan Lu*, Ruiyuan Gao*, Tobias Pfaff*, Seung Wook Kim
- ***Cosmos-Drive-Dreams* Toolkit**:
  Yifan Lu, Xuanchi Ren, Tianshi Cao
- ***Cosmos-7B-LiDAR-GEN-Sample-AV* Post-Training**:
  Shengyu Huang, Laura Leal-Taixe
- ***Cosmos-7B-LiDAR-GEN-Sample-AV* Downstream Task Evaluation**:
  Runjian Chen, Shengyu Huang
- **Data Curation**:
  Yifan Lu, Xuanchi Ren, Tianchang Shen, Mike Chen
- **Architectural Design**:
  Sanja Fidler◇, Huan Ling◇

A.2. Contributors

- **Data Pipeline Support**:
  Yuchong Ye, Zhuohao (Chris) Zhang
- **Engineering Support**:
  Lyne Tchapmi, Mohammad Harrim, Pooya Jannaty
- **Solution Architect Partner Support**:
  John Shao, Yu Chen, Summer Xiao
- **Product Manager**:
  Aditya Mahajan, Matt Cragun

References

[1] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 1, 2, 3, 4, 21

[2] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. 14

[3] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1090–1099, 2022. 18, 19

[4] D. Bogdoll, S. Guneshka, and J. M. Zöllner. One ontology to rule them all: Corner case scenarios for autonomous driving. In *European Conference on Computer Vision*, pages 409–425. Springer, 2022. 2

[5] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020. 15, 16

[6] K. Chen, Y. Li, W. Zhang, Y. Liu, P. Li, R. Gao, L. Hong, M. Tian, X. Zhao, Z. Li, et al. Automated evaluation of large vision-language models on self-driving corner cases. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7817–7826. IEEE, 2025. 2

[7] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*, pages 550–567. Springer, 2022. 15

[8] R. Chen, W. Shao, B. Zhang, S. Shi, L. Jiang, and P. Luo. Jisam: Alleviate labeling burden and corner case problems in autonomous driving via minimal real-world data. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 6792–6801, June 2025. 19

[9] Z. Chen, J. Yang, J. Huang, R. de Lutio, J. M. Esturo, B. Ivanovic, O. Litany, Z. Gojcic, S. Fidler, M. Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 2, 21

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 2, 20

[11] Epic Games. Unreal engine. 2, 20

[12] R. Gao, K. Chen, Z. Li, L. Hong, Z. Li, and Q. Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024. 21

[13] R. Gao, K. Chen, B. Xiao, L. Hong, Z. Li, and Q. Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024. 21

[14] R. Gao, K. Chen, E. Xie, L. Hong, Z. Li, D.-Y. Yeung, and Q. Xu. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023. 21

[15] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger, J. Zhang, and H. Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 21

[16] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai, and L. Van Gool. LiDAR Snowfall Simulation for Robust 3D Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 21

[17] M. Hahner, C. Sakaridis, D. Dai, and L. Van Gool. Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 21

[18] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2(7), 2019. 2

[19] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 21

[20] S. Huang, Z. Gojcic, Z. Wang, F. Williams, Y. Kasten, S. Fidler, K. Schindler, and O. Litany. Neural lidar fields for novel view synthesis. 2023. 21

[21] W.-C. Hung, H. Kretzschmar, V. Casser, J.-J. Hwang, and D. Anguelov. Let-3d-ap: Longitudinal error tolerant 3d average precision for camera-only 3d detection, 2022. 17

[22] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. 21

[23] R. Khanam and M. Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024. 16

[24] K. Li, K. Chen, H. Wang, L. Hong, C. Ye, J. Han, Y. Chen, W. Zhang, C. Xu, D.-Y. Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European Conference on Computer Vision*, pages 406–423. Springer, 2022. 2

[25] P. Li, K. Chen, Z. Liu, R. Gao, L. Hong, D.-Y. Yeung, H. Lu, and X. Jia. Trackdiffusion: Tracklet-conditioned video generation via diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3539–3548. IEEE, 2025. 21

[26] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 20

[27] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 17

[28] R. Liang, Z. Gojcic, H. Ling, J. Munkberg, J. Hasselgren, Z.-H. Lin, J. Gao, A. Keller, N. Vijaykumar, S. Fidler, and Z. Wang. Diffusionrenderer: Neural inverse and forward rendering with video diffusion models. *arXiv preprint arXiv: 2501.18590*, 2025. 21

[29] K. Lis, K. Nakka, P. Fua, and M. Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2152–2161, 2019. 2

[30] J. Y. Liu, Y. Chen, Z. Yang, J. Wang, S. Manivasagam, and R. Urtasun. Neural scene rasterization for large scene rendering in real time. In *The IEEE International Conference on Computer Vision (ICCV)*, 2023. 21

[31] Y. Lu, X. Ren, J. Yang, T. Shen, Z. Wu, J. Gao, Y. Wang, S. Chen, M. Chen, S. Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024. 21

[32] Y. Luo, C. Zheng, X. Yan, T. Kun, C. Zheng, S. Cui, and Z. Li. Latr: 3d lane detection from monocular images with transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7941–7952, 2023. 15

[33] E. Ma, L. Zhou, T. Tang, Z. Zhang, D. Han, J. Jiang, K. Zhan, P. Jia, X. Lang, H. Sun, et al. Unleashing generalization of end-to-end autonomous driving with controllable long video generation. *arXiv preprint arXiv:2406.01349*, 2024. 21

[34] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 21

[35] N. Moenne-Loccoz, A. Mirzaei, O. Perel, R. de Lutio, J. M. Esturo, G. State, S. Fidler, N. Sharp, and Z. Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024. 21

[36] D. C. Moura, S. Zhu, and O. Zvitia. Nexar dashcam collision prediction dataset and challenge, 2025. 8

[37] NVIDIA. Accelerate the future of ai-defined vehicles and autonomous driving - in-car computer: Alpamayo network architecture. https://www.nvidia.com/en-us/on-demand/session/gtc25-DD40000/, 2025. 19

[38] NVIDIA, H. Abu Alhaija, J. Alvarez, M. Bala, T. Cai, T. Cao, L. Cha, J. Chen, M. Chen, F. Ferroni, S. Fidler, D. Fox, Y. Ge, J. Gu, A. Hassani, M. Isaev, P. Jannaty, S. Lan, T. Lasser, H. Ling, M.-Y. Liu, X. Liu, Y. Lu, A. Luo, Q. Ma, H. Mao, F. Ramos, X. Ren, T. Shen, S. Tang, T.-C. Wang, J. Wu, J. Xu, S. Xu, K. Xie, Y. Ye, X. Yang, X. Zeng, and Y. Zeng. Cosmos-transfer1: Conditional world generation with adaptive multimodal control, 2025. 4, 5

[39] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4

[40] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1099–1106. IEEE, 2016. 2

[41] A. Pun, G. Sun, J. Wang, Y. Chen, Z. Yang, S. Manivasagam, W.-C. Ma, and R. Urtasun. Neural lighting simulation for urban scenes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 21

[42] X. Ren, J. Huang, X. Zeng, K. Museth, S. Fidler, and F. Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 21

[43] X. Ren, Y. Lu, H. Liang, J. Z. Zhang, H. Ling, M. Chen, S. Fidler, F. Williams, and J. Huang. Scube: Instant large-scale scene reconstruction using voxsplats. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2, 21

[44] X. Ren, T. Shen, J. Huang, H. Ling, Y. Lu, M. Nimier-David, T. Müller, A. Keller, S. Fidler, and J. Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[45] X. Ren, T. Yang, L. E. Li, A. Alahi, and Q. Chen. Safety-aware motion prediction with unseen vehicles for autonomous driving. In *ICCV*, 2021. 2

[46] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. *arXiv preprint arXiv:2005.03778*, 2020. 2, 20

[47] L. Russell, A. Hu, L. Bertoni, G. Fedoseev, J. Shotton, E. Arani, and G. Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025. 21

[48] J. Sarva, J. Wang, J. Tu, Y. Xiong, S. Manivasagam, and R. Urtasun. Adv3d: Generating safety-critical 3d objects through closed-loop simulation. *arXiv preprint arXiv:2311.01446*, 2023. 2

[49] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017. 2, 20

[50] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 14

[51] Q. Tian, X. Tan, Y. Xie, and L. Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 2, 21

[52] Y. Tian, Q. Ye, and D. Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025. 16

[53] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021. 2

[54] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024. 21

[55] Y. Wang, R. Gao, K. Chen, K. Zhou, Y. Cai, L. Hong, Z. Li, L. Jiang, D.-Y. Yeung, Q. Xu, et al. Detdiffusion: Synergizing generative and perceptive models for enhanced data generation and perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7246–7255, 2024. 21

[56] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. *arXiv preprint arXiv:2311.17918*, 2023. 21

[57] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8370–8380, June 2023. 2, 21

[58] Z. Wang, T. Shen, J. Gao, S. Huang, J. Munkberg, J. Hasselgren, Z. Gojcic, W. Chen, and S. Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 21

[59] Y. Wei, Z. Wang, Y. Lu, C. Xu, C. Liu, H. Zhao, S. Chen, and Y. Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15077–15087, 2024. 2, 21

[60] Y. Wen, Y. Zhao, Y. Liu, F. Jia, Y. Wang, C. Luo, C. Zhang, T. Wang, X. Sun, and X. Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6902–6912, 2024. 21

[61] H. Wu, X. Zuo, S. Leutenegger, O. Litany, K. Schindler, and S. Huang. Dynamic lidar re-simulation using compositional neural fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 21

[62] J. Z. Wu, Y. Zhang, H. Turki, X. Ren, J. Gao, M. Z. Shou, S. Fidler, Z. Gojcic, and H. Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. *arXiv preprint arXiv: 2503.01774*, 2025. 21

[63] Q. Wu, J. Martinez Esturo, A. Mirzaei, N. Moenne-Loccoz, and Z. Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 21

[64] W. Wu, Y. Zhao, H. Chen, Y. Gu, R. Zhao, Y. He, H. Zhou, M. Z. Shou, and C. Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems*, 36:54683–54695, 2023. 21

[65] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. 12

[66] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 21

[67] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 2, 21

[68] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4, 5

[69] G. Zhao, C. Ni, X. Wang, Z. Zhu, X. Zhang, Y. Wang, G. Huang, X. Chen, B. Wang, Y. Zhang, et al. Drivedreamer4d: World models are effective data machines for 4d driving scene representation. *arXiv preprint arXiv:2410.13571*, 2024. 21