CorrectAD: A Self-Correcting Agentic System to Improve End-to-end Planning in Autonomous Driving

Enhui Ma[1,2*†], Lijun Zhou[3*], Tao Tang[3†], Jiahuan Zhang[2‡], Junpeng Jiang[3†], Zhan Zhang[2‡], Dong Han[2‡], Kun Zhan[3], Xueyang Zhang[3], Xianpeng Lang[3], Haiyang Sun[3], Xia Zhou[3], Di Lin[4], Kaicheng Yu[2§]

[1]Zhejiang University
[2]Autolab, Westlake University
[3]Li Auto Inc.
[4]Tianjin University
{maenhui, kyu}@westlake.edu.cn

Abstract

End-to-end planning methods are the de-facto standard of the current autonomous driving system, while the robustness of the data-driven approaches suffers due to the notorious long-tail problem (i.e., rare but safety-critical failure cases). In this work, we explore whether recent diffusion-based video generation methods (a.k.a. world models), paired with structured 3D layouts, can enable a fully automated pipeline to self-correct such failure cases. We first introduce an agent to simulate the role of product manager, dubbed PM-Agent, which formulates data requirements to collect data similar to the failure cases. Then, we use a generative model that can simulate both data collection and annotation. However, existing generative models struggle to generate high-fidelity data conditioned on 3D layouts. To address this, we propose DriveSora, which can generate spatiotemporally consistent videos aligned with the 3D annotations requested by PM-Agent. We integrate these components into our self-correcting agentic system, CorrectAD. Importantly, our pipeline is end-to-end model-agnostic and can be applied to improve any end-to-end planner. Evaluated on both nuScenes and a more challenging in-house dataset across multiple end-to-end planners, CorrectAD corrects 62.5% and 49.8% of failure cases, reducing collision rates by 39% and 27%, respectively.

Introduction

End-to-end (E2E) autonomous driving has garnered increasing attention (Hu et al. 2023; Jiang et al. 2023; Yang et al. 2023b), which directly learns to plan motions from raw sensor inputs, thereby reducing heavy reliance on hand-crafted rules and avoiding cascading modules. Deploying robust E2E model is critical for real-world autonomy. However, long-tail scenarios encountered on the road can cause catastrophic failures due to limited representation in training data. To adapt to diverse and evolving driving environments, E2E models must be continuously refined. Yet, manually collecting high-quality data for such failure scenarios remains costly and risky, especially for dangerous situations. This problem leads to the emergence of an agentic system that helps E2E models self-correct, keeping them adaptable and effective.

To address this, we draw inspiration from the current data development paradigm of autonomous driving companies, which usually consists of the following steps: product managers receive failure case feedback from the deployment team, then they formulate data requirements and task the data team with collecting and annotating similar scenarios to augment the training set (see Fig. 1(a)). While effective, this manual process incurs drastically high costs in both data collection and annotation, often taking weeks and thousands of dollars per scenario. Alternative solutions (Liang et al. 2024) (see Fig. 1(b)) attempt to retrieve and auto-labeling similar data from the existing training dataset, but this severely limits scene diversity and cannot handle unseen failure cases.

In this paper, we propose a fully agentic system to simulate such process towards a self-correcting loop. As illustrated in Fig. 1(c), to substitute the data department's collection and annotation work, we use a generative model, dubbed as **DriveSora**, which can simulate the data collection and annotation process by generating multi-view videos controlled by precise 3D scene annotation. Unlike prior works that randomly generate scenes (Gao et al. 2023; Wen et al. 2023b; Yang et al. 2023a), our system focuses on generating targeted data tailored to failure correction. Yet, the generative model cannot directly take a failure case video to generate such data. To this end, we build an agent to simulate product manager, dubbed **PM-Agent**. This agent focuses on analyzing failure causes using VLM's reasoning abilities, and then formulates multimodal requirements (including bird's-eye-view layouts and scene descriptions) to interact with the generative model. Finally, by incorporating the generated data into the training dataset, our self-correcting agentic system, **CorrectAD**, significantly improves the robustness of downstream E2E models. Importantly, our approach is agnostic to E2E models and

[*]Co-first authors
[†]Work done during an internship at Li Auto Inc.
[‡]Work done during their visiting at Autolab, Westlake University.
[§]Corresponding author

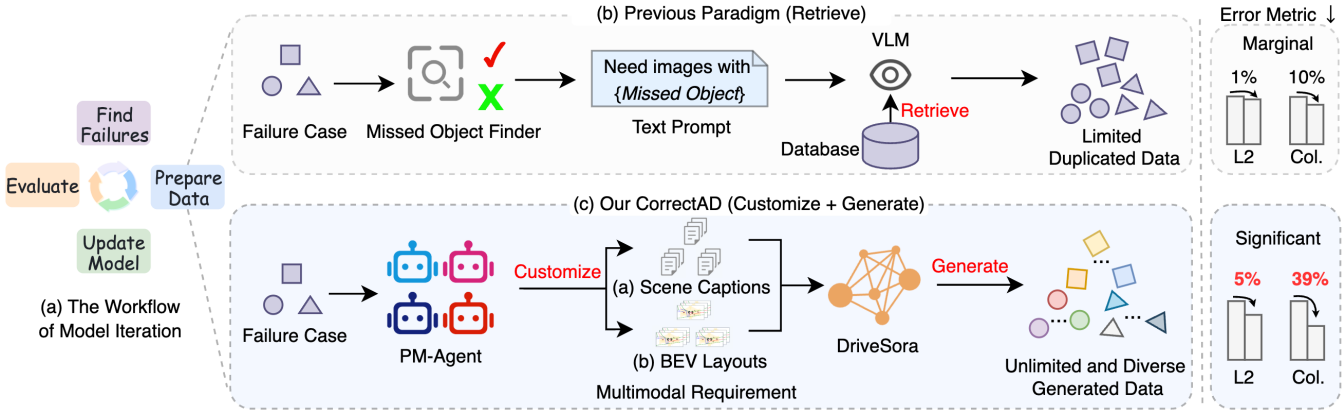

Figure 1: (a) A model iteration includes finding failures, preparing training data, updating the model, followed by evaluation and iteration again. The key challenge is how to prepare targeted training data to correct failures. (b) The previous retrieval-based paradigm relies on similar samples from existing datasets, restricting training diversity. (c) Our CorrectAD uses PM-Agent to analyze failures and formulate data requirements, and DriveSora to generate high-fidelity data aligned with the data requirements requested by PM-Agent, achieving lower L2 error and collision rate for end-to-end planning models.

can be applied across diverse planners. We demonstrate the effectiveness of CorrectAD on both nuScenes and a challenging in-house dataset, correcting 62.5% and 49.8% of failure cases respectively, and reducing collision rates by 39% and 27%. Our contributions can be summarized as follows:

- We introduce an agentic system to improve the E2E model by self-correcting failure cases.
- We propose PM-Agent that links failure cases and generative model, by analyzing failure causes and formulating multimodal requirements for data generation.
- We propose DriveSora, a controllable video generation model that surpasses prior works by 10.6% in FVD and 5.8% in NDS.
- We validate CorrectAD across datasets and planners, showcasing its E2E model-agnostic nature and substantial performance gains.

Related Work

Self-correction in Autonomous Driving. Self-correction involves a system detecting its errors and refining its decision-making ability to meet task requirements more effectively (Mitchell et al. 2018; Valmeekam, Marquez, and Kambhampati 2023). Vision language models (VLMs), with strong semantic and reasoning abilities, can assist in error validation and correction (Pan et al. 2023; Madaan et al. 2024). In autonomous driving, VLMs have improved decision reliability by providing external feedback to adjust autonomous driving outputs (Fu et al. 2024; Yang et al. 2023c; Cui et al. 2023; Wen et al. 2023a). However, this paradigm does not update the training data within the autonomous driving model, thus not to implement targeted optimizations based on failure cases. Recently, AIDE (Liang et al. 2024) mitigates novel object detection by retrieving and auto-labeling data from existing datasets. However, it is limited to detection models, and retrieval alone may lack data diversity. Contemporary works (Li et al. 2025) train specialized transformers to analyze driving accident causes but do not use these insights to improve E2E models. In contrast, our CorrectAD identifies failure causes from E2E reasoning results, including perception, prediction, and planning. This enables data generation tailored to these failure points, enhancing model diversity and effectiveness. In addition, through fully automated iterative cycles, CorrectAD can continuously optimize performance.

End-to-end Autonomous Driving. E2E models attract growing attention in autonomous driving by unifying perception, prediction, decision, and planning. ST-P3 (Hu et al. 2022) learns spatiotemporal features to enhance perception and planning. UniAD (Hu et al. 2023) jointly optimizes multiple perception and prediction tasks for improved planning. VAD (Jiang et al. 2023) adopts a vectorized scene representation to support map-free planning, and VADv2 (Chen et al. 2024) extends this with probabilistic planning on multi-view image sequences to predict control actions. We use UniAD, VAD, and our in-house E2E model to validate CorrectAD.

Multi-view Video Generation. Video generation is crucial for visual understanding. Progress in diffusion-based image synthesis (Nichol et al. 2021; Rombach et al. 2022; Ruiz et al. 2023) has enabled early video diffusion models (Harvey et al. 2022; Höppe 2022) with improved realism and controllability. BEV-conditioned generation is explored in BEVGen (Swerdlow, Xu, and Zhou 2023), BEVControl (Yang et al. 2023a), and MagicDrive (Gao et al. 2023) using ControlNet/ControlNet-like conditioning (Zhang and Agrawala 2023). Cross-frame modeling is extended for video in (Wen et al. 2023b; Wang et al. 2023b; Zhao et al. 2024). Layout-conditioned video generation for perception training appears in (Wen et al. 2023b; Wang et al. 2023b; Lu et al. 2025; Xie et al. 2025; **?**; Jia et al. 2023). World-model-style DiT video generation is advanced by Sora (**?**). Prior methods struggle with controllability and temporal consistency. We extend DiT to multi-view BEV geometry to provide higher spatiotemporal consistency for E2E training.

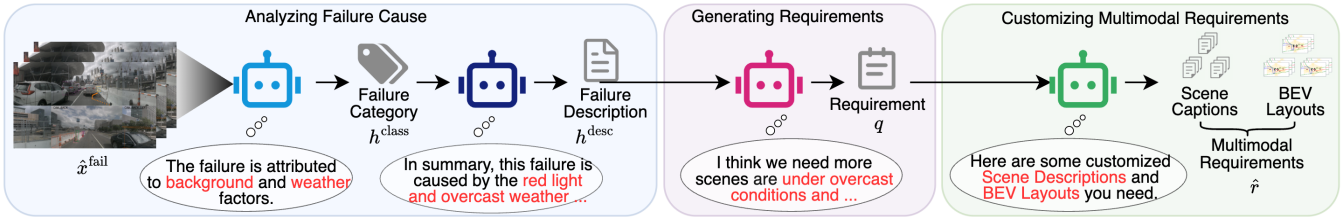

Figure 2: PM-Agent takes a failure case $\hat{x}^{\text{fail}}$, classifies failure causes into h^{class}, analyzes description h^{desc}, derives requirement query q, and formulates multimodal requirements $\hat{r}$ (BEV layout and scene caption) for the generative model interaction.

Method

Preliminary

Definition of Failure Cases. Given a dataset $D = \{D^{\text{train}}, D^{\text{val}}\}$, $D = (X, Y) = \{x_i, y_i\}_{i=1}^{|D|}$ consists of multi-view videos $x_i = \{x_i^j\}_{j=1}^{N_{\text{view}}}$ and corresponding 3D bboxes and map labels y_i. A failure case occurs when, following the planned trajectory for the next T_{e2e} timesteps from the E2E model $\mathcal{F}$, at least one collision occurs between the ego and others $V_{\text{other}} = \{v_j\}_{j=1}^{|V_{\text{other}}|}$ (including vehicles, pedestrian and barriers). Formally, the failure cases are defined as:

$$D^{\text{fail}} = \{(X, Y) \in D^{\text{train}} \mid \exists t \le T_{\text{e2e}}, \exists j \le |V_{\text{other}}|, \\ \|\mathbf{p}_{\text{ego}}(t) - \mathbf{p}_{\text{other}}^{j}(t)\| < \epsilon\}, \tag{1}$$

where $\mathbf{p}(t)$ is the vehicle's position at time t, $\|\cdot\|$ is the euclidean distance, and ϵ is the safety threshold.

Pre-identification of Failure Categories. To precisely analyze failures, we pre-identify the categories of failure causes in D. We use expert-annotated (details see Appendix) descriptions of failure causes $Y^{\text{desc}} = \{y_i^{\text{desc}}\}_{i=1}^{N_{\text{anno}}}$ from N_{anno} failure cases. We use LLM to extract keywords Y^{key} and apply an adaptive clustering algorithm to obtain K classes of causes $S = \{S_k\}_{k=1}^{K}$. The process is denoted as:

$$y_i^{\text{key}} = \mathcal{LLM}(y_i^{\text{desc}}) \tag{2}$$

$$S_k = \{y_i^{\text{key}} \in Y^{\text{key}} | \mathbf{d}(y_i^{\text{key}}, s_k) \le \mathbf{d}(y_i^{\text{key}}, s_j), \forall j \ne k\}, \tag{3}$$

where s_k is the center of the k-th cluster, and $\mathbf{d}(\cdot, \cdot)$ is the two points' distance. Then, we summarize the common cause features l_k contained in each cluster S_k for later CorrectAD, resulting in all possible failure categories $L = \{l_k\}_{k=1}^{K}$, where $l_k = \mathcal{LLM}(S_k)$.

CorrectAD Overview

The goal of CorrectAD is to generate new training data D^{gen} to specifically optimize failure cases D^{fail} of the E2E model $\mathcal{F}$, producing an updated $\mathcal{F}'$. At first, we preprocess the dataset: $D \leftarrow (X', C, E) = \{(x_i', c_i, e_i)\}_{i=1}^{|D|}$, where $x_i' = \text{concat}(x_i)$ represents the operation of concatenating the multi-view videos x_i in a cyclic order into a single large video x_i', $c_i = \mathcal{VLM}(x_i')$ represents the scene caption of the video x_i', and $e_i \leftarrow \text{project}(y_i)$ represents the BEV layout projected from BEV space into camera space. A similar definition applies to D^{train}, D^{val}, and D^{fail}.

To address the aforementioned challenge of generating new training data specifically for failure cases, we propose an automated data loop: First, the product manager, *i.e.*, **PM-Agent** $\mathcal{A}$, analyzes the failure and formulates multimodal requirements: $R \leftarrow \mathcal{A}(D^{\text{fail}})$. Next, the data department, *i.e.*, **DriveSora G**, generates the new training data: $D^{gen} \leftarrow \{(X^{gen}, R) \mid X^{gen} = \mathbf{G}(R)\}$. Then, $\mathcal{F}$ is updated by fine-tuning it on both old and new training data, followed by evaluation on D^{train} and iteration again.

PM-Agent

Since there is no effective way to link failure cases to the 3D generative model **G**, we propose the PM-Agent, as shown in Fig. 2, similar to a product manager, to bridge this gap by formulating 3D multimodal requirements R.

Analyzing Failure Cause. It is essential for precisely customizing requirements. The vanilla baseline uses one-step VLMs conversation. But this yields suboptimal accuracy due to VLMs' limitation in reasoning over complex tasks. We propose a multi-round inquiry strategy to decompose the task: first, classifying the cause, then analyzing the failure in detail. We first plot the output o^{fail} from $\mathcal{F}$ onto failure cases, resulting $\hat{x}^{\text{fail}} = \text{plot}(x'^{\text{fail}}, o^{\text{fail}})$, where o^{fail} includes detection, prediction and planning output for the next T_{e2e} timesteps. Next, we guide the VLMs to classify the failure cause, outputting the failure category h^{class}:

$$h^{\text{class}} = \mathcal{VLM}(\hat{x}^{\text{fail}}, L) = \{l_i \in L \mid \mathbf{q}(l_i \mid \hat{x}^{\text{fail}}) \ge \tau\}, \tag{4}$$

where $\mathbf{q}(\cdot \mid \cdot)$ is the probability that the later belongs to the former, τ is the classification threshold. Based on the classification result, we then perform a specificly analysis of the failure cause description h^{desc}:

$$h^{\text{desc}} = \mathcal{VLM}(\hat{x}^{\text{fail}}, h^{\text{class}}). \tag{5}$$

Generating Requirements. These requirements are essential for understanding the context and the details surrounding the failure, which will guide **G** to generate the desired data. For each failure case, we generate a requirement q based on both the class h^{class} and description h^{desc} of the failure cause:

$$q = \mathcal{LLM}(h^{\text{class}}, h^{\text{desc}}). \tag{6}$$

Formulating Multimodal Requirements. To better interface with **G**, we select the top-K samples from D^{train} whose scene captions c are most similar to q and extract the

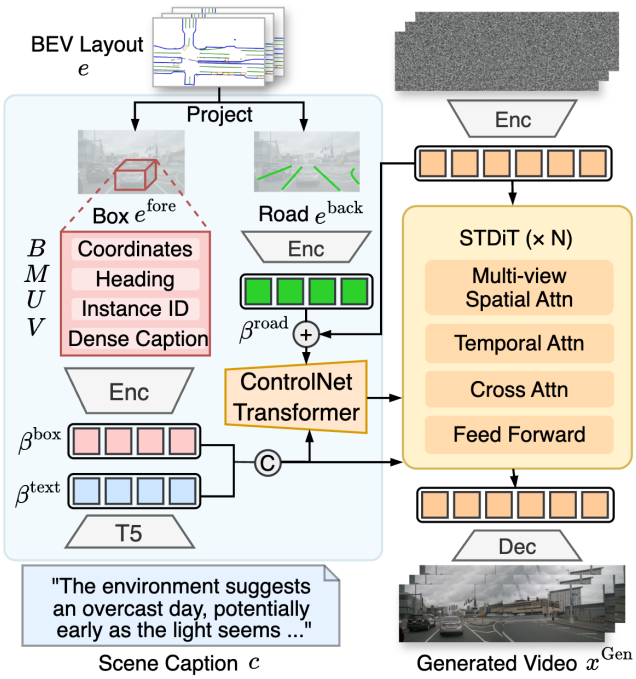

Figure 3: DriveSora generates high-quality, diverse data based on the data requirements specified by PM-Agent.

corresponding BEV layouts e to assemble the multimodal requirements $\hat{r}$:

$$\hat{r} = \mathcal{VLM}(q, D^{\text{train}}) = \{(c, e) \mid \mathbf{s}(c, q) \geq \delta\}, \quad (7)$$

where $\mathbf{s}(\cdot, \cdot)$ represents the similarity calculation, δ is the similarity threshold. Finally, the union of all $\hat{r}$, denoted as $R = \{\hat{r}_i\}_{i=1}^{|R|}$, serves as the set of multimodal requirements for the current iteration.

DriveSora

Since previous generative works struggle with the quality of generated data, we propose DriveSora **G**, akin to a data department, by specifically generating high-fidelity training data D^{gen} to enhance the ability of the E2E model $\mathcal{F}$ against complex scenario. As shown in Fig. 3, DriveSora takes the multimodal prompt R as input, based on the Spatial-Temporal Diffusion Transformer (STDiT) architecture to generate videos $X^{\text{gen}} = \{x_i^{\text{gen}}\}_{i=1}^{|X^{\text{gen}}|}$, where x_i^{gen} represents generated video which consists of T_{frame} frames and N_{view} views.

Multimodal Control Generation. We first improve generation fidelity by encoding more fine-grained conditions. The input multimodal prompt includes the scene caption c and the BEV layout e, where e is first decoupled into the foreground layout e^{fore} and the background layout e^{back}. $e^{\text{fore}} = (B, M, U, V) = \{(b_n, m_n, u_n, v_n)\}_{n=1}^{|N_{\text{view}}|}$, where $b_n \in [0,1]^{N^{\text{box}} \times 4}$ means bbox coordinates, $m_n \in [-180, 180)^{N^{\text{box}} \times 1}$ means heading, $u_n \in [0,1]^{N^{\text{box}} \times 1}$ means instance id, $v_n \in \mathbb{R}^{N^{\text{box}} \times 1}$ means dense caption, and N^{box} means the number of boxes. $e^{\text{back}} \in \mathbb{R}^{H \times W \times 3}$ means colored lines for road maps. To obtain the box embedding β^{box}, road embedding β^{road} and text embedding β^{text}, the encoding process is:

$$\begin{aligned} \beta^{\text{box}} &= \mathbf{Mlp}(\mathbf{Fe}(B) + \mathbf{Fe}(M) + \mathbf{Fe}(\mathrm{U}) + \mathbf{E_{text}}(\mathrm{V})), \\ \beta^{\text{road}} &= \mathbf{E_{image}}(\alpha), \qquad \beta^{\text{text}} = \mathbf{E_{text}}(c), \end{aligned} \quad (8)$$

where $\mathbf{Fe}(\cdot)$ is the Fourier Embedder (Mildenhall et al. 2021), $\mathbf{E_{text}}$ is the T5 Encoder (Raffel et al. 2020), and $\mathbf{E_{image}}$ is the VAE (Rombach et al. 2022). We concatenate box embedding β^{box} and text embedding β^{text} to enable text and vehicle control through cross-attention (**CA**) in STDiT:

$$\begin{aligned} q = \mathbf{Li}(z_{in}), \quad k = \mathbf{Li}([\beta^{\text{box}}, \beta^{\text{text}}]), \quad v = \mathbf{Li}([\beta^{\text{box}}, \beta^{\text{text}}]), \\ \mathbf{CA}(q, k, v) = \mathbf{Softmax}(\frac{q \cdot k^T}{\sqrt{d}}) \cdot v, \end{aligned} \quad (9)$$

where $\mathbf{Li}(\cdot)$ is a linear layer, and $z_{in} \sim \mathcal{N}(0, 1)$ is the noise latents. Following ControlNet (Zhang and Agrawala 2023), we add a trainable ControlNet-Transformer to STDiT for precise layout control with road embedding β^{road}. The STDiT block's calculation process is formulated as:

$$z_{out} = \mathbf{STDiT}(z_{in}) + \mathbf{Zero}(\mathbf{Control}(z_{in} + \beta^{\text{road}})), \quad (10)$$

where $\mathbf{Zero}(\cdot)$ is zero-initialized trainable convolution layers, and $\mathbf{Control}(\cdot)$ is the ControlNet-Transformer, which is detailed in Appendix.

Parameter-free Multi-view Spatial Attention. To enhance spatial consistency, we extend STDiT's Self-Attention with Multi-View Self-Attention (MVA). Unlike prior works using additional cross-view attention (Gao et al. 2023; Wen et al. 2023b), our parameter-free approach reshapes $z_{in} \in \mathbb{R}^{(BV) \times (TS) \times C}$ to $z'_{in} \in \mathbb{R}^{(BT) \times (VS) \times C}$ (S is embedding resolution) and applies self-attention directly:

$$\begin{aligned} z'_{in} &= \mathbf{Reshape}(z_{in}), \\ q = \mathbf{Li}(z'_{in}), \quad k = \mathbf{Li}(z'_{in}), \quad v = \mathbf{Li}(z'_{in}), \\ \mathbf{MVA}(q, k, v) &= \mathbf{Softmax}(\frac{q \cdot k^T}{\sqrt{d}}) \cdot v. \end{aligned} \quad (11)$$

Multi-conditional Classifier-free Guidance. We improve the condition-content alignment by conditional and unconditional denoising mode. Unlike (Gao et al. 2023), which concurrently sets all conditions to null ϕ in the unconditional mode, we alternately nullify each condition to strengthen individual guidance. The generator $\mathbf{G}_\theta(z_{\text{in}}, e^{\text{fore}}, e^{\text{back}}, c)$ takes box, road, and text conditions with guidance scales $\lambda_{\text{fore}}, \lambda_{\text{back}}, \lambda_{\text{text}}$. During training, we set each condition to ϕ independently with a 5% probability, and all jointly with the same rate. During inference, the process is formulated as:

$$\begin{aligned} \tilde{\mathbf{G}}_\theta(z_{\text{in}}, e^{\text{fore}}, c_R, c) &= \mathbf{G}_\theta(z_{\text{in}}, \phi, \phi, \phi) \\ &+ \lambda_{\text{text}} \cdot (\mathbf{G}_\theta(z_{\text{in}}, \phi, \phi, c) - \mathbf{G}_\theta(z_{\text{in}}, \phi, \phi, \phi)) \\ &+ \lambda_{\text{back}} \cdot (\mathbf{G}_\theta(z_{\text{in}}, \phi, e^{\text{back}}, c) - \mathbf{G}_\theta(z_{\text{in}}, \phi, \phi, c)) \\ &+ \lambda_{\text{fore}} \cdot (\mathbf{G}_\theta(z_{\text{in}}, e^{\text{fore}}, e^{\text{back}}, c) - \mathbf{G}_\theta(z_{\text{in}}, \phi, e^{\text{back}}, c)). \end{aligned} \quad (12)$$

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
UniAD metrics								
NMP	-	-	2.31	-	-	-	1.92	-
SA-NMP	-	-	2.05	-	-	-	1.59	-
FF	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
UniAD	**0.48**	0.96	1.65	1.03	0.05	0.17	0.71	0.31
AIDE*	0.51	0.96	1.60	1.02	0.05	0.16	0.64	0.28
CorrectAD*	0.50	**0.92**	**1.53**	**0.98**	**0.02**	**0.14**	**0.42**	**0.19**
ST-P3 Metrics								
ST-P3	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
VAD	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22
AIDE†	0.39	0.68	1.01	0.69	0.06	0.17	0.42	0.22
CorrectAD†	**0.34**	**0.60**	**0.94**	**0.62**	**0.05**	**0.14**	**0.40**	**0.20**

Table 1: E2E planning comparison on nuScenes validation set. * and † denotes frameworks initialized by UniAD and VAD, respectively.

Method	L2 (m) ↓				Hit Rate (%) ↑			
	1s	3s	8s	Avg.	1s	3s	8s	Avg.
Baseline	0.10	0.54	1.91	0.85	0.98	0.80	0.53	0.77
AIDE‡	0.09	0.50	1.79	0.79	0.98	0.81	0.54	0.78
CorrectAD‡	**0.08**	**0.44**	**1.33**	**0.62**	**0.99**	**0.83**	**0.63**	**0.82**

Table 2: E2E planning comparison on a large in-house validation set. Hit Rate measures recall of predicted trajectories against ground truth at multiple timesteps. ‡ denotes framework initialized by Baseline (our in-house E2E model).

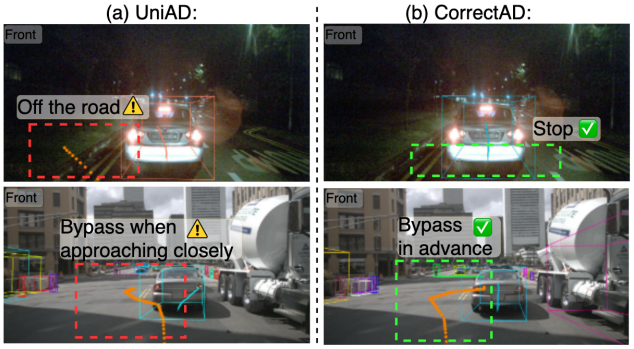

Figure 4: Two nuScenes validation examples before and after self-correction. Our framework can fix low-visibility night driving (top) and bypassing in dense traffic (bottom).

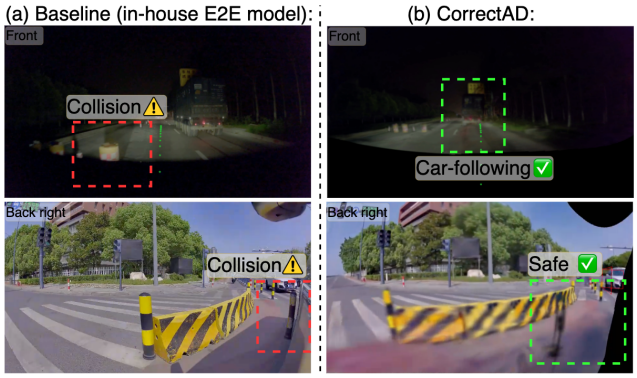

Figure 5: Two cases before and after self-correction on our in-house validation set, rendered in a proprietary closed-loop simulator based on Gaussian splatting.

Experiments

Experimental Setting

Dataset. We evaluate on two datasets: (1) the real-world nuScenes (Caesar et al. 2020) dataset with 700 training and 150 validation scenes of 20s 6-view videos at 12Hz; (2) a more challenging in-house E2E dataset with diverse driving behaviors, containing 3M training and 0.6M validation scenes of 15s 6-view videos at 10Hz. Behavior distribution is detailed in the Appendix.

Metrics. We evaluate CorrectAD in three E2E models: UniAD (Hu et al. 2023), VAD (Jiang et al. 2023) (using L2 error and collision rate), and our in-house E2E model (using L2 error and hit rate). For PM-Agent, we assess its analysis ability using the accuracy of the failure category and the semantic distance of the descriptions. For DriveSora, we assess the fidelity and consistency of the generated videos (using FID (Heusel et al. 2017), FVD (Unterthiner et al. 2018), and CLIP score (Yang et al. 2023a)), and detection score (using NDS (Wang et al. 2023a)) to measure the sim-to-real gap.

Methods for Comparison. To our knowledge, little work focuses on automated data pipeline for self-correcting failures in autonomous driving E2E models, making it difficult to find a fully comparable counterpart for CorrectAD. However, we noticed AIDE (Liang et al. 2024), a closed-source method for novel object detection tasks, which shares a similar process: identifying issues, curating data, updating the model, and verifying results. Key differences include: **1)** AIDE targets detection tasks, while our method focuses on E2E planning tasks; **2)** AIDE retrieves data from existing dataset, while we generate new data using a generative model. To ensure a fair comparison, we re-implemented AIDE's process for the planning task in this paper. Details are in the Appendix.

Main Results

Evaluating CorrectAD against state-of-the-art methods on the nuScenes validation set, our framework achieves superior performance in both L2 and collision rate metrics (see Tab. 1). In contrast to AIDE, which only retrieves training data, CorrectAD improves safety metrics by analyzing failure causes and specifically generating new training data. We also show how our CorrectAD achieves self-correction in Fig. 4. Only the front view is shown here for clarity. All multi-view results are in the Appendix.

Furthermore, evaluating on the large in-house E2E model (see Tab. 2), CorrectAD significantly outperforms AIDE in L2 error and hit rate, demonstrating strong generalization capability across different E2E models. Fig. 5 shows the

(1)	(2)	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
✗	✗	0.54	1.03	1.71	1.09	0.05	0.18	0.81	0.35
✗	✓	0.53	0.99	1.66	1.06	0.10	0.20	0.62	0.31
✓	✗	0.52	0.96	1.62	1.03	0.08	0.20	0.58	0.29
✓	✓	**0.50**	**0.92**	**1.53**	**0.98**	**0.02**	**0.14**	**0.42**	**0.19**

Table 3: Ablation on (1) PM-Agent and (2) DriveSora.

Method	Foreground acc.↑	Background acc.↑	Weather acc.↑	Semantic dist.↓
Baseline(1 step)	N/A	N/A	N/A	4.72
PM-Agent	**92.59%**	**87.41%**	**91.85%**	**3.49**

Table 4: Accuracy of VLM in analyzing failure causes.

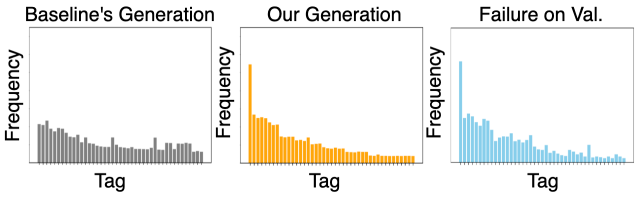

Figure 6: Distribution gap between generated data from AIDE baseline, our method, and failures on the validation set.

self-correction results on a large in-house dataset, which is visualized via our proprietary closed-loop simulator based on Gaussian Splatting (Yan et al. 2024), demonstrating effectiveness in fixing failures.

Statistical Distribution of Augmented Data. To better understand why our method significantly outperforms the AIDE baseline in enhancing the performance of the E2E model, we visualize the statistical distribution of the augmented data each method provides (see Fig. 6). A detailed explanation of our visualization approach is available in the Appendix. The rightmost column in the figure highlights the distribution of failure cases in the validation set, arguably the most critical distribution for the E2E planning model to learn from. Notably, the data generated by our method exhibit a much closer alignment with this failure distribution compared to other methods. This strong alignment is a key factor that enables our approach to deliver superior effectiveness.

Ablation Studies

Ablation on Proposed PM-Agent and DriveSora. To assess the individual contributions of the two proposed modules, we disable each in turn. In the first row of Tab. 3, we use augmented data created by randomly duplicating samples from the training set. This yields no gain due to redundant data without meaningful distributional alignment. Introducing DriveSora in the second row generates more diverse data, which partially mitigates this issue and leads to moderate improvements. As shown in the last two rows, incorporating PM-Agent to tailor the augmented data distribution to failure

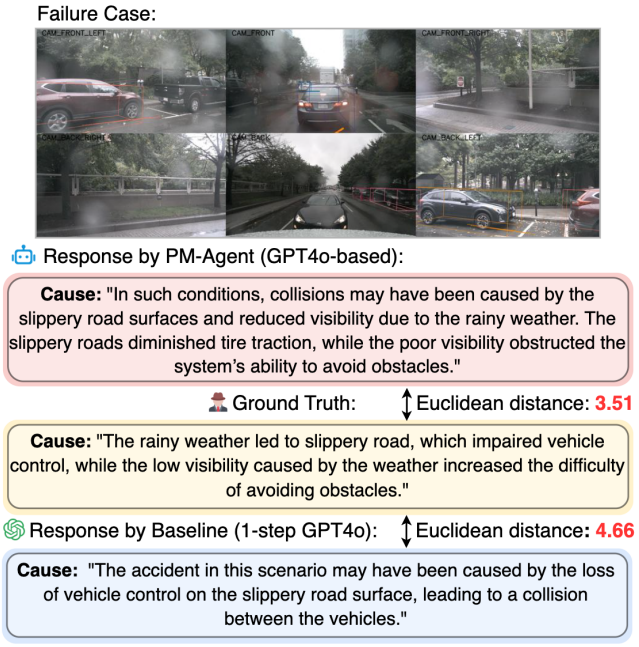

Figure 7: An cause example of GT and response by PM-Agent and baseline (one-step GPT4o).

cases yields further gains. Combining both DriveSora and PM-Agent, our full method achieves the best results: 0.98 L2 error and 0.19 collision rate, demonstrating the impact of using DriveSora for data diversity and PM-Agent for failure-focused distribution control. This validates the importance of both the distribution and diversity of the augmented data.

The Accuracy of PM-Agent. Tab. 4 compares PM-Agent's results with those obtained from a single direct prompt (one-step) to the VLM, where N/A means not available due to baseline skipping analysis failure category. Specifically, we used the expert-annotated data, as the ground truth (GT).

Subsequently, we measured the degree of alignment between the different outputs and the GT by calculating the textual semantic distance. The VLM we chose is GPT-4o, and the results show that our PM-Agent is effective. We can find that, by decomposing complex tasks into a series of subtasks for multi-step reasoning, PM-Agent significantly improved accuracy, reducing the semantic distance from 4.72 to 3.49. As a reference, we provide visual cases scoring both 3.51 and 4.66 in Fig. 7. We emphasize that using VLM to analyze causes is an exploratory area in the field. Real-world failures are more complex, and we expect that the proposed paradigm can offer insights to the industry.

Comparison of the Data Quality Generated by DriveSora. We assess the quality of video generation through a comprehensive evaluation including both quantitative and qualitative aspects, comparing our proposed DriveSora with previous generative methods. As shown in Tab. 5, we report metrics for three aspects: spatial and temporal consistency, and sim2real gap, on the nuScenes validation set. In short, our method surpasses state-of-the-art by a clear margin in video generation tasks. In Fig. 8, we

Generator	FID↓	CLIP↑	FVD↓	NDS↑
BEVGen	25.54	71.23	-	N/A
BEVControl	24.85	82.70	-	N/A
DriveDreamer	26.8	N/A	353.2	N/A
DriveDreamer-2	25.0	N/A	105.1	N/A
WoVoGen	27.6	N/A	417.7	N/A
MagicDrive	16.20	82.47	221.90	34.56
Panacea	16.96	84.23	139.0	32.10
Drive-WM	15.80	N/A	122.7	N/A
MagicDrive-v2	20.91	85.25	94.84	35.79
DriveSora (Ours)	**15.08**	**86.73**	**94.51**	**36.58**

Table 5: Comparison of DriveSora with state-of-the-art generators in terms of consistency and controllability on the nuScenes validation set. N/A means not available due to closed-source.

Generator	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
Panacea	**0.49**	0.98	1.62	1.03	0.08	0.18	0.56	0.27
MagicDrive-v2	0.50	0.96	1.55	1.00	0.05	**0.13**	0.51	0.23
DriveSora	0.50	**0.92**	**1.53**	**0.98**	**0.02**	0.14	**0.42**	**0.19**

Table 6: The effect of using different video generators in CorrectAD.

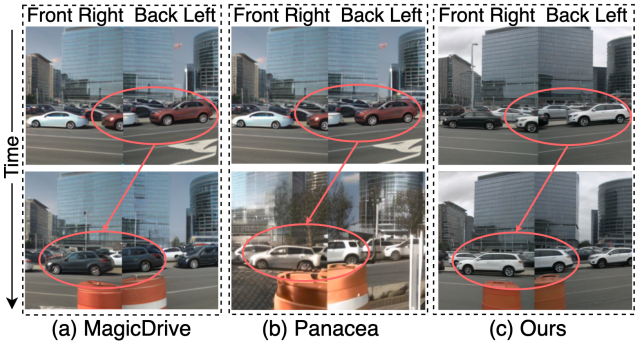

Figure 8: The visualization comparison of cross-frame consistency.

present videos generated by different methods on the same clip. Our method maintains a consistent spatial and temporal appearance, whereas the previous methods failed. It can be seen that our method has the powerful ability to generate high-quality videos with spatiotemporal consistency, which is beneficial for the training of E2E models.

Effects using Different Video Generators in CorrectAD. To further validate the impact of generated data quality on the performance of the E2E model, we replace the generative model within CorrectAD with previous methods (Wen et al. 2023b; Gao et al. 2025). The model trained with data generated by previous methods performs worse than the model trained with data from DriveSora (see Tab. 6), which highlights the importance of high-quality generated data for training E2E models.

Effectiveness of Multiple Iterations. Our CorrectAD is designed as an iterative self-correcting system for E2E models. Within the time constraints, we conducted several cycles of iteration. As shown in Tab. 7, both the L2 error and collision rate decreased progressively with more iterations. Fig. 9 illustrates the distribution differences between the generated data and the failures in the validation set for each iteration. The visualization demonstrates that, with more iterations, the distribution of the data generated by our method increasingly aligns with the distribution of failures, which explains why our method gradually reduces both the L2 error and collision rate. This highlights the self-correcting potential of our CorrectAD framework.

Iter.	D-D ↓	L2 (m) ↓				Collision (%) ↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
1	0.15	**0.50**	0.99	1.68	1.06	0.07	0.19	0.53	0.26
2	0.11	0.51	0.96	1.65	1.04	0.04	0.17	0.46	0.22
3	**0.09**	**0.50**	**0.92**	**1.53**	**0.98**	**0.02**	**0.14**	**0.42**	**0.19**

Table 7: Effect of multiple CorrectAD iterations. Iter. means iteration count. D-D metric shows the Hellinger Distance distribution between generated data and validation failures.

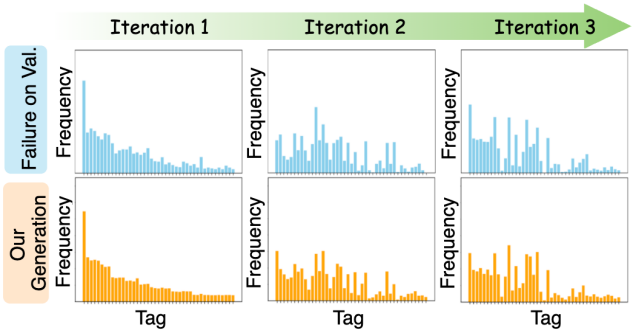

Figure 9: Distribution gap between augmented data and failures on the validation set over multi-iterations.

Conclusion

In this paper, we propose a self-correcting agentic system, CorrectAD, to effectively improve the E2E models in autonomous driving. We first propose a PM-Agent to analyze failure causes and formulate data requirements. Then, we introduce DriveSora to generate high-fidelity training data, thereby correcting the failures of E2E models. Experiments on multiple datasets proves that CorrectAD shows significant improvements in L2 error and collision rate, showcasing its excellent robustness, and providing a sustainable model self-correction solution for autonomous driving.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) under grant No. 62403389, the Provincial Natural Science Foundation of Zhejiang under grant No. QKWL25F0301, and the Zhejiang Key Laboratory of Low-Carbon Intelligent Synthetic Biology (2024ZY01025).

References

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, S.; Jiang, B.; Gao, H.; Liao, B.; Xu, Q.; Zhang, Q.; Huang, C.; Liu, W.; and Wang, X. 2024. VADv2: End-to-End Vectorized Autonomous Driving via Probabilistic Planning. *arXiv preprint arXiv:2402.13243*.

Cui, Y.; Huang, S.; Zhong, J.; Liu, Z.; Wang, Y.; Sun, C.; Li, B.; Wang, X.; and Khajepour, A. 2023. Drivellm: Charting the path toward full autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*.

Fu, D.; Li, X.; Wen, L.; Dou, M.; Cai, P.; Shi, B.; and Qiao, Y. 2024. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 910–919.

Gao, R.; Chen, K.; Xiao, B.; Hong, L.; Li, Z.; and Xu, Q. 2025. MagicDrive-V2: High-Resolution Long Video Generation for Autonomous Driving with Adaptive Control. *arXiv preprint arXiv:2411.13807*.

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

Harvey, W.; Naderiparizi, S.; Masrani, V.; Weilbach, C.; and Wood, F. 2022. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35: 27953–27965.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Höppe, T. 2022. Diffusion Models for Video Prediction and Infilling: Training a conditional video diffusion model for arbitrary video completion tasks.

Hu, S.; Chen, L.; Wu, P.; Li, H.; Yan, J.; and Tao, D. 2022. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision (ECCV)*.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17853–17862.

Jia, F.; Mao, W.; Liu, Y.; Zhao, Y.; Wen, Y.; Zhang, C.; Zhang, X.; and Wang, T. 2023. Adriver-i: A general world model for autonomous driving. *arXiv preprint arXiv:2311.13549*.

Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; and Wang, X. 2023. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 8306–8316.

Li, C.; Zhou, K.; Liu, T.; Wang, Y.; Zhuang, M.; Gao, H.-a.; Jin, B.; and Zhao, H. 2025. AVD2: Accident Video Diffusion for Accident Video Description. *arXiv preprint arXiv:2502.14801*.

Liang, M.; Su, J.-C.; Schulter, S.; Garg, S.; Zhao, S.; Wu, Y.; and Chandraker, M. 2024. AIDE: An Automatic Data Engine for Object Detection in Autonomous Driving. *arXiv preprint arXiv:2403.17373*.

Lu, J.; Huang, Z.; Yang, Z.; Zhang, J.; and Zhang, L. 2025. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. In *European Conference on Computer Vision*, 329–345. Springer.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5): 103–115.

Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.

Pan, L.; Saxon, M.; Xu, W.; Nathani, D.; Wang, X.; and Wang, W. Y. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22500–22510.

Swerdlow, A.; Xu, R.; and Zhou, B. 2023. Street-View Image Generation from a Bird's-Eye View Layout. *arXiv preprint arXiv:2301.04634*.

Unterthiner, T.; Van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

Valmeekam, K.; Marquez, M.; and Kambhampati, S. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023a. Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection. *arXiv preprint arXiv:2303.11926*.

Wang, X.; Zhu, Z.; Huang, G.; Chen, X.; and Lu, J. 2023b. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*.

Wen, L.; Fu, D.; Li, X.; Cai, X.; Ma, T.; Cai, P.; Dou, M.; Shi, B.; He, L.; and Qiao, Y. 2023a. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*.

Wen, Y.; Zhao, Y.; Liu, Y.; Jia, F.; Wang, Y.; Luo, C.; Zhang, C.; Wang, T.; Sun, X.; and Zhang, X. 2023b. Panacea: Panoramic and Controllable Video Generation for Autonomous Driving. *arXiv preprint arXiv:2311.16813*.

Xie, B.; Liu, Y.; Wang, T.; Cao, J.; and Zhang, X. 2025. Glad: A streaming scene generator for autonomous driving. *arXiv preprint arXiv:2503.00045*.

Yan, Y.; Lin, H.; Zhou, C.; Wang, W.; Sun, H.; Zhan, K.; Lang, X.; Zhou, X.; and Peng, S. 2024. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, 156–173. Springer.

Yang, K.; Ma, E.; Peng, J.; Guo, Q.; Lin, D.; and Yu, K. 2023a. BEVControl: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*.

Yang, Z.; Chen, L.; Sun, Y.; and Li, H. 2023b. Visual Point Cloud Forecasting enables Scalable Autonomous Driving. *arXiv preprint arXiv:2312.17655*.

Yang, Z.; Jia, X.; Li, H.; and Yan, J. 2023c. LLM4Drive: A survey of large language models for autonomous driving. In *NeurIPS 2024 Workshop on Open-World Agents*.

Zhang, L.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*.

Zhao, G.; Wang, X.; Zhu, Z.; Chen, X.; Huang, G.; Bao, X.; and Wang, X. 2024. DriveDreamer-2: LLM-Enhanced World Models for Diverse Driving Video Generation. *arXiv preprint arXiv:2403.06845*.