ScenarioControl: Vision-Language Controllable Vectorized Latent Scenario Generation

Lili Gao[1,*], Yanbo Xu[2,*], William Koch[2,*], Samuele Ruffino[1], Luke Rowe[3], Behdad Chalaki[1], Dmitriy Rivkin[1], Julian Ost[1,2], Roger Girgis[1,3], Mario Bijelic[1,2] and Felix Heide[1,2]

[1]Torc Robotics, [2]Princeton University, [3]Mila

Abstract

We introduce ScenarioControl, the first vision-language control mechanism for learned driving scenario generation. Given a text prompt or an input image, ScenarioControl synthesizes diverse, realistic 3D scenario rollouts – including map, 3D boxes of reactive actors over time, pedestrians, driving infrastructure, and ego camera observations. The method generates scenes in a vectorized latent space that represents road structure and dynamic agents jointly. To connect multimodal control with sparse vectorized scene elements, we propose a cross-global control mechanism that integrates cross-attention with a lightweight global-context branch, enabling fine-grained control over road layout and traffic conditions while preserving realism. The method produces temporally consistent scenario rollouts from the perspectives different actors in the scene, supporting long-horizon continuation of driving scenarios. To facilitate training and evaluation, we release a dataset with text annotations aligned to vectorized map structures. Extensive experiments validate that the control adherence and fidelity of ScenarioControl compare favorable to all tested methods across all experiments. Project webpage: https://light.princeton.edu/ScenarioControl

1. Introduction

Large-scale datasets have been indispensable for the progress of autonomous driving [24], providing multi-modal labeled data across regions and conditions [4, 15, 25, 46, 47]. However, real-world logs alone are insufficient to capture rare but safety-critical events, such as wrong-way drivers. Evaluating these edge cases is essential for safe and reliable systems, yet relying solely on collected data is highly sample-inefficient [44].

Simulators bridge this gap by enabling safe, repeatable, and scalable experimentation. Traditional rule-based simulators such as CARLA [11], SMARTS [64], and MetaDrive [27] provide controllable virtual worlds for perception and planning research, yet their handcrafted world design limits realism and diversity, particularly for rare events [7, 29]. Recent generative approaches, such as Driving Diffusion [28] and Panacea [56], introduce controllability by conditioning on structured "control layouts" derived from existing scenarios which steer generation by projecting logged scenes into intermediate control representations [13, 23, 28, 31, 39, 43, 55, 56]. Behavior- and interaction-level methods control agent dynamics to create challenging closed-loop interactions, but commonly assume a given road graph and initial scene context [12, 17, 41, 66]. Even for a given initial scenario description, these methods often struggle to generate long-tail events as the datasets used to train them contain few of such examples. To the best of our knowledge, no existing method allows for vision-language-controlled scenario generation.

In parallel, diffusion-based and data-driven simulation work has made substantial progress on generating realistic and diverse driving scenes from data. These methods treat scenario synthesis as a generative modeling problem over vectorized [42] or rasterized representations [7, 9, 32, 38], either placing actors on an existing road graph [32, 38] or generating both the road graph and actor positions jointly [9, 42]. Generative models have been shown to produce high-fidelity structured outputs – road topology and traffic participants that are directly consumable by downstream components such as motion planning, sensor simulation, end-to-end driving, and multimodal future generation [16, 17, 30, 52, 57, 66]. However, a crucial gap remains: while realism and diversity are consistently improving, the controllability at the scenario-level of the layouts themselves is still limited. Generation is either entirely uncontrolled - by sampling from the latent space - or with control signals

*Indicates equal contribution.

taken from existing scenarios, and extrapolation does not expose interpretable control knobs [9, 42]. We propose ScenarioControl to bridge this gap. ScenarioControl enables vision-language control of diffusion-based generation, unlocking controllable synthesis of structured driving scenarios and camera sensor simulation for the ego, and any other actor in the scene. Unlike other methods, it does so without relying on given logged scene configurations or auto-labeled control layouts. The resulting scenarios are plug-and-play with established simulators and autonomy stacks.

To this end, we introduce a novel cross-global control mechanism that conditions sparse vectorized scene tokens on dense features, from either a text prompt or a dashcam ego image, via two complementary branches: a cross-attention branch for fine-grained control and a lightweight global-context branch for capturing high-level scene intent. Conditioning on natural language and visual cues enables goal-directed scenario synthesis that both reflects real-world context and targets specific long-tail regimes. By steering road geometry, agent placement, and traffic conditions with a text prompt or a single ego dashcam-style image, generation moves beyond unconstrained sampling while preserving realistic structure and dynamics, crucial for synthesizing safety-critical cases and building targeted evaluation/training sets. In addition to enabling prompt- and image-conditioned scene generation, ScenarioControl also supports scene outpainting and long-horizon video continuation, maintaining temporal and visual consistency. We confirm ScenarioControl's fidelity and controllability with quantitative experiments, while qualitative results demonstrate adherence to conditioning and diverse long-horizon rollouts.

Our contributions are summarized as follows:

- We propose a vision–language conditioned vectorized latent diffusion model that generates full vectorized driving scenes, including lane topology, agent placement, and traffic signals conditioned on text prompts or dash-cam-style ego images.
- We introduce a new conditioning mechanism that fuses sparse, vectorized road layouts with dense prompt and image representations, enabling fine-grained control over generation.
- We evaluate controllability, diversity, and fidelity of the generated scenarios for arbitrary actors in the scene, confirming that the proposed method performs favorably compared to existing methods while providing fine-grained vision-language control.

2. Related Work

The field of synthetic data generation for autonomous driving tasks can be clustered into *traffic simulation*, *scenario generation* methods and *sensor data generation* methods. The former focuses on behavioral simulation of traffic participants, whereas the latter encompasses the generation of maps, agent start locations, and static road obstacles. The last one builds on established behavior rollouts and scenario layouts to generate corresponding multi-modal sensor data.

Traffic Simulation. Traditional autonomous driving simulators such as CARLA[11] and others [5, 40] provide reproducible testbeds but rely on hand-crafted rules and scripted behaviors, limiting their ability to capture real-world driving diversity. Procedural generation approaches, including MetaDrive and SMARTS [27, 64], improve scalability but struggle with behavioral realism and rare event coverage. Data-driven simulators address these limitations by learning directly from real-world logs. Systems such as Waymax and GPUDrive [17, 26] enable hardware-accelerated training on logged scenarios, improving behavioral fidelity. However, replay-based approaches remain fundamentally constrained to observed patterns and cannot synthesize novel situations or explore counterfactual futures.

This limitation motivates generative simulators that synthesize new diverse and realistic scenarios, with several works addressing the behavior gap of log replay [12, 41]. TrafficGen [12] generates both initial agent states and agent behavior, CtRL-Sim [41] specializes in the latter. They use a transformer-based driving policy to enable controllable and adversarial agent behaviors but assume pre-existing scene layouts.

Scenario Generation. Most traffic simulators require an initialized scene from which to roll out agent behaviors. A significant body of work focuses on producing suitable initial representations for agents [32, 34], lane graphs [35] or both [9, 42, 48], providing explicit structural control. When combined with traffic simulators, such initial scenes enable complete scenario generation with agent behaviors [9, 42]. SLEDGE [9] uses a raster-to-vector autoencoder with diffusion models to generate lane graphs and initial agent placements, but relies on rule-based traffic models for agent behaviors, limiting behavioral realism, while [42] builds on a vectorized scene representation to both accelerate the generation and increase quality. However, these methods do not allow for visual/textual grounding and focus on vectorized layouts for planners under flat-ground assumptions, rather than generating realistic sensor data.

Sensor Data Generation. A line of work focuses on reconstructing photorealistic sensor observations. Neural rendering methods based on NeRF and 3D Gaussian Splatting, including UniSim, NeuRAD, and Street Gaussians [49, 59, 61], achieve high visual fidelity but require explicit scene reconstruction and remain tethered to captured layouts, limiting their ability to generate counterfactual scenarios. Diffusion-based world models offer a more flexible alternative by directly synthesizing sensor videos. GAIA-1 [23] demonstrates controllable multi-camera gen-

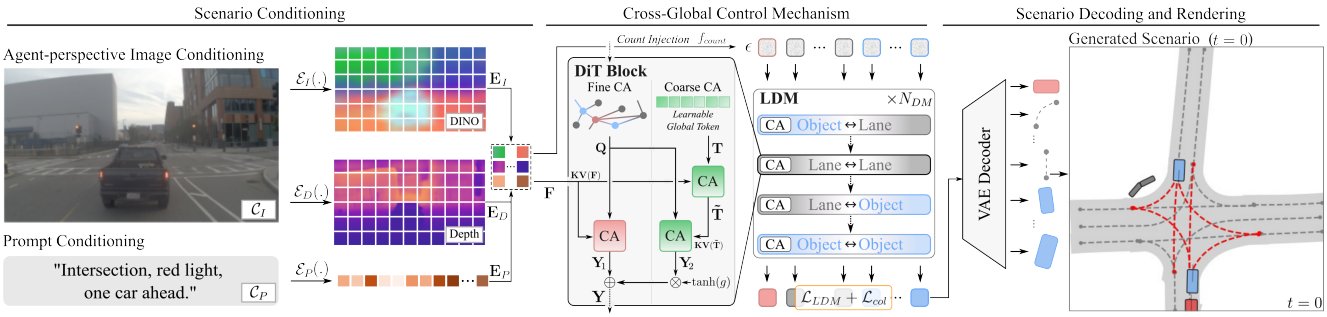


Figure 1. **ScenarioControl Conditioning and Initial Scene Generation.** Our method generates controllable, vectorized driving scenarios from visual (option 1) or prompt (option 2) conditioning. Text prompt or image embeddings guide the latent diffusion model (LDM) via cross-global control mechanism to ensure structurally valid scene generation (right). Red, blue, and grey tinting in both the latent and decoded spaces indicate ego vehicle, other vehicles, and lanes, respectively.

eration, while subsequent work scales to longer horizons (LongDWM) and improves multi-view consistency (GAIA-2) [43, 54]. Cosmos-Drive-Dreams [39] extends these capabilities by post-training the Cosmos world foundation model on large-scale driving data, enabling precise HDMap and 3D bounding box control, multi-view expansion from single views, and LiDAR point cloud generation alongside RGB synthesis. DriveArena [60] explores closed-loop evaluation with diffusion models. Methods such as SimGen, GeoDrive and UMGen [6, 57, 65] improve controllability by conditioning image generation on structural layouts and 3D geometry though these methods typically focus on rasterized map or non-structured scene representations.

Methods that enable trajectory-level control, such as Epona [62] and ProphetDWM [53], or safety-critical scenario synthesis like AdvDiffuser [8], still operate either purely in abstract action control modes [14] or purely in pixel space.

ScenarioControl bridges this gap and exposes explicit control handles that allow for a controllable scenario generation with descriptive, interpretable scene prompts, subsequent behavioral roll-outs, and final sensor data generation. Unlike existing vectorized methods that generate scenes unconditionally, we enable the conditional generation of initial scenes. This produces complete vectorized scene graphs with explicit topology and agent placement that reflect real-world observations, or can support the generation of large-scale new datasets for specific scenarios using prompting. These structured representations can then drive traffic simulators for behavioral rollouts and serve as conditioning for photorealistic video generation, enabling full multi-modal scenario generation that maintains both structural consistency and visual realism. Where diffusion video models provide visual diversity without structured guarantees and vectorized simulators provide structural control without visual grounding, we offer both: controllable structured synthesis realized as photorealistic generation.

3. ScenarioControl

In the following, we first describe our scene representation and vectorized scene generation method (Sec. 3.1). Next, we describe the proposed conditioning mechanisms for camera observations or natural language text prompts (Sec. 3.2), also illustrated by Figure 1. In Sec. 3.3 and Figure 2, we describe scenario-guided video generation through vision-language control.

3.1. Scene Representation and Generation

We generate the initial scene by jointly modeling the underlying map structure and the actors' initial states in a bird's-eye-view (BEV) representation with elevation information. Specifically, we generate each scene within a 64m × 64m field of view. A scene is then defined as a graph $\mathcal{I} = \{\mathcal{O}, \mathcal{M}\}$, comprising a set of objects $\mathcal{O}$ and a map structure $\mathcal{M} = \{\mathcal{L}, \mathbf{A}\}$. The map structure consists lane centerlines $\mathcal{L}$ and their connectivity graph $\mathbf{A}$. Our objective is to sample a scene $\mathcal{I} \sim p(\cdot|C)$ from some distribution p, conditioned on inputs C, which can be either an image captured from the perspective of an agent in the scene or a text prompt describing the scene.

In contrast to existing scenario diffusion approaches that abstract the environment in a purely two-dimensional BEV representation [9, 42], we augment the object parametrization with vertical structure by incorporating elevation z and object height h. This provides essential elevation cues for downstream camera and sensor-data synthesis, facilitating faithful reprojection of generated scenarios into the image domain as shown in Fig. 5.

3.2. ScenarioControl for Vectorized Latent Diffusion.

We propose a controllable vectorized latent diffusion model, as illustrated in Fig. 1, where we introduce the *cross-global control mechanism* that fuses sparse vector tokens with dense attention-based conditioning from prompts and images.

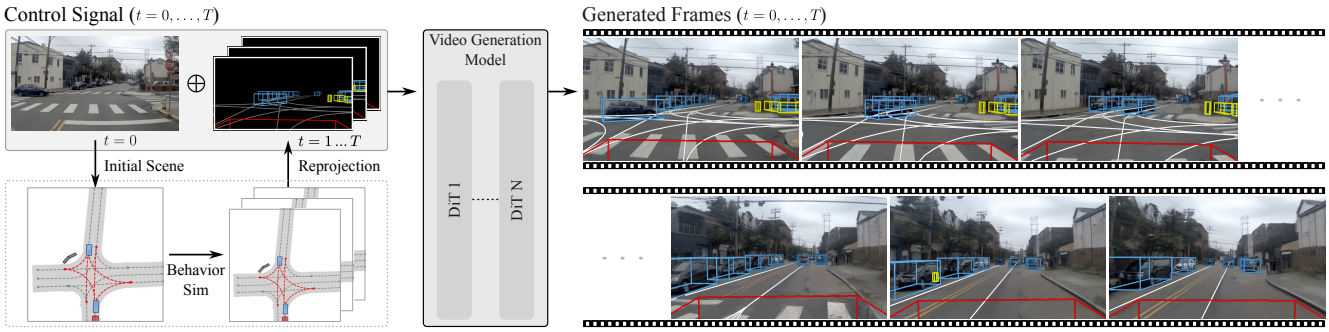


Figure 2. **Scenario-Controlled Video Generation.** Our method produces driving scene representations ($t = 0$) that we can simulate into scenario rollouts with a simulator, and use for downstream tasks such as video generation. The vectorized BEV scenario is projected into conditional camera-space layouts (BEV2Cam), and then fed into a LoRA-adapted Wan 2.2 (5B) model as a control signal for conditional video generation, together with the first frame.

First, we train a transformer-based vectorized autoencoder that encodes scene elements into a compact latent representation. We then train a diffusion model with ϵ-predictor ϵ_θ over the latents of the autoencoder $\mathbf{Z} = [\mathbf{Z}_O, \mathbf{Z}_\mathcal{L}]$ with variable cardinalities (N_o, N_l). With condition C, we minimize the DDPM ϵ-prediction objective [20] with

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\tau, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), C} \left\| \boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{Z}_\tau, \tau; C) \right\|_2^2, \tag{1}$$

where $\mathbf{Z}_\tau$ are noisy latents at step τ, composed of N_o object latents and N_l lane latents, and the noise vector decomposes as $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_O, \boldsymbol{\epsilon}_\mathcal{L})$. The conditioning inputs C comprise dense control signals C_I (*image*) and C_P (*prompt*) used in cross-attention, and the default control tokens $\bar{c}$ that encode the number of agents and lanes $N = (N_o, N_l)$ and the scene/domain label $\int$ (e.g., Singapore, Las Vegas, Boston, Pittsburgh in nuPlan), injected via AdaLN-Zero conditioning [37].

We next describe the proposed multi-modal control mechanism in more detail, which supports conditioning from both scene prompts and agent-perspective images.

Prompt Conditioning. Given a prompt describing the present scene and actors, we employ a text encoder $\mathcal{E}_P$ that extracts token embeddings as

$$\mathbf{E}_P = \mathcal{E}_P(C_P) \in \mathbb{R}^{M_P \times d_P}, \tag{2}$$

where M_P is the number of prompt tokens and d_P is the embedding dimension of the text encoder. The embeddings $\mathbf{E}_P$ are then projected through an MLP layer with the *linear projection weights* $\mathbf{W}_P$ to obtain control tokens $\mathbf{F}_P \in \mathbb{R}^{M_P \times d}$, where d denotes the latent dimension used for agent or lane representations

$$\mathbf{F}_P = \mathbf{E}_P \mathbf{W}_P. \tag{3}$$

These prompt control tokens enter the control mechanism/attention by providing keys/values and are combined with the self-attention outputs.

Agent-Perspective Image Conditioning. Given a forward-facing agent-perspective image C_I, e.g., a dashcam style image, we extract dense features with a vision backbone $\mathcal{E}_I$ and depth estimator $\mathcal{E}_D$ as

$$\begin{aligned} \mathbf{E}_{\text{feat}} &= \mathcal{E}_I(C_I) \in \mathbb{R}^{M_I \times d_{\text{feat}}}, \\ \mathbf{E}_{\text{depth}} &= \mathcal{E}_D(C_I) \in \mathbb{R}^{M_I \times d_{\text{depth}}}, \end{aligned} \tag{4}$$

where M_I denotes the number of image tokens, d_{feat} the feature dimension of the vision backbone, and d_{depth} the dimension of the estimated depth map. We use pretrained models for both the vision backbone and the depth estimator, and both are frozen during training. Both image features and depth maps are projected to the model's hidden dimension d via learned linear mappings. Further, we add a non-trainable sine-cosine positional embedding $\mathbf{P}$ to the image features [37] and depth maps, enforcing a shared spatial encoding across modalities

$$\mathbf{F}_m = \mathbf{E}_m \mathbf{W}_m + \mathbf{P}, \qquad m \in \{\text{feat}, \text{depth}\}. \tag{5}$$

where $\mathbf{W}_{\text{feat}}$ and $\mathbf{W}_{\text{depth}}$ are learned *linear projection weights* mapping modality-specific features to the shared hidden dimension d, and $\mathbf{P}$ ensures alignment within the same image coordinate frame across both image features and depth maps. Finally, we concatenate these position-aware tokens to form the control feature representation $\mathbf{F}_I = [\mathbf{F}_{\text{feat}}, \mathbf{F}_{\text{depth}}]$.

Cross-Global Control Mechanism. Given the conditioning features, $\mathbf{F}_I$ and $\mathbf{F}_P$, we introduce a control mechanism to steer scenario generation. We employ N_{DM} factorized attention blocks, each applying (in order) object-to-lane, lane-to-lane, lane-to-object, and object-to-object self-attention (SA) over the stacked object and lane tokens. The conditioning inputs are injected into each (CA) component, which we detail in the following section. An overview of the full mechanism is illustrated in Fig. 1.

Conditioning vectorized scene tokens (lanes and agents) on agent-perspective images or scene prompt embeddings is inherently *unaligned*: a single scene token may depend on evidence from arbitrary subsets of conditioning tokens (e.g., occluded actors, distant lane cues, or globally specified textual constraints). Although cross-attention can, in principle, model such dependencies by allowing each query to attend to all keys, it provides little inductive structure and can be sample-inefficient when learning global context.

We therefore compute cross-attention through two parallel branches that share the key/value projections of the conditioning stream. Given scene queries $\mathbf{Q} \in \mathbb{R}^{N_q \times d}$ and conditioning tokens $\mathbf{F} \in \mathbb{R}^{N_k \times d}$, we first compute cross-attention

$$\mathbf{Y}_1 \;=\; \mathrm{Attn}(\mathbf{Q},\, \mathbf{K}(\mathbf{F}),\, \mathbf{V}(\mathbf{F}))\,, \tag{6}$$

implemented efficiently with FlashAttention. In parallel, we introduce a small set of learned latent tokens $\mathbf{T} \in \mathbb{R}^{L \times d}$ that aggregate global context from $\mathbf{F}$, and expose this compact summary back to the scene queries

$$\mathbf{Y}_2 \;=\; \mathrm{Attn}\big(\mathbf{Q},\, \mathbf{K}(\tilde{\mathbf{T}}),\, \mathbf{V}(\tilde{\mathbf{T}})\big)\,, \tag{7}$$

with

$$\tilde{\mathbf{T}} \;=\; \mathrm{Attn}(\mathbf{T},\, \mathbf{K}(\mathbf{F}),\, \mathbf{V}(\mathbf{F}))\,, \tag{8}$$

with cost $O(LN_k + N_qL)$. We combine both branches $\mathbf{Y} = \mathbf{Y}_1 + \tanh(g)\,\mathbf{Y}_2$ with a learned gate g. The gate is initialized such that $\tanh(g) \approx 0$ (i.e., $g = 0$), ensuring that the module initially reduces exactly to standard cross-attention. During training, the model can then progressively improve by selectively incorporating the additional global-context pathway. Since both pathways share the $\mathbf{K}(\cdot)$ and $\mathbf{V}(\cdot)$ projections, the parameter overhead remains minimal. Finally, the output is fused with multi-head self-attention via AdaLN-Zero modulation.

Count Injection f_{count}. Graph-based representations offer a natural handle for controlling scene complexity: the number of lanes and agents can be set directly by initializing the corresponding numbers of lane and object nodes. For instance, generating a two-lane highway can be guided by instantiating the matching number of lane nodes, while in the image/prompt-conditioned setting, these counts can also be inferred from the conditioning signal. We therefore train a lightweight attention-based regressor f_{count} that predicts the number of agents and lanes (N_o, N_l) from the conditioning tokens $\mathbf{F}$.

We note that compared to raster encodings or post-hoc control modules (e.g., ControlNet/T2I-Adapter [36, 63]), our method directly operates on variable-length vector tokens, thereby preserving topology (lane connectivity) and scene structure.

3.3. Video Generation and Continuation

The use of vectorized scene graph representation $\mathcal{I}$ allows for direct BEV behavior simulation (with elevation) and camera sensor video synthesis without requiring an additional learned lifting step, as is the case for rasterized representations. For behavior simulation, the vectorized representation is used directly, while for video generation it is projected into the camera coordinate frame to create control inputs for a video diffusion model, both described in detail in the following.

Behavior Simulation. Real-world cameras capture far beyond the 64m of an initial scenario layout. We use diffusion outpainting to construct large-scale scenes, which are then temporally rolled out using a behavior model $\mathcal{B}_T$ to produce diverse and behaviorally consistent agent trajectories. From this, we obtain scenario rollout representations $\{\mathcal{I}_t\}_{t=1}^{T}$, which we reproject from BEV to camera space as wireframe sequences $\{\hat{\mathcal{C}}_{\text{wire},t}\}_{t=1}^{T}$.

Sensor Video Generation. We adapt a video generation model $\mathcal{V}_i$ to generate photorealistic clips from the behavior simulation. We train two variants: an image-conditioned model that uses the first frame C_I for appearance, and a prompt-conditioned model where the scene appearance is controlled by prompt description. Both are also trained to adhere to control sequences (wireframe renders of the behavior simulation) $\{\hat{\mathcal{C}}_{\text{wire},t}\}_{t=1}^{T}$ to obtain photorealistic multi-frame renders $\{\hat{I}_t\}_{t=0}^{T}$ that follow the simulated agent behavior (see Figure 2). Since we ground the video generation with the vectorized scene representation, we can transform the camera pose and generate videos from the perspectives of agents other than the ego agent whose camera capture was used to initialize the scene. We do so by re-rendering the wireframe representation with different camera extrinsics while maintaining a consistent text prompt.

After adaptation, the model achieves fine-grained control of traffic behavior – either as part of a video continuation task with the first-frame conditioning or in completely novel traffic situations from a prompt. Examples of controlled rollouts are reported in Figures 5 and 6, respectively.

3.4. Training

We adopt a two-stage training strategy. We first train the encoder f_e and decoder f_d to learn a reliable projection into the latent space. Next, we train the generative control mechanism and freeze the f_e, f_d weights and only train the dense condition blocks as described in Sec. 3.2, leveraging either the condition on Prompt or Image conditions.

Additionally, we also employ *classifier-free guidance* (CFG) [21]. Specifically, the conditioning inputs $\mathbf{F}_I$ and $\mathbf{F}_P$ are randomly dropped with probability p_{CFG} during training, encouraging the model to learn both conditional and unconditional behaviors. At inference, the guided prediction is

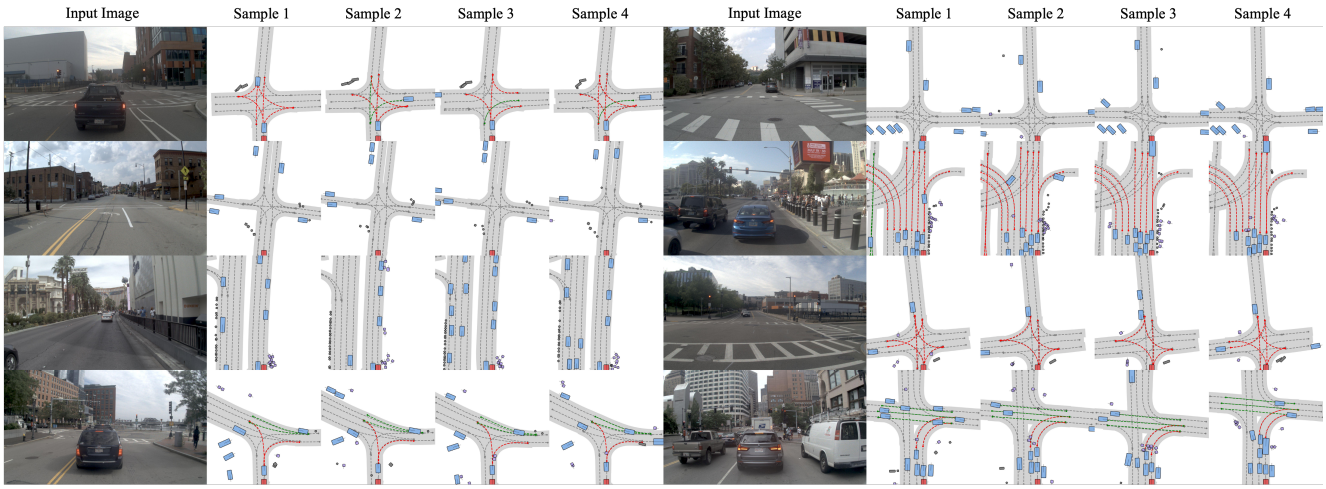


Figure 3. **Dashcam-Style Image-Conditioned Scene Generation**. Our model generates diverse initial scenes across different diffusion samples given a single input image. Elements with high certainty, such as vehicles in visible range, are found consistently across most samples. Elements with lower certainty, e.g., far away or outside the FOV, are sampled plausibly by our model.

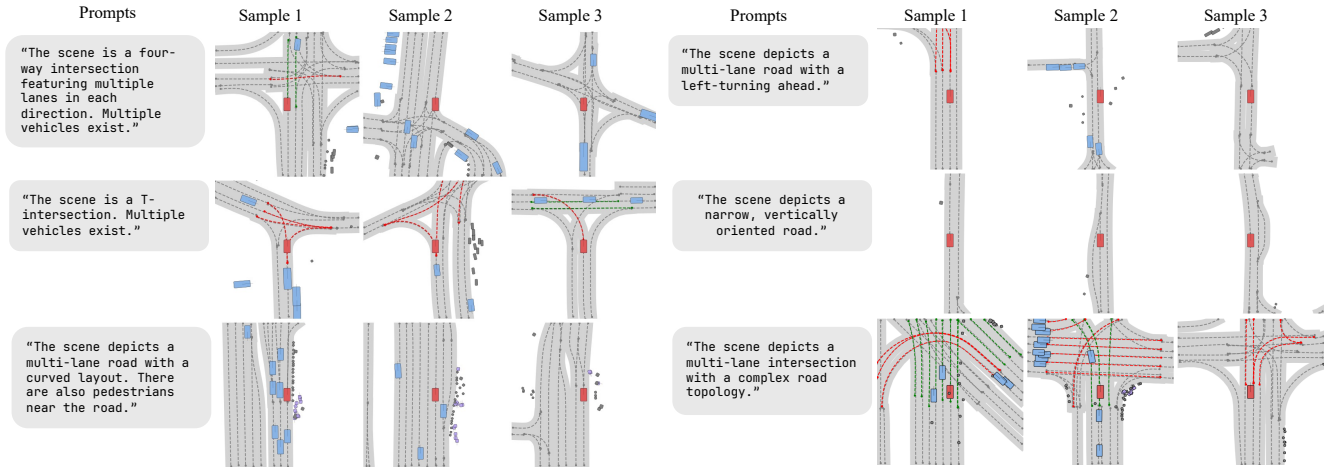


Figure 4. **Prompt-Controlled Scenario Generation**. Our model generates diverse initial scenes across different diffusion samples, with lane and object placement in each adhering to the text prompt. Text prompts are more open-ended than images, resulting in a larger variety across samples.

computed as

$$\epsilon_{\text{CFG}} = \epsilon_\theta(\mathbf{Z}_\tau, \tau; \varnothing) + w\big[\epsilon_\theta(\mathbf{Z}_\tau, \tau; C) - \epsilon_\theta(\mathbf{Z}_\tau, \tau; \varnothing)\big], \quad (9)$$

where w is the guidance weight controlling the strength of conditioning.

Scene Layouts. At test time, we support three generation modes that reflect the asymmetry between prompt and image conditioning. Text prompts can describe the full surrounding around the ego agent, whereas a single camera frame only constrains the visible region in front of the ego. We therefore define two canonical scene layouts with dimension of 64x64m: 1. an ego-centered crop $\mathcal{F}_P$ and 2. a forward-only crop $\mathcal{F}_I$, obtained by shifting the crop such that the ego lies near the edge, to maximize coverage ahead.

Crucially, we train the model on the same layout types used at inference, enabling consistent long-horizon rollouts. In practice, we use prompt-controlled synthesis for $\mathcal{F}_P$, image-conditioned completion for $\mathcal{F}_I$, and forward outpainting to extend either crop beyond the current field-of-view by sampling additional scene-graph nodes. Completion and outpainting are realized with masked denoising: latents corresponding to observed nodes are clamped, while the remaining tokens are sampled conditioned on C.

Collision Penalty. In practice, conditioning on prompt and images can induce *cluttered* scene hypotheses: images contain occlusions and missing context, while prompt is often underspecified. Both effects can place multiple agents into the same plausible region, resulting in overlapping boxes in the decoded scene graph. To encourage physically consistent layouts, we add a collision penalty

$\mathcal{L}_{\text{col}}$ during training. Concretely, we decode intermediate latents at selected timesteps and compute pairwise (i, j) overlaps (overlap(i, j)) between predicted agents i, j with an intersection-over-union of their corresponding bounding boxes. We define the collision regularization loss as

$$\mathcal{L}_{\text{col}} = \frac{1}{N} \sum_{i \neq j} \tanh\left(\frac{\text{overlap}(i, j)}{\zeta}\right), \tag{10}$$

where ζ controls smoothness. Since decoding is unreliable at low signal-to-noise ratios, we weight the penalty by $w_\tau = 1 - \sqrt{1 - \bar{\alpha}_\tau}$, emphasizing later diffusion steps (smaller τ) where the predicted geometry is more meaningful. This regularizer reduces agent overlap in the initial scene and improves global scene consistency under ambiguous conditioning.

Implementation Details. Further details on scene definitions, model architecture, and training and inference procedures are provided in the appendix.

4. Vision-language Scenario Dataset

To facilitate the training of our vision and language conditioned model, we curate a dataset consisting of images, natural language descriptions, and BEV maps. We select driving scenarios with corresponding camera captures from the nuPlan dataset [25]. To generate scene-level captions (e.g., "An intersection with a red light and multiple vehicles on the road. Pedestrians are standing on the sidewalk."), we first render BEV visualization images for each scene and use a VLM (GPT-4.1-mini) to produce descriptive captions.

This process results in a large-scale dataset containing approximately 500K high-quality captions, providing a rich foundation for training and evaluating our multimodal model. We provide additional details and dataset samples in the Appendix.

5. Experiments

Next, we first introduce the relevant evaluation metrics in Sec. 5.1. We then compare our proposed Cross-Global Control conditioning mechanism with other popular conditioning mechanisms in Sec. 5.2. Subsequently, we demonstrate superior adherence to control input with significant overlap in scene content in Sec. 5.3. Next, we analyze the inclusion to predict the actor and lane counts, f_{count}, and to suppress collisions through the loss $\mathcal{L}_{\text{col}}$. Lastly, we provide a generalization experiment on the Waymo motion dataset [47] in Sec. 5.5. As our method focuses on *controllable* scenario generation, unconditioned comparison with prior methods [9, 12, 42] are orthogonal but can still be found in the Appendix.

5.1. Evaluation Metrics

Our evaluation focuses on two key aspects: generation quality and controllability. For generation realism, we adopt the same lane- and agent-level metrics as [42] and are presented in the appendix. For controllability we introduce a set of control metrics to assess adherence to the provided conditioning signals that build on top of Agent Accuracy, Collision Rate and Control Adherence.

Agent Accuracy. As a first step, we match generated agents to ground-truth agents and evaluate placement accuracy using average precision (AP). We compute AP_δ under center-point distance thresholds $\delta \in \{1.0, 2.0, 3.0\}$ m and report their mean, yielding an equally weighted measure of how well the control signal localizes agents. For the image-conditioned setting, AP is evaluated only for agents within the camera field of view (FOV), whereas for the prompt-conditioned setting, AP is computed over all generated agents.

Collision Rate. We calculate the intersection over union over all predicted agents and report the fraction of scenarios with collisions in which actors and/or scene object bounding boxes intersect.

Control Adherence. To quantify how well the predicted global, lane, and agent attributes follow the intended controls, we report three complementary metrics: Cosine Control Similarity (CCS), Shuffled Perturbation Gap (SPG), and Control Sensitivity Correlation (CSC), which are detailed in the appendix. **CCS** measures the alignment between the conditioning signal and the corresponding change in the generated scene, capturing direct controllability. **SPG** evaluates causal dependence by comparing the model's response to correct versus randomly shuffled conditions; a larger gap indicates stronger conditional consistency. **CSC** measures the correlation between variations in the conditioning input and variations in the generated output, reflecting the sensitivity and smoothness of control. We evaluate across three levels (global, lane, and agent) to separately assess control over large-scale layout, map structure, and agents.

5.2. Analysis of Control Mechanism

We validate our cross-global control module by comparing it against a broad set of attention designs commonly used for fusing dense conditioning signals with token sequences. To our knowledge, this is the first study that systematically evaluates such mechanisms for conditioning a *vectorized 3D scene graph* on dense prompt or image features. Specifically, we compare against simple concatenation, full cross-attention [50], gated attention [1], linear attention [18], AgentAttention [19], SAAP cross-attention [33], windowed attention [10], deformable attention [67], and squeezed attention [22]. Quantitative results on lane- and agent-control metrics are reported in Tab. 3 with a qualitative result shown in Figure 7.

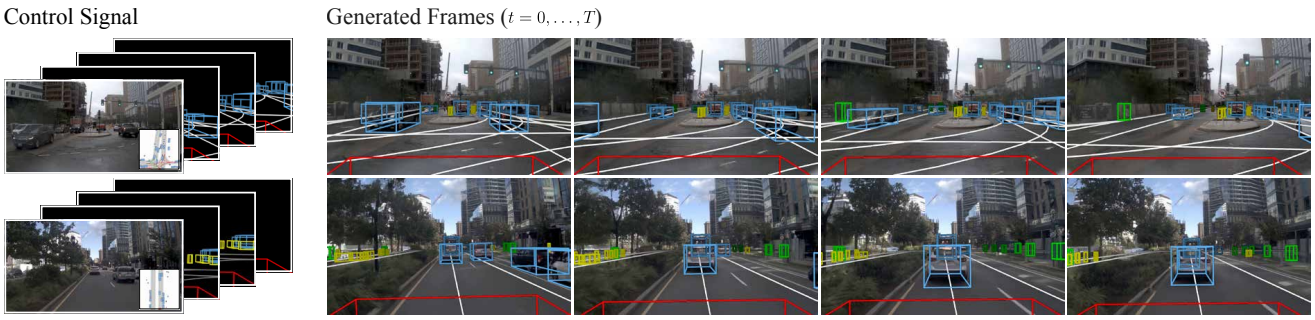
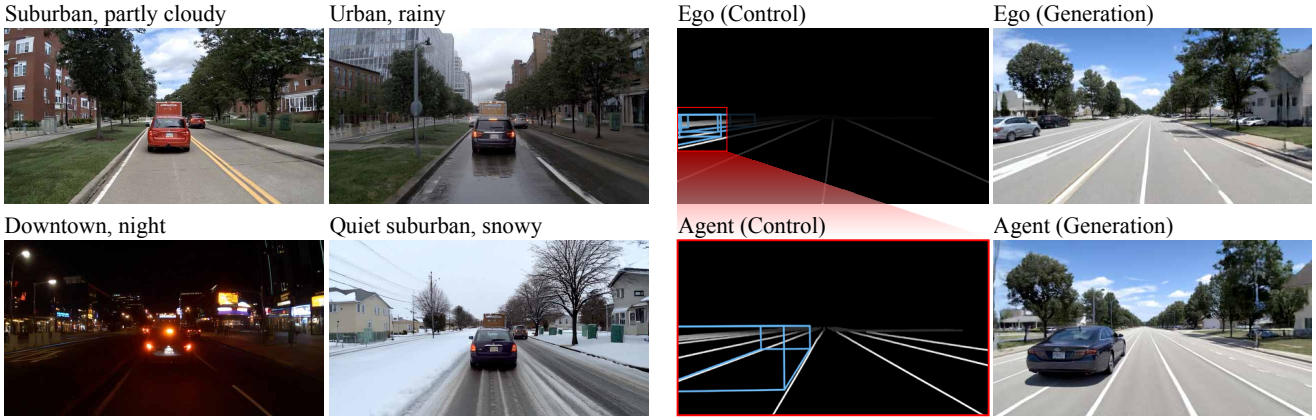


Figure 5. **Controllable Video Continuations.** Our video generation model respects control signals from the reprojected wireframe image sequences, and remains visually consistent with the initial frame. To show adherence with the control signal, we overlay the wireframe control signals over the generated sequences for qualitative evaluation. The samples shown are drawn from the NuPlan test split.



(a) Prompt-variations with controlled scenario (b) Different agent perspectives

Figure 6. **Prompt-conditioned Scenario Adaptations.** The prompt-conditioned video generation variant allows (a) different scene variants given the same generated traffic rollout, and (b) generating the same traffic situation from different perspectives by transforming the camera pose while keeping the text prompt constant.

Table 1. **Controllability Evaluation on Nuplan.** We evaluate controllability with three complementary metrics: cosine control similarity (CCS), shuffled perturbation gap (SPG), and control sensitivity correlation (CSC). Across both image- and text-conditioned settings, our method consistently improves all three scores, indicating stronger adherence to the specified controls $\mathcal{F}_i$.

$\mathcal{F}_i$	Method	Global Control			Lane Control			Agent Control		
		CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑
$i = P$	Scenario Dreamer [42]	0.9896	0.0082	0.3510	0.4074	0.3568	0.2885	0.1909	0.1732	0.0625
	ScenarioControl	**0.9910**	**0.0097**	**0.3764**	**0.4752**	**0.4335**	**0.3855**	**0.2992**	**0.3024**	**0.1499**
$i = I$	Scenario Dreamer [42]	0.9887	0.0141	0.4320	0.4400	0.3965	0.3166	0.1868	0.1800	0.0623
	ScenarioControl	**0.9926**	**0.0194**	**0.6274**	**0.6606**	**0.6108**	**0.6100**	**0.3853**	**0.3865**	**0.2255**

Table 2. **Controllability Evaluation on the Waymo Motion dataset.** We evaluate controllability on prompt-conditioned settings on Waymo, indicating strong cross-dataset generalization.

$\mathcal{F}_i$	Method	Global Control			Lane Control			Agent Control		
		CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑
$i = P$	Scenario Dreamer [42]	0.9922	0.0123	0.6701	0.3278	0.2775	0.2128	0.2264	0.2084	0.1220
	ScenarioControl	**0.9950**	**0.0171**	**0.7587**	**0.6739**	**0.6204**	**0.5179**	**0.3586**	**0.3529**	**0.2112**

Table 3. **Analysis of Cross-Global Control Mechanism.** We analyze the effect of our cross-global control mechanism with three complementary metrics: cosine control similarity (CCS), shuffled perturbation gap (SPG), and control sensitivity correlation (CSC). Across both image- and prompt-conditioned settings, our method consistently improves all three scores, indicating stronger adherence to the specified controls $\mathcal{F}_i$.

$\mathcal{F}_i$	Method	Global Control			Lane Control			Agent Control			Agent	Collision
		CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑	CCS ↑	SPG ↑	CSC ↑	AP ↑	RATE ↓
$i = P$	Concatenation	0.9898	0.0089	0.3606	0.4768	0.4286	0.3845	0.2786	0.2693	0.1340	23.03	24.43
	Full Cross-Attention [50]	0.9903	0.0093	0.3638	0.4803	0.4274	0.3830	0.2762	0.2722	0.1347	22.54	21.56
	Gated Attention [1]	0.9894	0.0082	0.3309	0.4013	0.3505	0.2695	0.1668	0.1624	0.0581	20.66	21.15
	Linear Attention [18]	0.9905	0.0086	0.3705	0.4416	0.3942	0.3505	0.2296	0.2303	0.0934	19.85	**19.18**
	AgentAttention [19]	0.9902	0.0091	0.3639	0.4827	0.4312	0.3893	0.2716	0.2667	0.1334	22.78	21.11
	SAAP Cross-Attention [33]	0.9904	0.0096	0.3628	0.4821	0.4278	0.3722	0.2556	0.2562	0.1264	22.03	27.51
	Windowed Attention [10]	0.9721	0.0078	0.1981	0.0899	0.1387	0.0893	0.0607	0.0655	0.0414	5.55	45.53
	Deformable Attention [67]	0.9802	0.0081	0.2503	0.3678	0.3301	0.2385	0.1439	0.1463	0.0525	21.65	23.62
	Squeezed Attention [22]	0.9882	0.0084	0.3404	0.4707	0.4202	0.3702	0.2021	0.2048	0.0869	**23.97**	20.00
	Cross-Global Control (Ours)	**0.9908**	**0.0098**	**0.3809**	**0.4900**	**0.4367**	**0.4006**	**0.3163**	**0.3096**	**0.1626**	23.36	20.07
$i = I$	Concatenation	0.9881	0.0153	0.4105	0.5564	0.5116	0.4094	0.2501	0.2537	0.1314	23.37	30.18
	Full Cross-Attention [50]	0.9924	0.0191	0.6137	0.6596	0.6098	0.6075	0.3766	0.3788	0.2195	32.00	16.07
	Gated Attention [1]	0.9896	0.0156	0.4637	0.5795	0.5315	0.4953	0.2958	0.2945	0.1514	25.13	14.92
	Linear Attention [18]	0.9901	0.0177	0.5299	0.6390	0.5899	0.5861	0.3527	0.3500	0.2012	27.20	16.92
	AgentAttention [19]	0.9895	0.0159	0.4729	0.5963	0.5478	0.5110	0.3032	0.3051	0.1632	24.54	17.18
	SAAP Cross-Attention [33]	0.9907	0.0180	0.5414	0.5849	0.5336	0.4670	0.2531	0.2700	0.1318	26.05	18.17
	Windowed Attention [10]	0.9902	0.0180	0.5467	0.6356	0.5848	0.5644	0.2880	0.2905	0.1450	23.60	16.22
	Deformable Attention [67]	0.9883	0.0151	0.4546	0.4799	0.4415	0.3194	0.2058	0.2155	0.1016	21.60	**13.82**
	Squeezed Attention [22]	0.9901	0.0169	0.4927	0.6183	0.5675	0.5423	0.3167	0.3186	0.1689	28.25	19.94
	Cross-Global Control (Ours)	**0.9926**	**0.0194**	**0.6274**	**0.6606**	**0.6108**	**0.6100**	**0.3853**	**0.3865**	**0.2255**	**33.46**	17.07

Overall, attention-based conditioning mechanisms consistently outperform simple concatenation, and this trend holds for both prompt- and image-conditioned control. In the image-conditioned setting, our method reduces collision rate by 43.4% and improves global CSC by 52.8% over concatenation; moreover, it also improves over full cross-attention with a 2.7% gain in agent CSC and a 4.6% gain in AP, while also achieving stronger global and lane control. In the prompt-conditioned setting, our method also surpasses all other baselines on most controllability metrics. This confirms that our cross-global attention better captures the dense-to-sparse correspondences needed to align scene tokens with the specified controls $\mathcal{F}_i$.

5.3. Analysis of Control Adherence

Figures 3 and 4 illustrate controlled initial scenes generated with image and prompt conditioning, respectively: image inputs preserve visible structure and yield plausible completions of unobserved regions, while text prompts focus on key differentiators like intersection types allowing for more diverse sampling.

Figure 5 shows controllable video continuations that stay visually consistent with the initial view and follow the projected wireframe control signals over time. Finally, Table 1 reports higher CCS, SPG, and CSC across global, lane, and agent levels for the conditioned model, confirming improved control adherence and semantic alignment without compromising generation stability relative to the unconditioned model in [42].

Table 4. **Analysis of f_{count} and $\mathcal{L}$** We report agent AP and simulated collision rates, while ablating the collision loss $\mathcal{L}_{col}$ and predicted agent counts f_{count}. Our final model significantly reduces collisions, with only a minor drop in AP.

$\mathcal{L}_{col}$	f_{count}	Collision RATE ↓	Agent AP ↑
✗	✗	19.81	39.55
✓	✗	14.07	38.99
✗	✓	18.78	38.70
✓	✓	13.36	38.03

5.4. Analysis of f_{count} and $\mathcal{L}_{\text{col}}$

Table 4 analyzes supervision for predicting the number of actors and lane elements from the conditioning signal via f_{count}, and for encouraging actor separation via the collision loss. The ablation study is conducted on a subset of the test set. Adding the collision loss consistently reduces collisions in the generated scenarios, regardless of whether f_{count} is enabled. Using f_{count} further lowers collision rate but also reduces AP, reflecting a trade-off: when the count is predicted rather than provided, the model tends to miss unseen or occluded actors outside the FOV. This reduces the number of placed agents, which decreases AP and, as a side effect, lowers the collision rate.

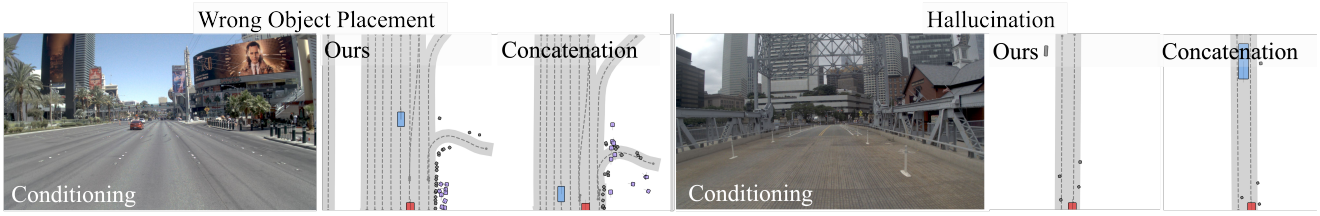


Figure 7. **Qualitative comparison with Concatenation.** Concatenation performs worse across all metrics and frequently violates the control input, e.g., generating off-FOV content (left) or hallucinated agents (right).

5.5. Generalization

We find that ScenarioControl generalizes across driving datasets to [47]. We evaluate transfer to the Waymo Open Motion dataset and report results in Tab. 2. By injecting explicit text prompts as control signals during generation, our model produces scenarios that better match the target distribution, outperforming Scenario Dreamer [42] across all reported metrics by up to *143*%.

5.6. Comparisons to Agent Placement Techniques

ScenarioControl generates both lane topology and agent placements, producing truly novel scene realizations with a given conditioning signal. It can also be applied in map-conditioned settings: by encoding a pre-existing lane topology from an existing map and denoising only the agent latents the model places vehicles on a provided road topology without altering the underlying geometry. This allows a direct comparison with methods such as TrafficGen [12] that focus solely on agent placement given a fixed map context.

TrafficGen [12] is an autoregressive method that generates vehicle initial states conditioned on the road context without any vision-language control, sampling actor positions following a general data distribution. Our text-conditioned model achieves an agent AP of 26.8%, compared to 5.5% for TrafficGen, representing a $\propto 4\times$ improvement. This confirms that grounding agent placement in a conditioning prompt provides a strong signal for realistic and accurate vehicle initialization, substantially outperforming unconditioned placement on the same maps. The experiments are carried out on 14688 scenarios from the test split of the Waymo Open Motion dataset [47].

6. Conclusion

We introduce ScenarioControl, a multimodal conditional method for generating controllable and realistic driving scenarios with text prompts or images. We achieve prompt and visual conditioning through our proposed cross-global attention mechanism for vectorized scenarios that fuse dense multimodal cues with sparse map–agent structures. Additionally, the proposed method supports sensor video generation, producing temporally consistent renderings that visually ground the synthesized scenarios. Through extensive evaluations, we confirm that our control mechanism is favorable for prompt and image conditions, while allowing for diversity and fidelity, effectively bridging structured scenario simulation with realistic sensor-level inputs and human-interpretable guidance.

Our method opens several promising directions for further research. An interesting direction is to leverage controllable scene generation to help autonomous vehicles anticipate traffic situations beyond line of sight information by synthesizing plausible continuations of partially observed environments. Extending our method to full multi-view and long-horizon visual conditioning could further improve consistency. Finally, extending the latent representation to unify initial scene generation and traffic simulation could allow the capture of additional semantics such as intent, weather, or interaction cues – providing even finer control over scenario structure and agent behavior.

References

[1] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022) 7, 9

[2] aigc apps: Videox-fun: A more flexible framework that can generate videos at any resolution and creates videos from images. https://github.com/aigc-apps/VideoX-Fun (2025), apache 2.0 License 20

[3] Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) 20

[4] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) 1

[5] Cai, P., Lee, Y., Luo, Y., Hsu, D.: Summit: A simulator for urban driving in massive mixed traffic. In: 2020 IEEE International Conference on robotics and automation (ICRA). pp. 4023–4029. IEEE (2020) 2

[6] Chen, A., Zheng, W., Wang, Y., Zhang, X., Zhan, K., Jia, P., Keutzer, K., Zhang, S.: Geodrive: 3d geometry-informed driving world model with precise action control. arXiv preprint arXiv:2505.22421 (2025) 3

[7] Chen, D., Zhu, M., Yang, H., Wang, X., Wang, Y.: Data-driven traffic simulation: A comprehensive review. IEEE Transactions on Intelligent Vehicles **9**(4), 4730–4748 (2024). https://doi.org/10.1109/TIV.2024.3367919 1

[8] Chen, X., Gao, X., Zhao, J., Ye, K., Xu, C.Z.: Advdiffuser: Natural adversarial example synthesis with diffusion models. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4539–4549 (2023). https://doi.org/10.1109/ICCV51070.2023.00421 3

[9] Chitta, K., Dauner, D., Geiger, A.: Sledge: Synthesizing driving environments with generative models and rule-based traffic. In: European Conference on Computer Vision. pp. 57–74. Springer (2024) 1, 2, 3, 7, 20, 21

[10] Dao, T.: FlashAttention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023) 7, 9

[11] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017) 1, 2

[12] Feng, L., Li, Q., Peng, Z., Tan, S., Zhou, B.: Trafficgen: Learning to generate diverse and realistic traffic scenarios. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 3567–3575. IEEE (2023) 1, 2, 7, 10

[13] Gao, R., Chen, K., Xiao, B., Hong, L., Li, Z., Xu, Q.: MagicDrive-V2: High-resolution long video generation for autonomous driving with adaptive control. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2025) 1

[14] Gao, S., Yang, J., Chen, L., Chitta, K., Qiu, Y., Geiger, A., Zhang, J., Li, H.: Vista: A generalizable driving world model with high fidelity and versatile controllability. Advances in Neural Information Processing Systems **37**, 91560–91596 (2024) 3

[15] Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The international journal of robotics research **32**(11), 1231–1237 (2013) 1

[16] Guan, Y., Liao, H., Wang, C., Liu, X., Zhang, J., Li, Z.: World model-based end-to-end scene generation for accident anticipation in autonomous driving. Communications Engineering **4**(1), 144 (2025) 1

[17] Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., et al.: Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. Advances in Neural Information Processing Systems **36**, 7730–7742 (2023) 1, 2

[18] Han, D., Pan, X., Han, Y., Song, S., Huang, G.: FLatten Transformer: Vision Transformer using Focused Linear Attention . In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5938–5948. IEEE Computer Society, Los Alamitos, CA, USA (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00548, https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00548 7, 9

[19] Han, D., Ye, T., Han, Y., Xia, Z., Pan, S., Wan, P., Song, S., Huang, G.: Agent attention: On the integration of softmax and linear attention. In: Computer Vision – ECCV 2024: 18th European Conference. pp. 124–140. Springer-Verlag, Berlin, Heidelberg (2024) 7, 9

[20] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models, 2020. URL https://arxiv. org/abs (2006) 4

[21] Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022), https://arxiv.org/abs/2207.12598 5

[22] Hooper, C.R.C., Kim, S., Mohammadzadeh, H., Maheswaran, M., Zhao, S., Paik, J., Mahoney, M.W., Keutzer, K., Gholami, A.: Squeezed attention: Accelerating long context length LLM inference. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 32631–32652. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.1568, https://aclanthology.org/2025.acl-long.1568/ 7, 9

[23] Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving (2023), https://arxiv.org/abs/2309.17080 1, 2

[24] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17853–17862 (2023) 1

[25] Karnchanachari, N., Geromichalos, D., Tan, K.S., Li, N., Eriksen, C., Yaghoubi, S., Mehdipour, N., Bernasconi, G., Fong, W.K., Guo, Y., et al.: Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 629–636. IEEE (2024) 1, 7, 20

[26] Kazemkhani, S., Pandya, A., Cornelisse, D., Shacklett, B., Vinitsky, E.: Gpudrive: Data-driven, multi-agent driving simulation at 1 million fps. arXiv preprint arXiv:2408.01584 (2024) 2

[27] Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., Zhou, B.: Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. IEEE transactions on pattern analysis and machine intelligence **45**(3), 3461–3475 (2022) 1, 2

[28] Li, X., Zhang, Y., Ye, X.: Drivingdiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In: European Conference on Computer Vision. pp. 469–485. Springer (2024) 1

[29] Li, Y., Yuan, W., Zhang, S., Yan, W., Shen, Q., Wang, C., Yang, M.: Choose your simulator wisely: A review on open-source simulators for autonomous driving. IEEE Transactions on Intelligent Vehicles **9**(5), 4861–4876 (2024) 1

[30] Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., Wang, X.: Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. CVPR (2024) 1

[31] Lin, H., Guo, Z., Zhang, Y., Niu, S., Li, Y., Zhang, R., Cui, S., Li, Z.: Drivegen: Generalized and robust 3d detection in driving via controllable text-to-image diffusion generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27497–27507 (2025) 1

[32] Lu, J., Wong, K., Zhang, C., Suo, S., Urtasun, R.: Scenecontrol: Diffusion for controllable traffic scene generation. In:

IEEE International Conference on Robotics and Automation (ICRA) (2024) 1, 2

[33] Mazaré, P.E., Szilvasy, G., Lomeli, M., Massa, F., Murray, N., Jégou, H., Douze, M.: Inference-time sparse attention with asymmetric indexing. arXiv preprint arXiv:2502.08246 (2025) 7, 9

[34] Meng, Y., Wu, H., Zhang, Y., Xie, W.: Scenegen: Single-image 3d scene generation in one feedforward pass. arXiv preprint arXiv:2508.15769 (2025) 2

[35] Mi, L., Zhao, H., Nash, C., Jin, X., Gao, J., Sun, C., Schmid, C., Shavit, N., Chai, Y., Anguelov, D.: Hdmapgen: A hierarchical graph generative model of high definition maps. arXiv (2021) 2, 20

[36] Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'24/IAAI'24/EAAI'24, AAAI Press (2024). https://doi.org/10.1609/aaai.v38i5.28226, https://doi.org/10.1609/aaai.v38i5.28226 5

[37] Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4195–4205 (2023) 4

[38] Pronovost, E., Ganesina, M.R., Hendy, N., Wang, Z., Morales, A., Wang, K., Roy, N.: Scenario diffusion: Controllable driving scenario generation with diffusion. Advances in Neural Information Processing Systems **36**, 68873–68894 (2023) 1

[39] Ren, X., Lu, Y., Cao, T., Gao, R., Huang, S., Sabour, A., Shen, T., Pfaff, T., Wu, J.Z., Chen, R., et al.: Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. arXiv preprint arXiv:2506.09042 (2025) 1, 3

[40] Rong, G., Shin, B.H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., Boise, E., Uhm, G., Gerow, M., Mehta, S., et al.: Lgsvl simulator: A high fidelity simulator for autonomous driving. In: 2020 IEEE 23rd International conference on intelligent transportation systems (ITSC). pp. 1–6. IEEE (2020) 2

[41] Rowe, L., Girgis, R., Gosselin, A., Carrez, B., Golemo, F., Heide, F., Paull, L., Pal, C.: Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning. arXiv preprint arXiv:2403.19918 (2024) 1, 2, 20

[42] Rowe, L., Girgis, R., Gosselin, A., Paull, L., Pal, C., Heide, F.: Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17207–17218 (2025) 1, 2, 3, 7, 8, 9, 10, 14, 20, 21

[43] Russell, L., Hu, A., Bertoni, L., Fedoseev, G., Shotton, J., Arani, E., Corrado, G.: Gaia-2: A controllable multi-view generative world model for autonomous driving. arXiv preprint arXiv:2503.20523 (2025) 1, 3

[44] Scanlon, J.M., Kusano, K.D., Daniel, T., Alderson, C.J., Ogle, A., Victor, T.: Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. Accident Analysis and Prevention **163**, 106454 (2021), https://api.semanticscholar.org/CorpusID:232285642 1

[45] Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025) 20

[46] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: 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) 1

[47] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: 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) 1, 7, 10

[48] Sun, S., Gu, Z., Sun, T., Sun, J., Yuan, C., Han, Y., Li, D., Ang, M.H.: Drivescenegen: Generating diverse and realistic driving scenarios from scratch. IEEE Robotics and Automation Letters **9**(8), 7007–7014 (2024). https://doi.org/10.1109/LRA.2024.3416792 2

[49] Tonderski, A., Lindström, C., Hess, G., Ljungbergh, W., Svensson, L., Petersson, C.: Neurad: Neural rendering for autonomous driving (2024) 2

[50] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 7, 9

[51] Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 20

[52] Wang, T., Zhang, C., Qu, X., Li, K., Liu, W., Huang, C.: Diffad: A unified diffusion modeling approach for autonomous driving. ArXiv **abs/2503.12170** (2025), https://api.semanticscholar.org/CorpusID:277066594 1

[53] Wang, X., Peng, P.: Prophetdwm: A driving world model for rolling out future actions and videos. arXiv preprint arXiv:2505.18650 (2025) 3

[54] Wang, X., Wu, Z., Peng, P.: Longdwm: Cross-granularity distillation for building a long-term driving world model. arXiv preprint arXiv:2506.01546 (2025) 3

[55] Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: Drivedreamer: Towards real-world-drive world models for autonomous driving. In: European conference on computer vision. pp. 55–72. Springer (2024) 1

[56] Wen, Y., Zhao, Y., Liu, Y., Jia, F., Wang, Y., Luo, C., Zhang, C., Wang, T., Sun, X., Zhang, X.: Panacea: Panoramic and controllable video generation for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6902–6912 (2024) 1

[57] Wu, Y., Zhang, H., Lin, T., Huang, L., Luo, S., Wu, R., Qiu, C., Ke, W., Zhang, T.: Generating Multimodal Driving Scenes via Next-Scene Prediction . In: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6844–6853. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2025). https://doi.org/10.1109/CVPR52734.2025.00642, https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.00642 1, 3

[58] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mt5: A massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies. pp. 483–498 (2021) 20

[59] Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: European Conference on Computer Vision. pp. 156–173. Springer (2024) 2

[60] Yang, X., Wen, L., Ma, Y., Mei, J., Li, X., Wei, T., Lei, W., Fu, D., Cai, P., Dou, M., Shi, B., He, L., Liu, Y., Qiao, Y.: Drivearena: A closed-loop generative simulation platform for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26933–26943 (2025) 3

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

[62] Zhang, K., Tang, Z., Hu, X., Pan, X., Guo, X., Liu, Y., Huang, J., Yuan, L., Zhang, Q., Long, X.X., Cao, X., Yin, W.: Epona: Autoregressive diffusion world model for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025) 3

[63] Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023) 5

[64] Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., Huang, A.C., Wen, Y., Hassanzadeh, K., Graves, D., Chen, D., Zhu, Z., Nguyen, N., Elsayed, M., Shao, K., Ahilan, S., Zhang, B., Wu, J., Fu, Z., Rezaee, K., Yadmellat, P., Rohani, M., Nieves, N.P., Ni, Y., Banijamali, S., Rivers, A.C., Tian, Z., Palenicek, D., bou Ammar, H., Zhang, H., Liu, W., Hao, J., Wang, J.: Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving (11 2020), https://arxiv.org/abs/2010.09776 1, 2

[65] Zhou, Y., Simon, M., Peng, Z., Mo, S., Zhu, H., Guo, M., Zhou, B.: Simgen: Simulator-conditioned driving scene generation. Advances in Neural Information Processing Systems **37**, 48838–48874 (2024) 3

[66] Zhou, Z., HU, H., Chen, X., Wang, J., Guan, N., Wu, K., Li, Y.H., Huang, Y.K., Xue, C.J.: BehaviorGPT: Smart agent simulation for autonomous driving with next-patch prediction. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=GRmQjLzaPM 1

[67] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (ICLR) (2021) 7, 9

Appendix

The appendix provides additional details and experiments supporting the findings in the main manuscript. In Sec. A, we report additional qualitative results. Sec. B provides further quantitative results. Additional implementation details are provided in Sec. C, and further information about the dataset is given in Sec. D. In Sec. E, we present additional evaluation details.

A. Additional Qualitative Results

As illustrated in Fig. 8, the image-conditioned model produces coherent initial scenes with well-aligned 3D structure. The projected 3D bounding boxes and road points clearly show that the generated layouts respect the underlying road geometry, with the ego vehicle (red) and surrounding vehicles (blue) positioned plausibly within the scene. These examples highlight the model's ability to capture both spatial context, lane structure and object placement from visual clues. While existing scenario generation methods operate purely in a 2D setting [42], our method generates 3D scenes.

We additionally observe that vehicle placement is generally more accurate than that of static objects or pedestrians. However, the locations and heights of these non-vehicle objects are still close to the expected positions and remain physically reasonable. It is important to note that our method is a scene generation method rather than a detection approach. Consequently, our primary objective is that the generated scenes both align with the image conditioning and adhere to plausible scene layouts and priors. For video continuation with image-conditioned scene generation, our primary focus lies in maintaining consistent and controllable vehicle behavior across frames.

We also provide examples of scenarios generated from text prompts in our test set, reported in Fig. 9. Notably, our model robustly handles prompts of varying complexity and successfully generates coherent scenarios even for highly detailed or intricate prompts.

Fig. 10, Fig. 11 and Fig. 12 further confirm that our model can expand an initial scene into diverse large-scale environments. In Fig. 10, we show several test-set images, each used to generate a 64×64m initial scene (F_I). From each image-conditioned initial scene, the model then produces multiple large-scale extensions via unconditional outpainting (F_O), illustrating the diversity achievable beyond the conditioned region. Combining conditioned initial scene generation and outpainting, Fig. 11 further validates that for each conditioning image we can sample multiple diverse large-scale scenes, which can be combined with CtRL-Sim for video generation.

Similarly, Fig. 12 presents examples where text prompts guide the generation of diverse initial scenes (F_P). Each of these initial scenes is then expanded into multiple large environments through outpainting (F_O), resulting in a wide variety of realistic road structures, including distinct intersection types, road layouts, and agent distributions.

Figure 8. Visualization of fully generated initial scenes conditioned on input images from the test-set. Projected 3D object bounding boxes and lane geometry are overlaid onto the first camera frame. Red indicates the ego vehicle, blue denotes other vehicles, green corresponds to pedestrians, yellow represents static objects, and white lines show the lane geometry.

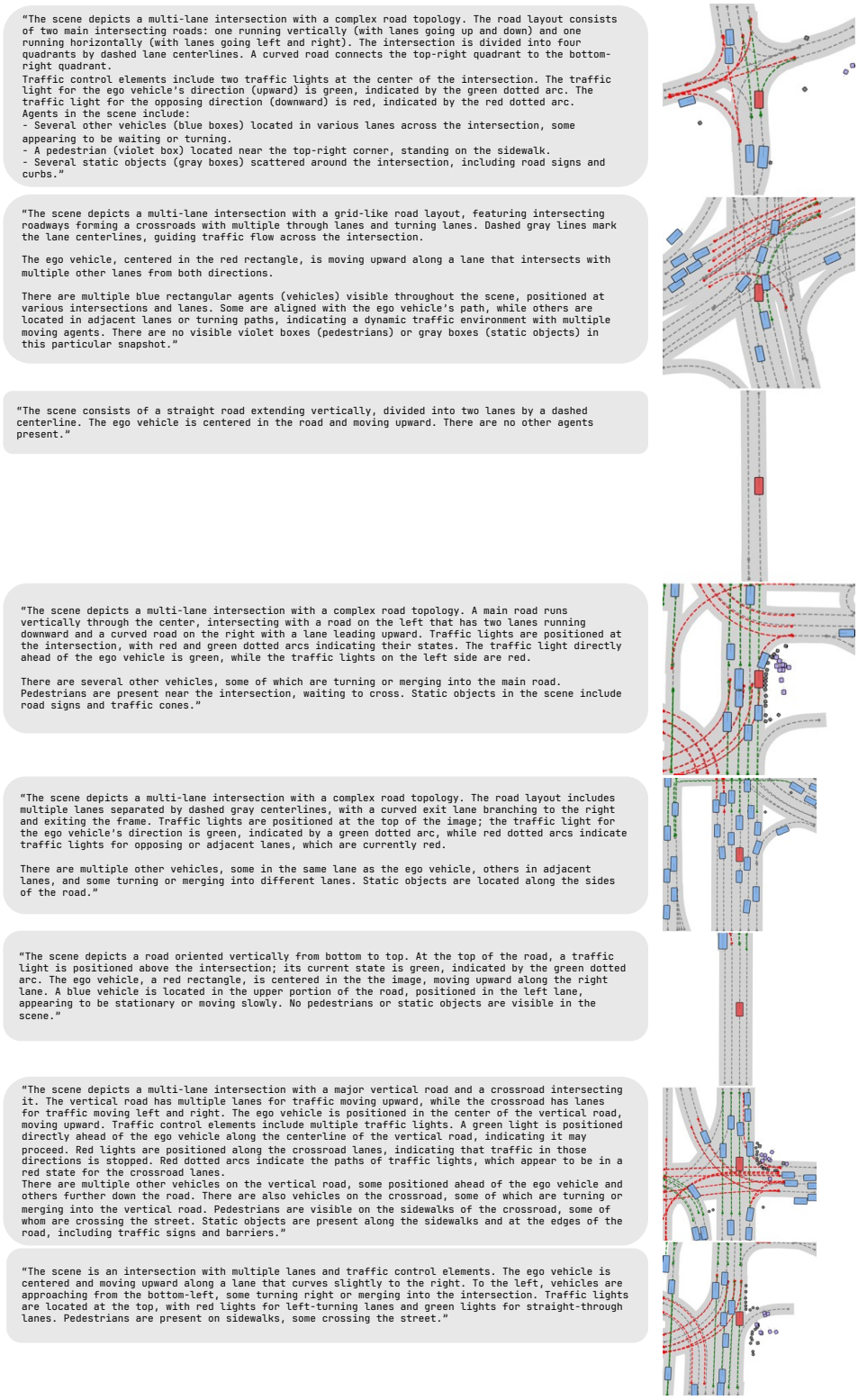


Figure 9. Visualization of generated initial scenes conditioned on test-set prompts. For each prompt, we show one generated scene. It can be observed that our model is capable of generating various scenarios, described by natural language. The ego vehicle is shown in red, and other vehicles are shown in blue.

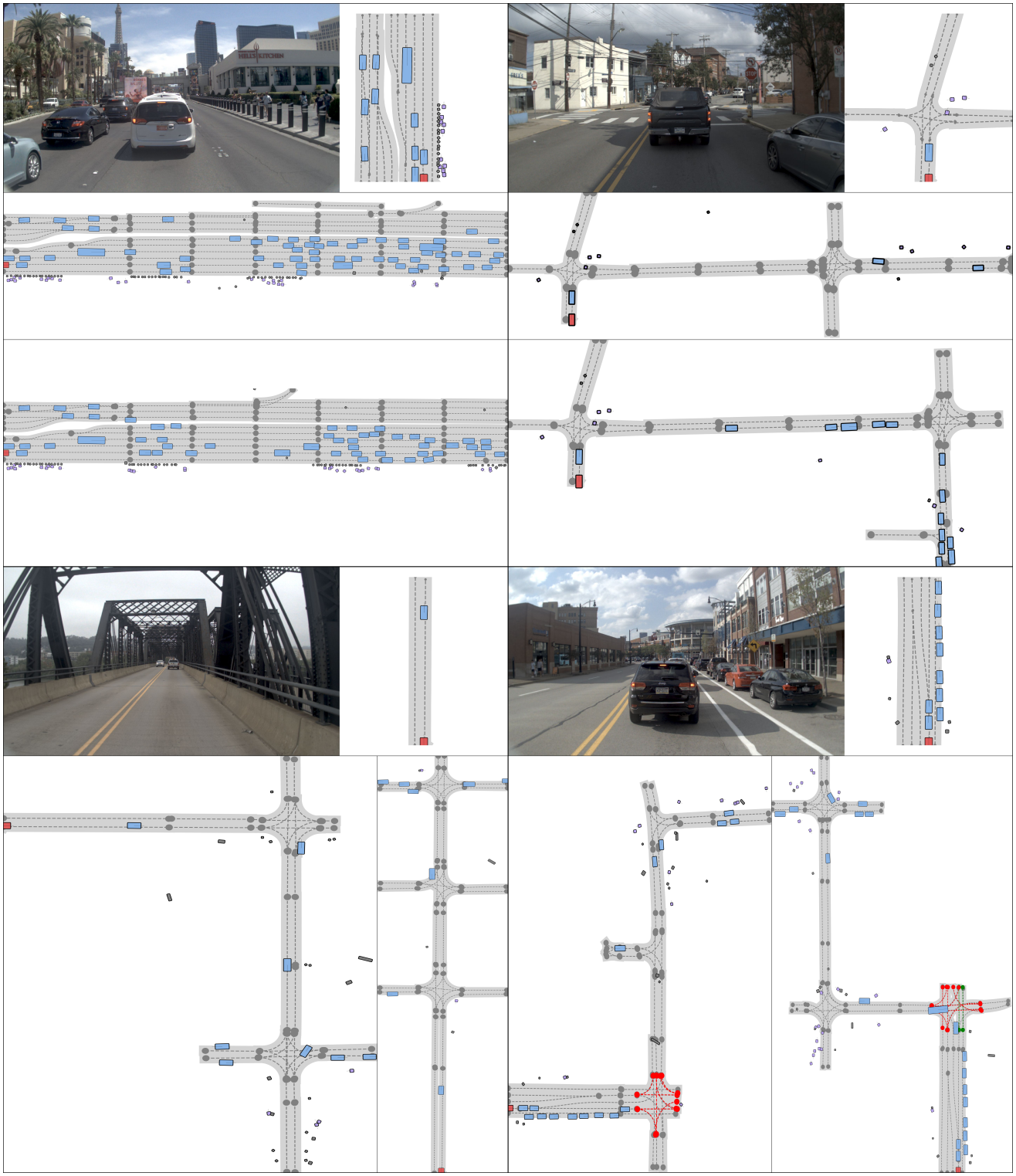


Figure 10. Four large-scale scenes generated using image conditioning accross four blocks. Starting from each image-conditioned initial scene, our model can outpaint beyond the visible region to produce diverse, extended environments. Red boxes represent the ego vehicle, and blue boxes denote other vehicles. In these examples, the generated scenes reach route lengths of up to 250 m. For each block top row shows the input image alongside the corresponding initial scene produced by our model, while the bottom rows illustrate two out-painted large-scale continuations conditioned on the same initial scene.

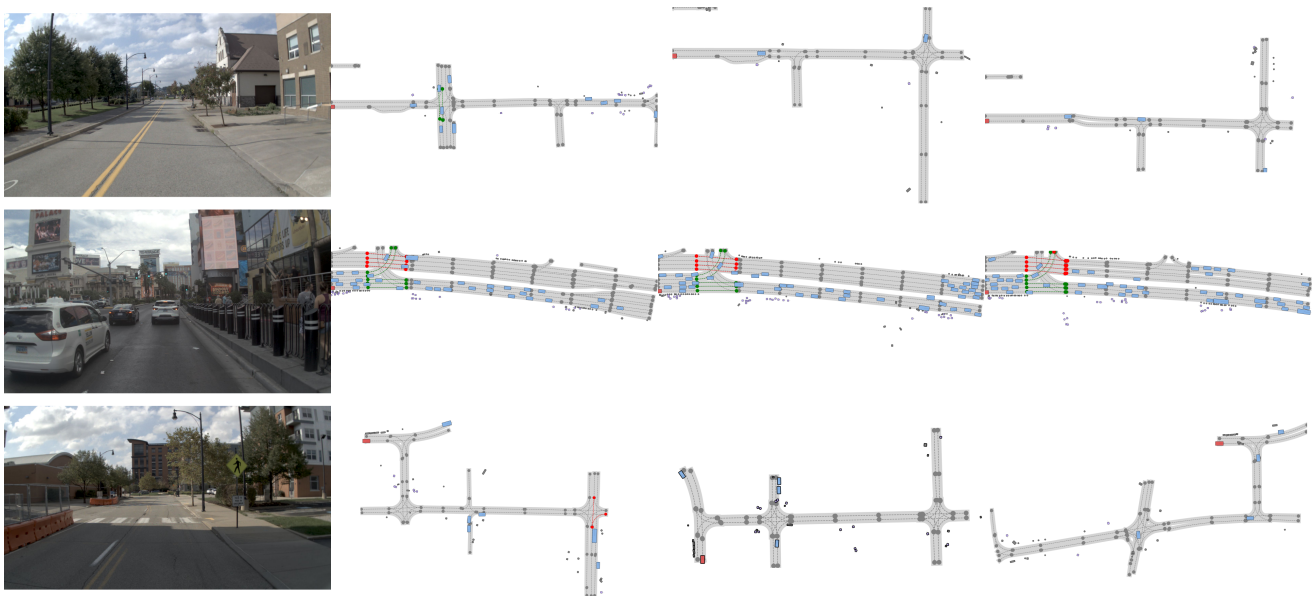


Figure 11. Three large-scale scenes generated from three different conditioning images (one per row). For each image, the model first generates a 64×64m initial scene (F_I) and then iteratively outpaints (F_O) three diverse large-scale realizations, illustrating the variety of road structures and agent distributions achievable from a single conditioning input. Route lengths extend up to 250m. The red box indicates the ego vehicle; blue boxes denote other vehicles.

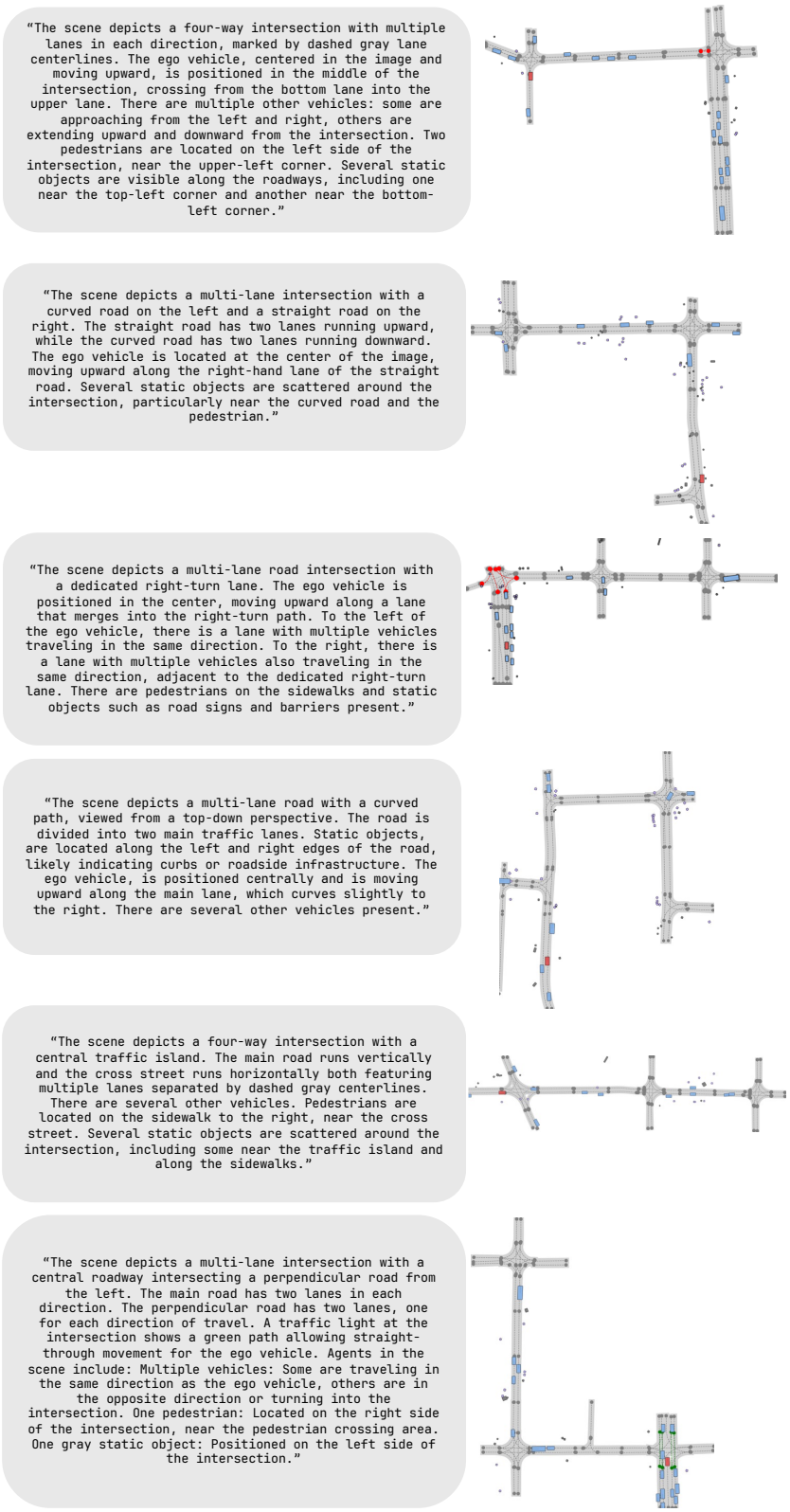


Figure 12. Prompt-conditioned generation of diverse large-scale scenes. Given a text prompt (left), the model first generates a 64×64m initial scene (F_T) and then produces multiple large-scale realizations via iterative outpainting (F_O), each yielding a distinct road layout and agent distribution. Six diverse examples are shown per row, with route lengths extending up to 250m. Red boxes indicate the ego vehicle; blue boxes denote other vehicles.

Table 5. **Analysis of f_{count} and $\mathcal{L}_{col}$ on control and safety metrics.** We report global, lane, and agent-level control metrics, together with agent AP and simulated collision rate, while ablating the collision loss $\mathcal{L}_{col}$ and predicted agent counts f_{count}.

$\mathcal{L}_{col}$	f_{count}	Global CCS ↑	Global SPG ↑	Global CSC ↑	Lane CCS ↑	Lane SPG ↑	Lane CSC ↑	Agent CCS ↑	Agent SPG ↑	Agent CSC ↑	Agent AP ↑	Collision Rate ↓
✗	✗	0.9885	0.0271	0.5114	0.6547	0.6176	0.5784	0.4158	0.4169	0.2637	39.55	19.81
✓	✗	0.9886	0.0268	0.5155	0.6543	0.6172	0.5810	0.4110	0.4128	0.2610	38.99	14.07
✗	✓	0.9818	0.0187	0.3737	0.5246	0.4724	0.4232	0.3516	0.3532	0.2066	38.70	18.78
✓	✓	0.9819	0.0185	0.3697	0.5242	0.4721	0.4247	0.3441	0.3479	0.2031	38.03	13.36

B. Additional Quantitative Results

In the following, we provide further results on unconditional generation in Sec. B.1 with metrics described below.

Lane Graph Quality. Following prior work [9, 35], we report the Urban Planning metrics: Connectivity, Density, Reach, and Convenience, which characterize the structural properties of the generated lane topology. We further evaluate Fréchet Distance (FD) for perceptual similarity, Route Length for global traversability, and Endpoint Distance for topological consistency.

Agent Quality. To evaluate agent generation, we compute the Jensen–Shannon divergence (JSD) between generated and real distributions of nearest-neighbor distance, lateral deviation, angular deviation, length, width, height, and speed. We also report the Collision Rate, which measures the percentage of scenes containing overlapping agent bounding boxes. Together, these metrics reflect geometric consistency and physical plausibility of generated agents.

B.1. Results Comparison to Unconditional Generation

Table 6 compares our lane-graph and initial agent bounding-box generation results with those of SLEDGE [9] and Scenario Dreamer [42] for unconditioned scene generation. Following Scenario Dreamer, we construct a balanced subset of the nuPlan [25] test dataset containing 12,500 samples per city. For distribution metrics, we randomly subsample 5,000 scenes ten times and report the mean and standard deviation. Our method achieves generation fidelity comparable to state-of-the-art baselines in both lane topology and agent initialization.

B.2. Analysis of Collision Loss and Agent Count

Table 5 provides an extended version of the ablation from the main paper (Table 4), including detailed controllability metrics at the global, lane, and agent levels (CCS, SPG, CSC), together with agent AP and simulated collision rate.

Consistent with the analysis in the main paper, adding the collision loss $\mathcal{L}_{col}$ substantially reduces collision rates while leaving control metrics and AP largely unchanged. The agent count can either be set directly by specifying the number of object nodes in the graph – giving full explicit control over scene population – or inferred from the conditioning signal via f_{count}. When using f_{count}, we observe a noticeable drop in control metrics and AP, which we attribute to missed or occluded actors that the model cannot reliably count from the conditioning signal alone. As discussed in the main paper, this also results in fewer placed agents and therefore indirectly lowers the collision rate. Overall, the combination of $\mathcal{L}_{col}$ and f_{count} achieves the lowest collision rate while maintaining comparable controllability scores.

B.3. Inference Latency

Inference latency is measured on a single NVIDIA A100 GPU with batch size 64. Image-conditioned inference takes 723 ms per sample on average, while text-conditioned inference requires 650 ms for initial scene generation. Subsequently, simulation steps require only outpainting, which takes 127 ms per timestamps and sample.

C. Implementation Details

We train both the prompt- and image-conditioned variants with a batch size of 64 across 4 NVIDIA A100-level GPUs for approximately 2 days. During training we use classifier-free guidance with a drop probability of 0.3. We extract image features f_{img} using DINOv3 [45], while we use ZoeDepth [3] to estimate mono-depth f_{depth}. The prompt tokens are extracted from T5 [58]. We utilized [41] as a base model. For video generation, we use VideoX-Fun [2] to run the LoRA adaptation on the public Wan 2.2 Fun Control (5B) checkpoint [51], a video diffusion model which supports video conditioning inputs. Both the prompt- and first-frame-conditioned video generation models are each trained on 4 NVIDIA A6000 GPUs for 2 days with default settings. In detail, we use a LoRA rank of 128 and network alpha of 64, and a learning rate of 1e-4. For the first-frame-conditioned model, the first frame C_I is concatenated with the wireframe renders and supplied as the control tensor for the 3D transformer. We mask the first frame in the control sequence by zeroing the inpainting mask for t=0 to ensure that $\hat{I}_0 = C_I$.

For the prompt-conditioned variant, we caption the first frame of each training sequence using GPT-4.1-mini and supply the resulting description as the text prompt during the LoRA adaptation, instead of the fixed generic caption used for the first-frame-conditioned model. No masking is applied, so all (including the first) frames are generated. At inference time, this allows arbitrary control of the scene's appearance while being fully grounded in a vectorized driving scenario.

C.1. Scene Layout Definitions

For our method, we define three scene layouts that correspond to the generation modes described in Sec. 3.4 of the main paper:

Table 6. **Evaluation of Unconditioned Scene Generation Quality.** We evaluate unconditioned lane-graph generation and initial agent bounding-box generation on the nuPlan dataset. For each metric the best-performing method is **bolded**. Overall, our method achieves scene-generation quality comparable to the unconditional baseline, while additionally providing controllability over the generation process. Note, Sledge[9] generates scenarios based on dense image projection and hence is not able to predict agent elevation.

$\mathcal{F}_i$	Method	Lane Graph Evaluation							Agent Evaluation						
		FD ↓	Conn. ↓	Dens. ↓	Reach ↓	Conve. ↓	Route Dist↑	End. Dist.↓	Near. Dev.↓	Lat. Dev.↓	Ang. ↓	Len. ↓	Wid. ↓	Height ↓	Speed ↓
$i = T$	SLEDGE (DiT-XL) [9]	1.68 ±0.04	1.70 ±0.04	1.78 ±0.09	0.52 ±0.02	1.69 ±0.12	35.83 ±8.37	0.42 ±0.29	0.47 ±0.01	0.46 ±0.01	3.18 ±0.03	11.23 ±0.10	10.43 ±0.06	— —	0.44 ±0.01
	Scenario Dreamer[42]	2.44 ±0.07	**0.22** ±0.02	0.21 ±0.03	0.12 ±0.01	0.73 ±0.09	**37.40** ±9.99	**0.40** ±0.88	**0.15** ±0.00	**0.11** ±0.00	**0.34** ±0.02	0.42 ±0.01	**0.11** ±0.01	1.60 ±0.02	0.09 ±0.01
	ScenarioControl (Ours)	**0.51** ±0.02	0.32 ±0.01	**0.17** ±0.02	**0.11** ±0.01	**0.35** ±0.04	36.99 ±11.87	0.56 ±1.01	**0.15** ±0.00	0.25 ±0.01	0.46 ±0.03	**0.39** ±0.02	0.14 ±0.01	**1.46** ±0.02	**0.05** ±0.01
$i = I$	Scenario Dreamer [42]	9.94 ±0.20	4.19 ±0.02	6.52 ±0.04	1.01 ±0.01	**2.91** ±0.05	59.68 ±0.09	**0.42** ±0.00	0.14 ±0.00	**0.21** ±0.00	**0.91** ±0.03	**0.31** ±0.01	**0.16** ±0.01	**1.54** ±0.03	0.15 ±0.01
	ScenarioControl	**2.26** ±0.01	**3.79** ±0.02	**5.99** ±0.03	**0.95** ±0.01	3.00 ±0.04	**59.84** ±16.76	0.63 ±0.00	**0.11** ±0.00	0.28 ±0.00	2.48 ±0.04	0.46 ±0.02	0.23 ±0.01	1.76 ±0.02	**0.08** ±0.01

- F_P defines an ego-centered layout covering the spatial range $[-32, 32] \times [-32, 32]$ and is used for *prompt-controlled scene generation*, where text prompts specify a scene surrounding the ego placed at the center.
- F_I positions the ego vehicle at the bottom of the scene with a field of view of $[0, 64] \times [-32, 32]$ and is used for *image-conditioned completion*. Shifting the ego to the bottom exposes a larger forward region that can be effectively guided by the conditioning image, since a single camera frame only constrains the visible region in front of the ego.
- F_O shares the same spatial layout but is partitioned at x=0 for *outpainting*. The observed half ($x < 0$) is treated as known context whose latents are clamped, while the forward half ($x > 0$) is generated via masked denoising conditioned on C. Note, F_O can start from both F_P and F_I.

C.2. Additional Model Details

We support 3-dimensional objects and the forward-facing scene layout (F_I). Using the control cross-attention module described in the main paper, the model size is 517M parameters for the prompt-conditioned version and 504M parameters for the image-conditioned version, due to differences in the modality projector input dimensions

C.3. Training and Inference Procedure

We adopt a two-stage training strategy. In Stage 1, we jointly pretrain the unconditional model with all three layouts: F_P and F_I for whole-scene generation, and F_O for conditioned partitioned-scene generation. For F_O, the model receives the bottom half of the latent representation as conditional input and is trained to predict the top half, enabling outpainting.

In Stage 2, we continue training F_P and F_I with their respective prompt or image conditioning, while also maintaining F_O training to preserve outpainting capability. Following classifier-free guidance, we randomly drop 30% of the conditioning input.

During sampling, all three layouts remain supported: prompt-conditioned generation (F_P), image-conditioned generation (F_I), and outpainting (F_O). To generate a large-scale initial scene, we first produce a whole-scene sample using either F_P or F_I. We then take the top half of the generated latent as the condition and apply F_O to outpaint the next segment, without any prompt or image conditioning. Iterating this process yields extended scenes covering up to 250m of route. These large-scale scenes are subsequently used for behavior simulation via $\mathcal{B}$ and for video continuation as described in Sec. 3.3 of the main paper. For each text prompt or image condition, our pipeline can (1) generate diverse initial scenes, (2) extend them into different large environments via outpainting, (3) produce multiple behaviorally distinct rollouts through simulation, and (4) render different visual continuations of each rollout.

D. Additional Multimodal Dataset Details

For the dataset introduced in this work, we first extract non-overlapping temporal sequences of 81 frames from each scenario in the nuPlan dataset. We then subsample the sequences by a factor of 10. This preprocessing yields the dataset used for unconditional pretraining:

- **Training set:** 139,271 sequences containing 696,351 samples.
- **Validation set:** 12,490 sequences containing 62,450 samples.
- **Test set:** 15,730 sequences containing 78,650 samples.

From the original temporal sequences, we retain only samples with corresponding camera images available and

subsequently subsample the sequences by a factor of 4. This preprocessing yields the following dataset splits:

- **Training set:** 21,295 sequences containing 447,154 samples.
- **Validation set:** 4,844 sequences containing 84,000 samples.
- **Test set:** 4,206 sequences containing 71,983 samples.

D.1. Caption Generation.

Given the rendered BEV image of a scene, we utilize the Vison-Language Model GPT 4.1-mini to perform automated captioning for the scenarios. In the following, we describe the detailed prompts we used for this process.

We first ask the VLM to generate a caption by defining its system prompt as:

```
You are an autonomous-driving BEV
(bird's-eye view) scene captioner.
Goal:  produce a concise,
specific, and faithful description
of the scene visible in the image.
Rules:
- Only describe what is visible.
If something is unclear, do not
mention it.
- Use the BEV conventions:  the
ego vehicle is centered, moving
from bottom to top.
- Do NOT explain the legend or
restate what colors or shapes
represent.
- Always output exactly two
sections:
(1) ROAD & CONTROL:
road topology, lanes,
intersections/merges/splits,
crosswalks, and traffic lights
relative to the ego path.
(2) AGENTS: other agents and
static obstacles described
using ego-relative positions
(front/behind/left/right, adjacent
lane) and motion if inferable.
- Prefer relative spatial language
and approximate counts (e.g.,
``several vehicles'').
- Keep the description compact:
2--5 sentences.
```

, followed by the caption instruction together with the input BEV image, as such

```
Caption this BEV snapshot.

Follow the required two sections:
ROAD & CONTROL, then AGENTS.

Use ego-centric relative positions.
```

E. Evaluation Details

In this section, we describe additional details of the evaluation metrics we use to assess the proposed work.

E.1. Control Metrics.

Let $\{(C_i, \mathcal{I}_i)\}_{i=1}^{N}$ denote a set of N conditioning inputs C_i (e.g., prompt or image) and their corresponding generated scenes $\mathcal{I}_i$, and let $f(\cdot)$ be a scene feature extractor (lane/object statistics, graph features, etc.). We define centered condition and scene features

$$u_i = \phi(C_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(C_j)$$

$$v_i = f(\mathcal{I}_i) - \frac{1}{N}\sum_{j=1}^{N} f(\mathcal{I}_j)$$

where $\phi(\cdot)$ is a condition embedding function (such as the prompt and image embedding networks we used).

Cosine Control Similarity (CCS). We measure the alignment between the direction of the conditioning signal and the corresponding change in the generated scene as

$$\mathrm{CCS} = \frac{1}{N}\sum_{i=1}^{N}\frac{\langle u_i, v_i\rangle}{\|u_i\|_2\,\|v_i\|_2}. \tag{11}$$

Shuffled Perturbation Gap (SPG). To quantify causal dependence on the correct conditioning, we compare CCS with aligned pairs to CCS with randomly shuffled conditions. Let π be a random permutation of $\{1, \ldots, N\}$ and define

$$\mathrm{CCS}_{\mathrm{correct}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\langle u_i, v_i\rangle}{\|u_i\|_2\,\|v_i\|_2}$$

$$\mathrm{CCS}_{\mathrm{shuffled}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\langle u_{\pi(i)}, v_i\rangle}{\|u_{\pi(i)}\|_2\,\|v_i\|_2}$$

Averaging over T random permutations $\{\pi_t\}_{t=1}^{T}$, the shuffled perturbation gap is

$$\mathrm{SPG} = \mathrm{CCS}_{\mathrm{correct}} - \frac{1}{T}\sum_{t=1}^{T}\mathrm{CCS}^{(t)}_{\mathrm{shuffled}}. \tag{12}$$

Control Sensitivity Correlation (CSC). Finally, we measure how smoothly the output changes with the conditioning by correlating pairwise condition and scene distances. Let

$$d_{ij}^{(c)} = \|u_i - u_j\|_2, \qquad d_{ij}^{(x)} = \|v_i - v_j\|_2,$$

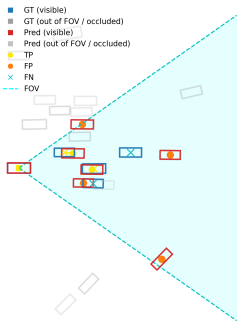


Figure 13. Visibility Filtering and Distance-based Matching for Image-conditioned Scene Generation. Only visible ground-truth and predicted vehicles inside the FOV are evaluated (blue and red), while others appear in gray. TP, FP, and FN under threshold $\delta = 2.0$ are shown in yellow, orange, and cyan and used to compute AP_δ.

and collect these into vectors $d^{(c)}, d^{(x)} \in \mathbb{R}^{N(N-1)/2}$ over all $i < j$. The control sensitivity correlation is the Pearson correlation

$$\mathrm{CSC} = \mathrm{corr}\Big(d^{(c)}, d^{(x)}\Big) = \frac{\sum_{i<j}\big(d_{ij}^{(c)} - \overline{d}^{(c)}\big)\big(d_{ij}^{(x)} - \overline{d}^{(x)}\big)}{\sqrt{\sum_{i<j}\big(d_{ij}^{(c)} - \overline{d}^{(c)}\big)^2}\sqrt{\sum_{i<j}\big(d_{ij}^{(x)} - \overline{d}^{(x)}\big)^2}}$$

A high CSC indicates that larger changes in the conditioning input induce proportionally larger changes in the generated scene, reflecting smooth and well-behaved controllability.

E.2. Agent Accuracy.

To evaluate the controllability of scene generation, we compute average precision (AP) of the agents. For the prompt-conditioned setting, we compute AP over the entire generated scene. For the image-conditioned setting, we evaluate only over *visible* objects, which are inside the camera's field of view (FOV) and not occluded by other vehicles.

Let $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_M\}$ and $\hat{G} = \{\hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_N\}$ denote the sets of ground-truth and generated centers after visibility filtering, where M and N are the numbers of visible objects in the ground-truth and generated scenes, respectively. For a threshold δ, we compute pairwise distances $d_{ij} = \|\hat{\mathbf{g}}_i - \mathbf{g}_j\|_2$ and form all candidate pairs with $d_{ij} \le \delta$. These pairs are sorted by distance, and greedy one-to-one matching is performed to obtain true positives (TP), false positives (FP), and false negatives (FN).

For each $\delta \in \{1.0, 2.0, 3.0\}$ meters, we compute Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, Recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, and $\mathrm{AP}_\delta = \mathrm{Precision} \times \mathrm{Recall}$. We adopt the product of precision and recall as an AP surrogate because our evaluation produces only a single precision–recall point per distance threshold, without confidence scores to generate a full curve. In this setting, the precision–recall product provides a simple and consistent measure that jointly penalizes over-generation and under-generation. The final score is the mean of AP_δ over the three thresholds. Evaluating only over visible objects ensures the metric reflects what is observable from the conditioning image. Fig. 13 shows an example of the visibility-aware matching of the image-conditioned scene generation.