Appendix

A. Discussions

To better understand our work, we supplement with the following question-answering.

Q1. *What makes Nexus stand out compared to driving simulators?*

Current simulators [10, 27] rely on hand-crafted rules, thus struggling with complex, out-of-scope scenarios. Generating corner cases requires manually positioning attack vehicles and adjusting traffic responses, making large-scale closed-loop adversarial scene generation impractical. While adversarial attacks [61] can create scenarios, log-replayed environmental agents lack realism and fail to ensure attack validity. In contrast, Nexus proposes a scalable, user-friendly approach for realistic and controllable hazard scenario generation. It leverages diffusion models to capture vehicle interactions, ensuring realism. Our method requires only goal points for the ego and attack vehicles, easily defined as lane center points, enabling efficient large-scale scenario expansion. Details of scene generation are in Appendix D.3.

Q2. *What is the definition of safety-critical scenarios and how to ensure they are realistic and feasible?*

Defination. A safety-critical scenario is a situation where one or more vehicles collide with the ego vehicle, which is rare to collect in real-world datasets like nuPlan. We utilize CAT [61] to generate risky behaviors from logged scenarios to ensure the reality and feasibility of training data, which uses a data-driven motion prediction model that predicts several modes of possible trajectories of each traffic vehicle. Please refer back to [61] for a detailed description of safety-critical scenarios.

Rationality of goal conditioning. Nexus emphasizes goal-controlled scenario generation, as it enables the convenient and scalable creation of *collision-prone* corner cases. Given the trajectory of the target agent, an attack can be easily executed by setting the attacker's goal to a future waypoint of the target agent. Nexus's design incorporates scenario interactions to enhance the realism of collisions.

Evaluation. Evaluating the quality of generated corner cases scientifically is a well-recognized challenge in academia. Our preliminary attempt combines quantitative and qualitative assessments. We use goal-driven kinematic metrics to measure trajectory authenticity, where Nexus excels, and ensure generated scenarios meet industry-standard corner cases [12, 54] like cut-ins, sudden braking, and collisions (see Fig. 11 for more visualizations).

Q3. Broader impact. *What are potential applications and future directions with the provided Nexus-data and the Nexus model, for both academia and industry?*

Datasets. Nexus-Data collects massive data from simulators, significantly enhancing the layout diversity of driving scenarios. This dataset provides the community with high-quality resources for studying complex agent interactions, multi-agent coordination, and safety-critical decision-making in autonomous driving.

Models. Beyond data augmentation, we believe our model can also drive broader applications within the community. This work showcases Nexus's potential as both a closed-loop world generator and a data engine. It could be adapted for downstream tasks, such as closed-loop training of autonomous driving agents [38]. Our model presents a promising generative world model, providing an alternative to traditional rule-based simulators. Please note that our model will be publicly released to benefit the community and can be further fine-tuned flexibly according to custom data within the industry.

Negative societal impacts. The potential downside of Nexus could be its unintended use in generating counterfeit driving scenarios due to the hallucination issues that may arise with diffusion models. We plan to introduce rule-based validation mechanisms, such as collision consistency checks, kinematic feasibility constraints, and behavioral plausibility tests, to filter out unrealistic generated scenarios. Besides, we plan to regulate the effective use of the model and mitigate possible societal impacts through gated model releases and monitoring mechanisms for misuse.

Q4. Limitations. *What are the issues with current designs and corresponding preliminary solutions?*

Visual synthesis is necessary for current end-to-end models in autonomous driving. Yet, datasets with visual data [41] are still much less abundant compared to those containing only driving logs [3]. Nexus currently lacks visual generation, limiting its use in applications requiring realistic sensor data, such as perception model training and end-to-end learning for autonomous vehicles.

However, as a work exploring how to incorporate world generators with generative models, the primary focus of this work is the decoupled diffusion for adaptive scene generation. Future work may integrate Nexus with neural radiance fields (NeRFs)[35] for high-fidelity 3D scene synthesis or video diffusion models[1, 60] for temporally consistent video generation, enabling full visual simulation of dynamic driving scenarios. This would allow Nexus to generate scenarios with both rich agent behaviors and realistic visual information, improving the training of end-to-end models as a world model.

Table 6. **Behavior distribution statistics.** Proportion (%) of agent behaviors in the dataset, excluding keeping forward. Our collected data provides a more balanced distribution for lane changes.

Dataset	Time (Hrs)	Inter. Passing	Left Turn	Right Turn	L. Lane Change	R. Lane Change	U-Turn	Stop
nuScenes [2]	5.5	13.1	18.0	10.2	5.0	2.5	0.0	4.1
nuPlan [3]	1.2K	13.8	1.5	1.6	14.4	14.6	0.9	46.8
Nexus-Data	540	35.3	1.7	2.5	22.2	23.3	1.2	10.0

B. Nexus-Data

B.1. Layout Diversity Highlights

We applied handcrafted rules to analyze behavior distributions in nuScenes [2], nuPlan [3], and our Nexus-Data shown in Tab. 6. For brevity, we omit the proportion of normal forward driving. Beyond forwarding, turning, and stopping, our dataset demonstrates greater diversity in lane-changing scenarios. Fig. 8 visually displays the top-down views of various dangerous driving scenarios, including collisions, quick stops, and reckless merging.

B.2. License and Privacy Considerations

All the data is under the CC BY-NC-SA 4.0 license[1]. Other datasets (including nuPlan [41], Waymo Open [49], Metadrive [27]) inherit their own distribution licenses. We only distribute lane geometries and vehicle trajectories, ensuring compliance with dataset licenses and removing personally identifiable information to prevent privacy risks.

C. Implementation Details of Nexus

C.1. Model Design

As shown in Tab. 7, the Nexus architecture is built upon SimpleDiffusion [16]. The model uses rotary embedding for position encoding, which is based on both physical time and denoising steps simultaneously. The backbone incorporates four layers of TemporalBlock and SpatialBlock, enabling the model to capture temporal and spatial dependencies through attention and feedforward layers. The Global Encoder uses Perceiver IO [22] for map feature extraction. The final output is projected through a linear layer.

C.2. Training Details

Nexus is trained over 1200 hours of real-world driving logs from the nuPlan dataset [3] and 480 hours of collected data from the simulator [27]. The training data consists of 10-second driving logs sampled at 2Hz, resulting in 21 frames per sequence (4 historical, 1 current, and 16 future frames). The dataset includes 528K scenarios, each covering a 104-meter range. Training on Waymo [49] used 531K scenes, each lasting 9 seconds, sampled at 2Hz, with 2 historical frames, 1 current frame, and 16 future frames. The training task follows a denoising diffusion process, where random noise is added to each agent token across the entire sequence. The model is then trained to recover the original sequence, learning to reconstruct motion trajectories under noisy conditions.

We train the model for 80K iterations on 8 GPUs with a batch size of 1024 with AdamW [33]. The initial learning rate is 1×10^{-3}. We use a learning rate scheduler with a warm-up and cosine decay strategy. After the warm-up, the learning rate will gradually decrease according to a cosine function. The default GPUs in most of our experiments are NVIDIA Tesla A100 devices unless otherwise specified.

C.3. Sampling Details

Classifier guidance for human-behavior alignment. Diffusion models can generate unrealistic driving scenarios due to randomness, requiring human-guided constraints to enhance scene quality. As shown in Fig. 9, we consider three human-behavior rubrics: 1) Collision avoidance: At each step t, if two vehicles' bounding boxes overlap, they are pushed apart along their center-connecting line. It can be written as the following equation:

[1] https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en

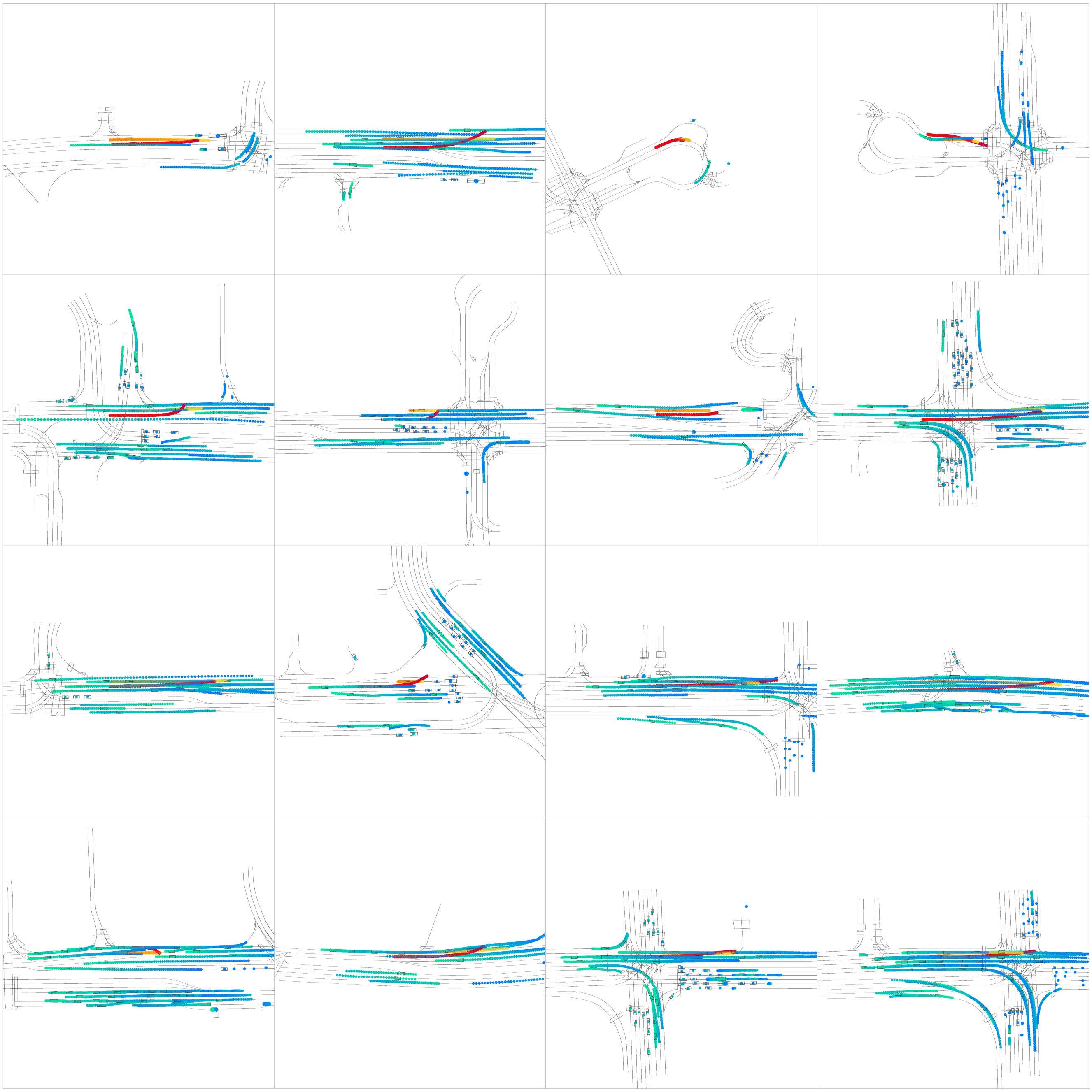

Figure 8. **Various safety-critical layouts from Nexus-Data.** All scenarios are initialized by the nuPlan [3] and generated by adversarial interactions [61] within simulators.

$$f_{\text{collision}}(\mathbf{x}^t, t) = \left[\mathbf{x}^t_{\text{loc}}, \mathbf{x}^{t,3:d}\right], \tag{5}$$

$$\text{where } \mathbf{x}^t_{\text{loc}} \leftarrow \mathbf{x}^t_{\text{loc}} + \lambda_t \sum_{i \neq j} \mathbb{I}\{B(\mathbf{x}^t_i) \cap B(\mathbf{x}^t_j) \neq \varnothing\} \cdot \frac{\mathbf{x}^t_{i,\text{loc}} - \mathbf{x}^t_{\text{j,loc}}}{\|\mathbf{x}^t_{i,\text{loc}} - \mathbf{x}^t_{j,\text{loc}}\|}, \tag{6}$$

where λ_t is a scalar coefficient used to control the extent of separation at time t. $\mathbb{I}$ is an indicator function that takes the value 1 when the bounding boxes of vehicle i and vehicle j overlap and 0 otherwise. B is the function used to form the vehicle's

Table 7. **Architecture of the Nexus Model.**

Component	**Details**
Top-Level Model	LightningModuleWrapper
Main Model	Nexus
Diffusion Backbone	SimpleDiffusion
Cross Attention	LayerNorm (256) + MultiHeadAttention
Attention Projection	Linear(256, 256)
Input Projection	Linear(25, 256)
Timestep Embedder	Linear(256, 256) + SiLU + Linear(256, 256)
Backbone Structure	CombinedAttention with TemporalBlock & SpatialBlock
TemporalBlock	LayerNorm(256) → MultiHeadAttention(256) → LayerNorm(256) → FeedForward MLP → SiLU + Linear(256, 1536) (AdaLN Modulation)
SpatialBlock	LayerNorm(256) → MultiHeadAttention(256) → LayerNorm(256) → FeedForward MLP → SiLU + Linear(256, 1536) (AdaLN Modulation)
MultiHead Attention	Linear(256, 256) with Dropout(0.0)
FeedForward MLP	Linear(256, 1024) + GELU + Linear(1024, 256)
Final Normalization	LayerNorm(256)
Output Projection	Linear(256, 8)
Global Encoder	PercieverEncoder
Cross Attention	LayerNorm(7) + MultiHeadAttention(256)
Self-Attention	2x SelfAttention Blocks
Other Modules	MapRender, NaivePlanner

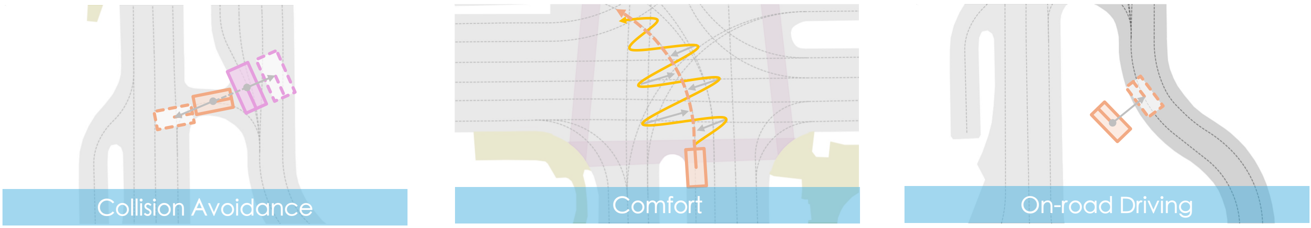

Figure 9. **Classifier guidance with human-behavior alignment.** The three different constraints are applied at each sampling step, contributing to the realism of the generated scenario.

bounding box. The fractional term represents the unit direction vector of the centerline between vehicle i and vehicle j.

2) Comfort: Enforcing smooth longitudinal and lateral accelerations by averaging adjacent trajectory points.

$$f_{\text{comfort}}(\mathbf{x}^t, t) = \left[\mathbf{x}^t_{\text{loc}}, \mathbf{x}^{t,3:d}\right], \tag{7}$$

$$\text{where } \mathbf{x}^t_{\text{loc}} \leftarrow \mathbf{x}^t_{\text{loc}} - \lambda_t \mathbf{a}^t, \tag{8}$$

$$\mathbf{a}^t = \frac{1}{2}(\mathbf{x}^t_{\tau-1,\text{loc}} - 2\mathbf{x}^t_{\tau,\text{loc}} + \mathbf{x}^t_{\tau+1,\text{loc}}). \tag{9}$$

First, the longitudinal and lateral accelerations a^t are approximated using the second-order difference at time τ and smoothed by averaging adjacent trajectory points. Then, the trajectory is refined by subtracting a proportion $lambda_t$ of the acceleration, reducing abrupt speed changes for smoother motion.

3) On-road driving: Pull the vehicle toward the nearest centerline point when it strays too far.

$$f_{\text{on road}}(\mathbf{x}^t, t) = \left[\mathbf{x}^t_{\text{loc}}, \mathbf{x}^{t,3:d}\right], \tag{10}$$

$$\text{where } \mathbf{x}^t_{i,\text{loc}} \leftarrow \mathbf{x}^t_{i,\text{loc}} + \lambda_t \mathbb{I}\{\|\mathbf{x}^t_{i,\text{loc}} - \mathbf{c}^t_i\| > d_{\text{th}}\} \cdot (\mathbf{c}^t_i - \mathbf{x}^t_{i,\text{loc}}), \tag{11}$$

$$\mathbf{c}^t_i = \text{argmin}_{l,n} \|\mathbf{x}^t_{i,\text{loc}} - \mathbf{c}_{l,n,\text{loc}}\|. \tag{12}$$

The vehicle identifies the closest lane point $\mathbf{c}^t_i$ among all points $\mathbf{c} \in \mathbb{R}^{L\times N\times D}$ by minimizing the Euclidean distance using $\arg\min_{l,n}$. When the deviation exceeds the threshold d_{th}, the vehicle adjusts its position by moving from $\mathbf{x}^t_{i,\text{loc}}$ toward the closest centerline point $\mathbf{c}^t_i$, with the adjustment magnitude controlled by λ_t.

Sampling. The sampling process is inherited from SimpleDiffusion [16]. It starts with random Gaussian noise and is performed by Denoising Diffusion Implicit Models (DDIM) [47] for 32 steps. For classifier guidance [45], we set the total value of λ for the three constraints to be 0.2. If more than two constraints are active simultaneously, the value of λ will be evenly distributed among them. The sampling speed is **206** milliseconds per step per batch.

D. Experiments

We conduct extensive experiments on multiple datasets to evaluate the performance of our method. Our baseline is built on a reproduced full-sequence training Diffusion Policy [6]. For comparison convenience, we trained two models on the nuPlan and Nexus-Data datasets, respectively, namely Nexus-*Full* and Nexus, adopting the same training strategy.

D.1. Protocols and Metrics

ADE: It measures the average displacement differences between the generated and ground truth trajectories, excluding goal points and invalid trajectory points from the calculation.

$R_{\textbf{road}}$**:** It measures the off-road rate of vehicles in the generated scenes. Off-road instances are detected by checking whether a vehicle's center deviates from its assigned centerline at each timestep. The rate is calculated as the number of vehicles that have gone off-road divided by the total number of valid vehicles.

$R_{\textbf{col}}$**:** It measures the collision rate among agents in the generated scenes. Collisions are detected by checking for overlaps between agent bounding boxes at each timestep. The rate is calculated as the number of collided vehicles divided by the total number of valid vehicles.

$M_{\textbf{k}}$**:** It measures the stability of generated trajectories using the average of four metrics: tangential and normal acceleration along the heading at each timestep and their derivatives (jerk). Lower values indicate smoother and more comfortable trajectories.

Composite Metric: It is a comprehensive metric for evaluating the realism of scene generation. During evaluation, the model generates a scene 32 times based on a 1-second history, forming a distribution. The likelihood between this distribution and the ground truth is then computed across factors like speed, distance, and collisions.

Score, $S_{\textbf{col}}$**, and** $S_{\textbf{p}}$**:** It is the main metric for assessing the reasonableness of agent planning trajectories in the nuPlan closed-loop evaluation. The metric considers comfort, collisions, road adherence, lane changes, and mileage completion, scoring 1 for success and 0 for failure per scenario. S_{col} and S_{p} are sub-metrics measuring trajectory collision rate and distance completion rate, respectively.

D.2. Evaluation Tasks

Free exploration. This task conditions the past 2 seconds of all vehicle states and lane centerlines from the nuPlan driving log to freely generate an 8-second future scene at a 0.5-second time interval. In the generation process, the noise level of each token is determined by the scheduling strategy. Invalid vehicles at the corresponding timestep are ignored. In the experiments, we used off-the-shelf IDM [56] and GUMP [17], as well as our implementations of Diffusion Policy [6] and SceneDiffuser [25].

Conditioned generation. On top of free exploration, goal points are added to valid vehicles by setting the token's noise level at that timestep to 0 during inference.

Waymo open sim agent evaluation. The Waymo evaluation requires generating 32 future scene predictions based on 1 second of historical observations, including vehicles, pedestrians, and cyclists. The evaluation is conducted at 10Hz, and we interpolate the 2Hz model to match the required scene frequency.

Closed-loop evaluation. In the nuPlan closed-loop evaluation, the environment and agent are treated separately. The agent predicts an 8-second trajectory based on 2 seconds of historical observations and takes 0.1-second actions. The environment

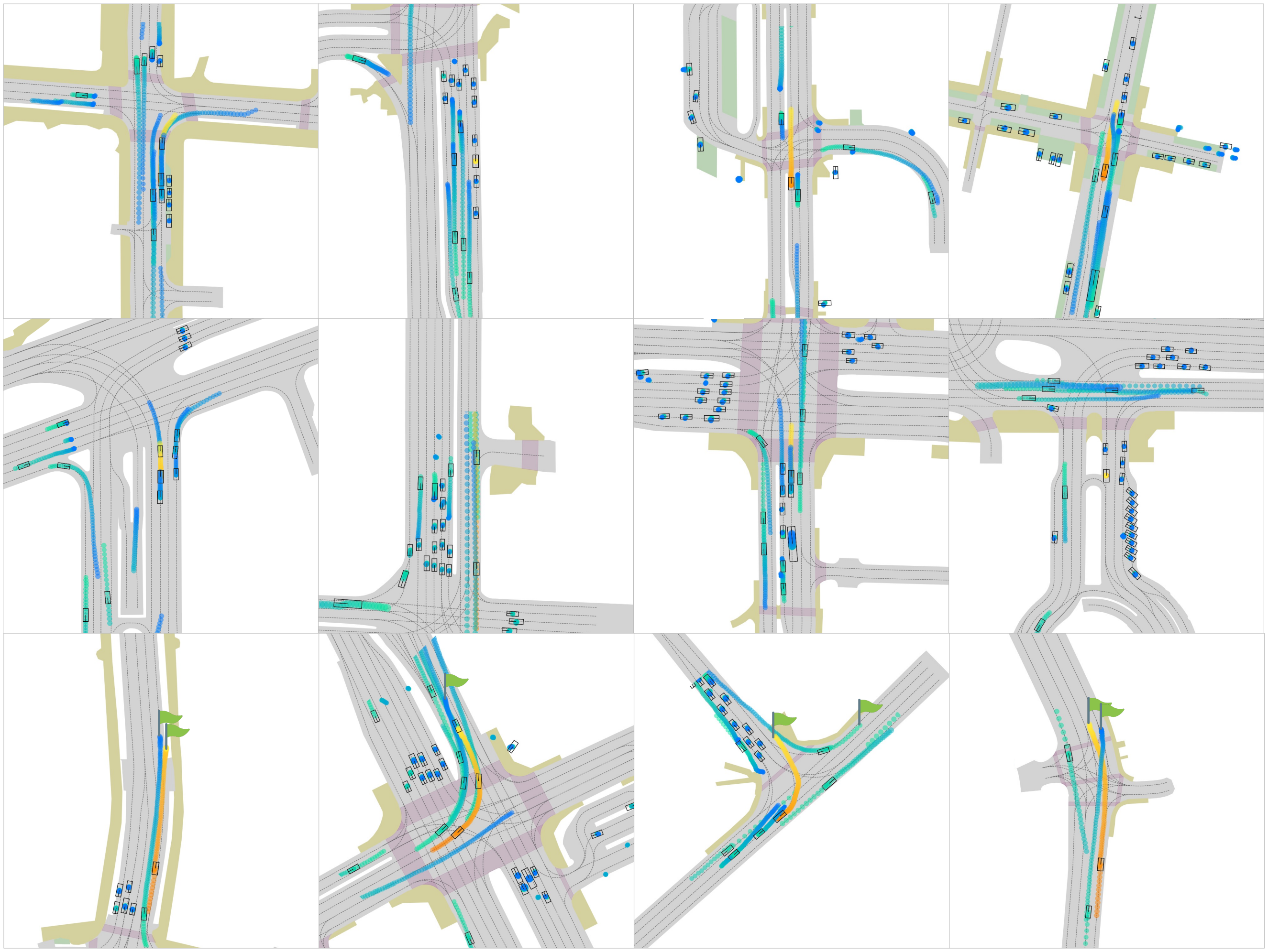

Figure 10. **More visualizations of Nexus on free exploration and conditioned generation.**

updates the scene based on the agent's actions, running at 10Hz. In the experiment using the generative model as a world generator, we replace the original nuPlan environment. Starting with a 2-second historical scene, it generates and updates the next scene (0.1 seconds ahead) based on the agent's actions. In the experiment using synthetic data to augment the planner, we train the agent with different amounts of synthetic and real data and then evaluate it in the nuPlan closed-loop environment.

D.3. Generation of Novel Scenarios

Nexus can serve as a data engine to automatically generate new scenarios in batches. Specifically, we use the first two seconds of nuPlan raw logs as initial conditions and generate new scenarios through free exploration, conditioned generation, and attacks on the ego vehicle. For attack-based scenario generation, we follow a similar approach to Sec. 3.3 to select attacking vehicles. For goal point selection, we define a sector along the historical trajectory direction of a chosen attack vehicle, with the sector's radius determined by speed and an angle α. The future positions of other vehicles within this sector represent highly probable goal points that could lead to a collision. During generation, we maintain a 4:4:2 ratio among the three types of scenario data to ensure a balanced distribution of scenarios.

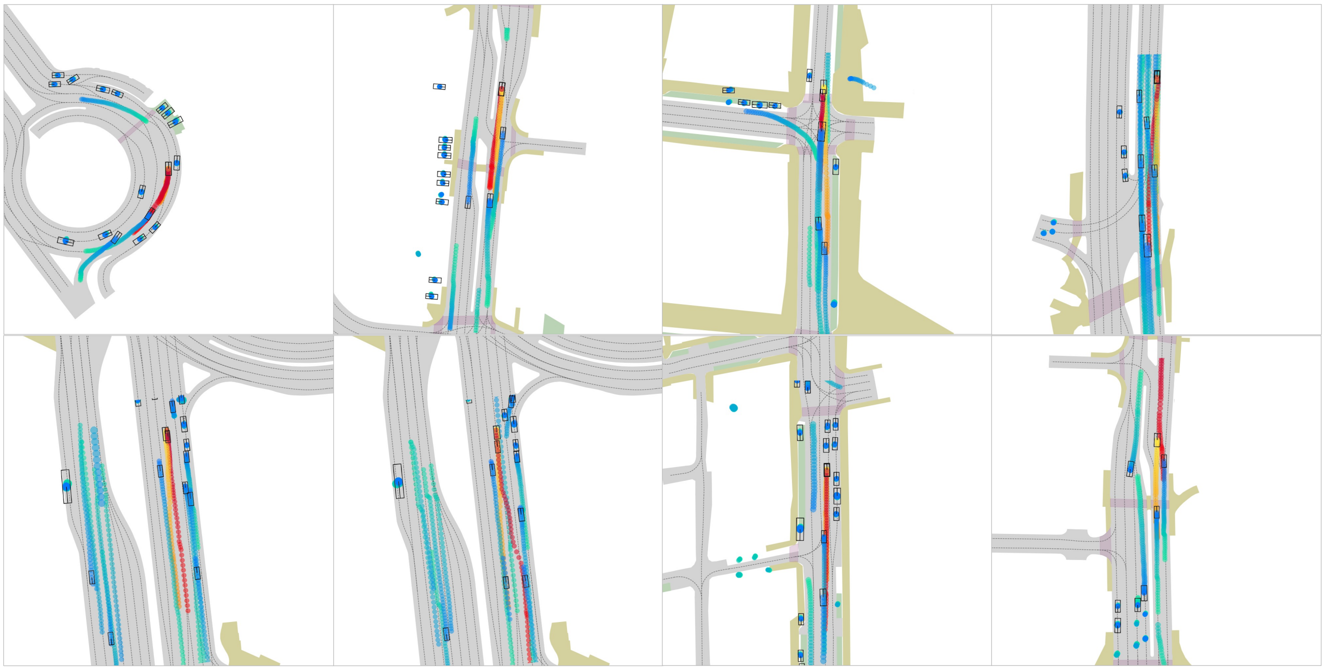

Figure 11. **Applications of Nexus for generating diverse corner cases in autonomous driving.**

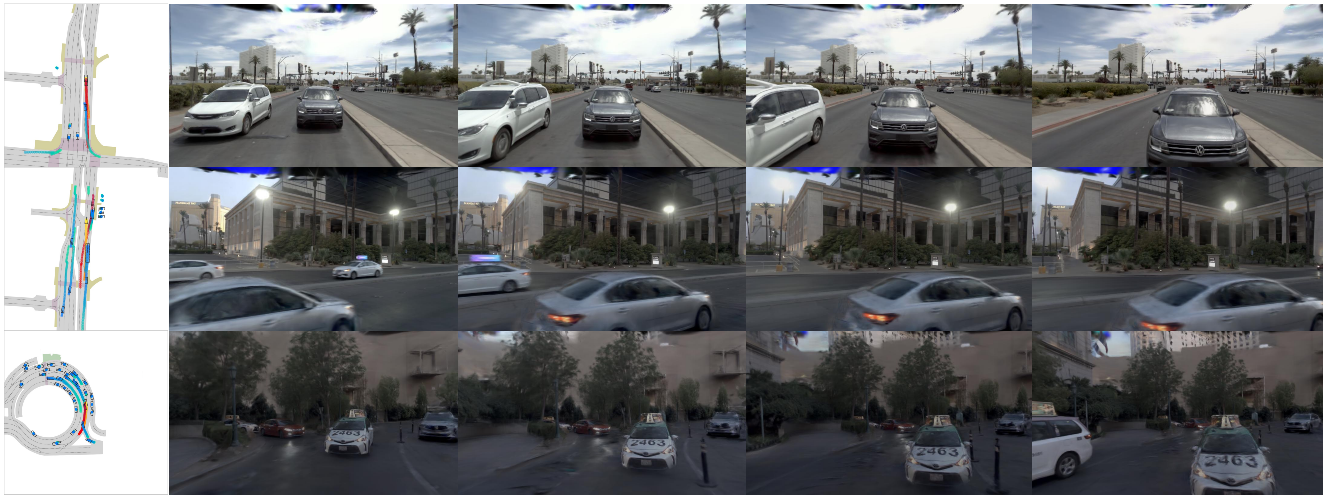

Figure 12. **Leveraging neural radiance fields to provide realistic visual appearances for scenes generated by Nexus.** On the left is the bird's-eye-view layout, and on the right is the rendered scene.

D.4. Qualitive Results

Thanks to the decoupled diffusion structure, Nexus can transition among free exploration, conditioned generation, and diverse corner case synthesis seamlessly, enabling adaptive scene generation. Moreover, leveraging the neural rendering field (NeRF) [55], Nexus transforms generated traffic layouts into photorealistic scenes, enabling controllable visual synthesis.

Free exploration and conditioned generation. Fig. 10 showcases Nexus's versatility in generating driving scenarios. The first two rows depict free exploration, where decoupled noise states enable diverse traffic layouts without explicit condition-

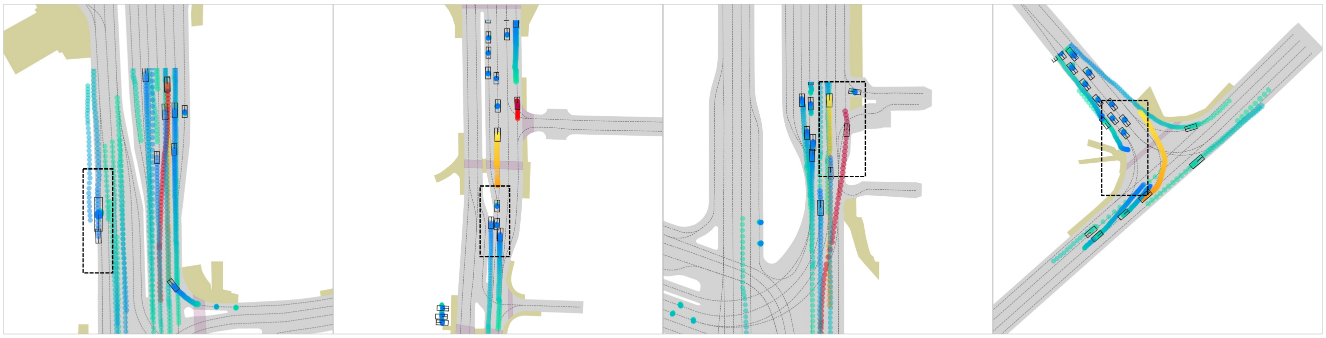

Figure 13. **Failure cases of Nexus.**

ing, capturing complex interactions and behaviors. The last row illustrates conditioned generation, where low-noise target tokens guide scene evolution toward goal states (green flags), enhancing controllability and reactivity while maintaining realism.

Diverse corner case generation. As shown in Fig. 11, Nexus generates diverse corner cases, including abrupt cut-ins, sudden braking, and potential collisions. This strengthens Nexus's utility for training robust autonomous systems.

Controllable visual rendering with NeRF integration. Fig. 12 showcases NeRF-based rendering, converting Nexus-generated layouts into photorealistic scenes. The left panel depicts the bird's-eye view, while the right presents the rendered scene, demonstrating controllable visual synthesis for interactive simulations and closed-loop evaluation.

D.5. Failure Cases

Fig. 13 illustrates Nexus failure cases. The two left cases show incorrect collisions: one between a bus and a sedan and another with overlapping bounding boxes. The two right cases highlight the model's difficulty in making decisions in complex road networks due to limited map information. Future improvements will include adaptive collision awareness and the addition of road boundaries and drivable areas to address these issues.