Decoupled Diffusion Sparks Adaptive Scene Generation

Yunsong Zhou[1,2*] Naisheng Ye[1,3*] William Ljungbergh[4] Tianyu Li[1] Jiazhi Yang[1]
Zetong Yang[5] Hongzi Zhu[2] Christoffer Petersson[4] Hongyang Li[1]

[1] OpenDriveLab at Shanghai AI Lab [2] Shanghai Jiao Tong University [3] Zhejiang University
[4] Zenseact [5] GAC R&D Center

https://opendrivelab.com/Nexus

Abstract

Controllable scene generation could reduce the cost of diverse data collection substantially for autonomous driving. Prior works formulate the traffic layout generation as a predictive progress, either by denoising entire sequences at once or by iteratively predicting the next frame. However, full sequence denoising hinders online reaction, while the latter's short-sighted next-frame prediction lacks precise goal-state guidance. Further, the learned model struggles to generate complex or challenging scenarios due to a large number of safe and ordinary driving behaviors from open datasets. To overcome these, we introduce Nexus, a decoupled scene generation framework that improves reactivity and goal conditioning by simulating both ordinary and challenging scenarios from fine-grained tokens with independent noise states. At the core of the decoupled pipeline is the integration of a partial noise-masking training strategy and a noise-aware schedule that ensures timely environmental updates throughout the denoising process. To complement challenging scenario generation, we collect a dataset consisting of complex corner cases. It covers 540 hours of simulated data, including high-risk interactions such as cut-in, sudden braking, and collision. Nexus achieves superior generation realism while preserving reactivity and goal orientation, with a 40% reduction in displacement error. We further demonstrate that Nexus improves closed-loop planning by 20% through data augmentation and showcase its capability in safety-critical data generation.

1. Introduction

Diversity is crucial for autonomous driving datasets, as data-driven solutions struggle with the scarcity of critical long-tail scenarios. Due to the high cost of collecting rare long-tail driving data, high-fidelity world generators offer a cost-effective alternative for producing diverse scenarios with rare driving behaviors [34]. Besides delivering realistic visuals [14, 28, 58, 66], crafting reasonable and diverse traffic layouts is vital for an adaptive generator to be applicable. This drives two key requirements: 1) Reactivity, which incorporates environmental feedback to model interactions between agents and adjust scene evolution dynamically in response to real-time variations in driving decisions. 2) Goal orientation, which ensures controlled, non-stochastic scene generation guided by predefined future states, allowing the synthesis of realistic safety-critical scenarios with a well-defined outcome.

In this field, diffusion models [6, 59] show promising results in generating realistic scene layouts conditioned on

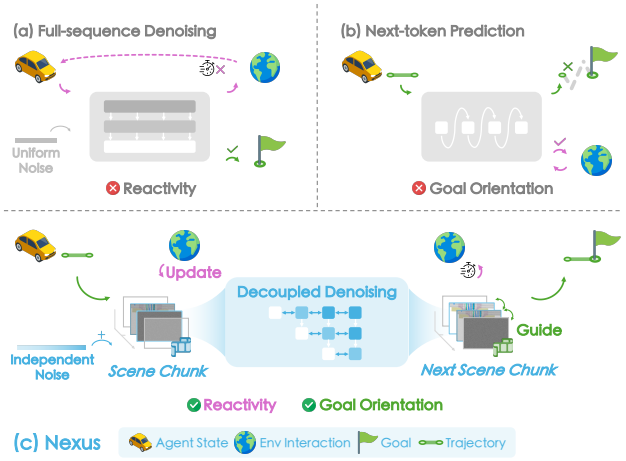

Figure 1. **Nexus** is a noise-decoupled prediction pipeline designed for adaptive driving scene generation, ensuring both timely reaction and goal-directed control. Unlike prior approaches that use **(a)** full-sequence denoising or **(b)** next-token prediction, **(c)** Nexus introduces *independent* yet structured noise states, enabling more controlled and interactive scene generation. It leverages low-noise goals to steer generation while incorporating environmental updates dynamically, which are captured in subsequent denoising.

*Equal contribution.

text prompts [53], protocols [25], and road maps [13]. However, these models struggle to respond to real-time agent interactions due to their rigid full-sequence denoising process, which prevents immediate response to new environmental changes (Fig. 1 (a)). Updates from interactions cannot affect the generation timely, forcing the model to discard previously generated future states and regenerate them entirely. In addition, the rarity of critical scenarios in public datasets [3, 49] limits their ability to generate diverse situations, as these datasets primarily capture routine driving behaviors and lack sufficient risky cases. Alternatively, predictive transformers [17, 39, 42], which excel in responding to environments by continuously rolling out the next frames of the current scenario in Fig. 1 (b). However, they lack awareness of the goal state, as future states are inaccessible to the model during causal generation, making precise control over safety-critical scenarios difficult. Even with global contexts as guidance, precise controls like directing for a collision remain challenging. As a result, existing approaches fail to simultaneously provide both real-time reactivity and goal-directed scene generation, limiting their use in high-fidelity world modeling.

To this end, we introduce **Nexus**, a decoupled predictive model that integrates independent noise across diffusing steps, marrying the reactivity of predictive models with the goal-awareness of diffusion-based approaches. As shown in Fig. 1 (c), Nexus integrates differentiated agent state with decoupled denoising, moving beyond full-sequence scenario generation by adaptively evolving scene chunks over time. Each chunk, a localized subset of the scenario, encodes uncertainty using noise as a soft mask; low-noise regions guide generation, while high-noise tokens allow the reaction to new environmental changes.

Specific designs are proposed to achieve the functionality through the decoupled diffusion model. For goal orientation, our noise-masking training strategy enables Nexus to reconstruct the original sequence from individually corrupted tokens. This facilitates the flexible combination of low-noise goals and high-noise target scenarios during inference, free of adaptation for guided generation. For reactivity, unlike slow autoregressive approaches, our noise-aware scheduling directly adapts token states, ensuring rapid response to environmental changes without unnecessary recomputation. Changes are directly reflected by overwriting the corresponding token states, while the pipelined sampling distributes cost across frames and pops zero-noise tokens at each denoising step for interaction.

To foster a general goal orientation ability for rare or unseen corner cases, we construct the large corpus of safety-critical driving scenarios, **Nexus-Data**. The simulator captures complex behaviors that seldom appear in real-world data through interactions with the physics engine. We generate high-quality training data using the MetaDrive simulator [27], where virtual traffic flows are synthesized via adversarial learning [61] and filtered through automated validity checks to ensure diverse and realistic driving interactions. Nexus-Data comprises 540 hours of simulated driving, representing the largest-scale collection of challenging scenarios, including merging, cut-in, and collision.

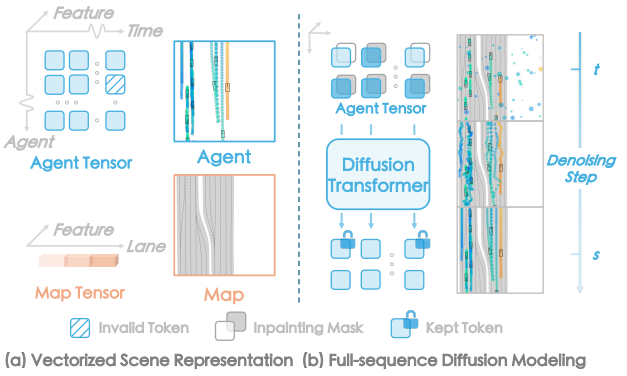

Figure 2. **Preliminary on the scene generation.** **(a)** Current methods encode scenes with tokens for agent and map attributes, formulating scene generation as generating future agent tensors from historical ones conditioned on a global map tensor. **(b)** Diffusion models take the entire sequence as input, using hard masks to fix conditions and enable controllable generation via inpainting, yet fail in a timely reaction.

We summarize our contributions as follows: 1) We introduce Nexus, a decoupled diffusion model that enables adaptive scene generation by learning independent yet structured noise states, improving both goal conditioning and reactive scene updates. 2) We propose noise-masking training, which allows Nexus to integrate goal conditioning with diverse scenario evolution seamlessly. Besides, our noise-aware scheduling mechanism ensures real-time responsiveness by selectively updating only relevant token states. 3) We construct Nexus-Data, a scaled dataset of high-risk driving scenes, enhancing the model's generalization to safety-critical cases. Building on this data and decoupled diffusion, Nexus surpasses Diffusion Policy [6], GUMP [17], and SceneDiffuser [25] in controllability, interactivity, and kinematics, reducing displacement error by 40%.

2. Preliminary

Traffic layout simulation for autonomous driving requires structured representations of both agent behaviors and map features. Recent works frame scene generation as a sequence modeling task, where driving scenarios are represented as structured tokenized states, enabling simultaneous prediction of all agent futures [17, 39, 52].

Vectorized representation of scene generation. As shown in Fig. 2 (a), driving scenarios are encoded as structured token representations, consisting of an *Agent Tensor* for dy-

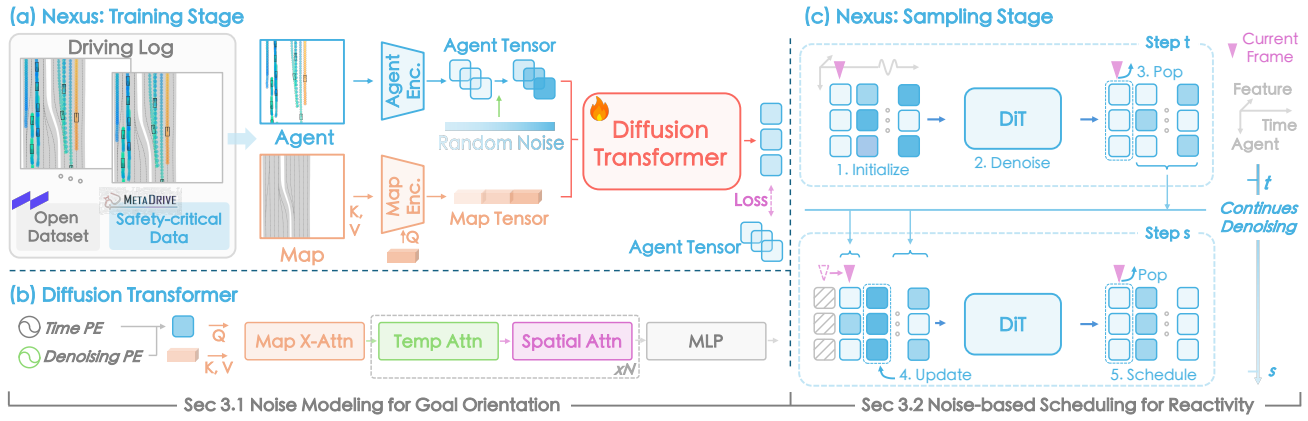

Figure 3. **Framework of Nexus. (a)** Nexus learns from realistic and safety-critical driving logs and encodes agents and maps separately before feeding them into a diffusion transformer. The model is trained to restore sequences from partially masked agent tokens guided by low-noise ones. **(b)** Agent tokens are encoded with time and denoising steps, then interact with the maps and dynamics via attention. **(c)** Tokens with varying noise are scheduled within a chunk for a timely reaction. Each denoising step updates and pops zero-noise tokens, replacing them with next-frame tokens to iteratively generate the scene.

namic entities and a *Map Tensor* for static environment features. We denote the agent tensor as $\mathbf{x} \in \mathbb{R}^{A\times\mathcal{T}\times D}$, where A is the maximum number of agents, $\mathcal{T}$ is the number of physical timesteps, and D is the dimension of agent attributes. The attribute for each agent includes positional coordinates (x, y), heading (sin_α, cos_α), velocities (v_x, v_y), and dimensions (l, w). A valid mask $\mathbf{m} \in \mathbb{B}^{A\times\mathcal{T}}$ is initialized to indicate which agents in the agent tensor $\mathbf{x}$ are valid at each timestep. As for the map information, the map tensor $\mathbf{c} \in \mathbb{R}^{L\times N\times D'}$ is used to represent the lanes' conditions, where L, N, and D' stand for the number of lanes, points per lane, and attributes (coordinates and types), respectively. Based on the vectorized representation, sequential modeling of driving scenes can be expressed as generating the future scene tensor $\mathbf{x} \odot \mathbf{m}_{:,\tau:,:}$ given the current timestep $\tau < \mathcal{T}$, historical scene tensor $\mathbf{x}_{:,:\tau,:}$, and global map tensor $\mathbf{c}$. To simplify the model's learning task, all feature channels are normalized with corresponding means and deviations before concatenating.

Full-sequence diffusion modeling. Diffusion transformers (DiTs) [37] are a class of generative models that generate the agent tensor $\mathbf{x}$ by reversing a stochastic differential process [25, 59]. It can be implemented as stacked transformer blocks ϵ_θ. Let $\mathbf{x}^0 \in \mathcal{X}$ represent a latent feature from the distribution $p(\mathbf{x})$. Training begins with an initial latent state $\mathbf{x}^0$, which undergoes progressive noise injection over timesteps $t \in (0, 1]$ until reaching a Gaussian noise distribution at $\mathbf{x}^1$. The model is optimized by minimizing the mean-square error (MSE):

$$\mathbf{x}^t = \alpha_t \mathbf{x}^0 + \sigma_t \epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{x}^0 \sim p(\mathbf{x}), \quad (1)$$

$$\forall t,\ \min_\theta \mathbb{E}||(\epsilon - \epsilon_\theta(\mathbf{x}^t; \mathbf{c}, t)) \odot \mathbf{m}||_2^2, \quad (2)$$

where α_t, σ_t are scalar functions that describe the magnitude of the data $\mathbf{x}^0$ and the noise ϵ at the denoising step t, θ parameterizes the denoiser ϵ_θ, and $\mathbf{c}$ is the map tensor guiding the denoising process. As illustrated in Fig. 2 (b), all agent tokens are iteratively generated from the standard Gaussian noise with a *uniform* denoising step t during sampling. The full sequence inpainting enables goal-oriented generation, in which the model sets a keep mask $\mathbf{m}_c$ to ensure targets and past tokens remain fixed during sampling:

$$p(\mathbf{x}^s|\mathbf{x}^t) = \mathcal{N}(\mathbf{x}^s|\mu(\mathbf{x}^t, t), \Sigma(\mathbf{x}^t, t)) \odot \bar{\mathbf{m}}_c + \mathbf{x}^t \odot \mathbf{m}_c, \quad (3)$$

where s is the next denoising step, μ and Σ are determined by DiT ϵ_θ. However, fixed-length denoising prevents intermediate state updates, making the model unable to react dynamically to environmental changes during generation.

3. Nexus Framework

Nexus adaptively generates realistic driving scenarios by leveraging decoupled diffusion states for goal-oriented guidance and responsive scheduling. The training stage begins with encoding the agent and the map into tokens with randomly added noise (Fig. 3). For goal orientation, Nexus treats independent noise states as partial masks and uses a diffusion transformer to learn from low-noise guidance via sequence completion (Sec. 3.1). For reactivity, Nexus schedules tokens dynamically based on noise states during sampling, updating active elements continuously while removing generated ones to ensure timely scenario reaction (Sec. 3.2). The training data for conditioned generation in safety-critical scenarios is presented in Sec. 3.3.

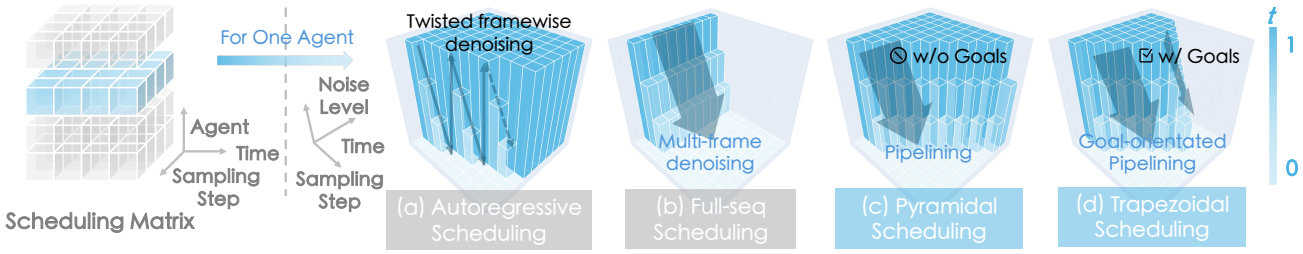

Figure 4. **Diagram of the scheduling strategy.** An agent's noise varies between zero and one across timesteps, determining the balance between stochasticity and goal-driven guidance at each sampling step. **(a)** is hindered by excessive steps per frame. **(b)** reduces costs and follows guidance but can't react to abrupt changes. **(c)** distributes cost by progressively adding tokens to the active chunk only at the start of each step, ensuring smoother transitions and better reactivity. **(d)** enhances future guidance and reduces cost by completing the path from both ends when the goal is fixed. The last two are our options.

3.1. Noise-masking Training for Goal Orientation

Existing approaches [25] train diffusion models with *uniform* noise, relying on hard-masked conditioning to inpaint missing scene components. Yet, sampling requires continuous denoising of a fixed-length sequence to incorporate future guidance. In response to updates, the model discards and regenerates upcoming parts, reducing flexibility. Instead, we propose decoupled diffusion, where *independent* noise levels act as soft masks, enabling Nexus to selectively follow low-noise goal tokens while flexibly reusing or skipping steps for efficient adaptation.

Noising as partial masking. Generative models are essentially various forms of mask modeling [4], which share the practice of occluding a subset of data and training a model to recover unmasked portions. In particular, training full-sequence diffusion can be treated as *noise-axis masking*, namely adding unified noise to the data $\mathbf{x}^0$ over a fixed-length sequence. The sampling process gradually denoises $\mathbf{x}^t$ from Gaussian noise, with the mask being progressively removed. While next-token prediction, which masks each token $\mathbf{x}_{\tau+1:}$ at τ and masks predictions from the past $\mathbf{x}_{:\tau}$, is a form of *time-axis masking*. Masked parts gradually reveal over time with no restrictions on length or composition.

We explore unifying the best of both by leveraging the noise states as partial masks across all dimensions to merge diffusion with next-token prediction. According to [4], any collection of tokens can be viewed as an ordered set with unified indices along all axes without loss of generality. Inspired by [4], we introduce tri-axial mask modeling, where independent noise levels align across agent indices, temporal timesteps, and denoising steps to unify diffusion with next-token prediction. Specifically, we denote $\boldsymbol{x}_{a,\tau}^{k_{a,\tau}}$ as the token of a-th agent $\boldsymbol{x}_{a,\tau}$ within $\mathbf{x}^{k_{a,\tau}}$ at noise level $k_{a,\tau}$ under the forward diffusion process in Eq. (1); $\boldsymbol{x}_{a,\tau}^0$ and $\boldsymbol{x}_{a,\tau}^T$ represent the unnoised token and the pure noise. The noise level matrix $\mathbf{k} = [k_{a,\tau}] \in (0,1]^{A\times\mathcal{T}}$ of the sequence is assigned a random matrix, representing the degrees of Gaussian noise added to corresponding tokens. The optimizing process of the scene generation model can be rewritten as:

$$\forall\, \mathbf{k} \in (0,1]^{A\times\mathcal{T}},\ \min_{\theta} \mathbb{E}||(\epsilon - \epsilon_\theta(g(\mathbf{x}^0, \mathbf{k}); \mathbf{c}, \mathbf{k}))||_2^2, \quad (4)$$

where g represents the function that adds noise to $\mathbf{x}^0$ using matrix $\mathbf{k}$, where each token is masked to varying degrees. The model is learned by completing the full sequence from soft-masked tokens, following information from low-noise tokens when generating other parts. During sampling, setting history and goals to low noise and others to high noise ensures conditional guidance in scene generation.

Scene tokenizing and encoding. Nexus builds upon the diffusion transformer (DiT) [37], employing structured tokenization and encoding to provide a unified representation of driving scenarios. In Fig. 3 (a), the model first extracts vectorized map tensor $\mathbf{c}$ and agent tensor $\mathbf{x}$ from offline-collected driving logs (Sec. 2). Each tensor is channel-normalized and encoded via MLP for unified processing of coordinates, size, speed, *etc*. To ensure stable learning, we initialize a set of learnable queries and use Perceiver IO [22] to encode the map into fixed-length tokens. After adding random noise to the agent tensor, a two-dimensional rotary positional embedding is applied to let the model have a sense of both physical time and denoising steps.

Modeling interactions with multiple attention. The design of Nexus's diffusion transformer focuses on the integration of agent-agent interactions with structured map-based reasoning, ensuring realistic coordination of trajectories and lane-following behaviors (Fig. 3 (b)). Firstly, a map cross-attention queries the map using the agent tensor, aiding in agent-map interactions like lane following and merging. Then, the agent tensor is used to condition a set of temporal and spatial transformer blocks via AdaLN [37]. It captures trajectory continuity and spatial interactions like following and yielding. Besides, the validity $\mathbf{m}$ is used as an attention mask within the transformer denoiser, and invalid and skipped tokens outside the chunk are excluded. The final MLP decodes the agent tokens to compute the reconstruction loss against the ground truth.

3.2. Noise-aware Scheduling for Reactivity

After training, Nexus defines the chunk as a localized subset of the scenario, where varying noise states guide the model to prioritize low-noise cues during denoising. To optimize reactivity, we introduce a noise-aware scheduling strategy that arranges the denoising sequence of scene components for real-time adaption to environmental updates.

Scene generation from next-chunk prediction. Nexus structures each chunk with historical context, future frames to be denoised, and optional goal tokens while adjusting noise levels dynamically at each denoising step. As shown in Fig. 3 (c), the chunk includes frames at time τ, $\tau+1$, and goals, with the darker shade of blue indicating higher noise. After denoising at t, all tokens' noise levels drop. The denoised token at τ is popped out, and a high-noise frame at $\tau+2$ is pushed into the chunk for the next denoising step s. Any environmental changes to the agent tensor can replace the agent state directly and reduce the noise. Thus, the chunk slides temporally as tokens are updated.

We define the scheduling process as a three-dimensional matrix $\mathcal{K} \in [\mathbf{k}]^M$ (Fig. 4 left), where each entry encodes the noise level t for an agent a at physical timestep τ during sampling step m. The sequence is initialized with white noise and the scheduling matrix $\mathcal{K}$ as 1. The noise level of historical frames and the optional goal is set to 0. Nexus selects the noise level $\mathcal{K}_m$ along the sampling step indice m. Fig. 4 (right) shows a fixed agent's noise level changes over sampling steps (height and color), with arrows indicating denoising. Tokens enter the chunk when their noise level changes and exit when it reaches zero, repeating until the entire sequence is generated.

Scheduling for pipelined generation. Tokens with different noise levels are strategically scheduled with matrix $\mathcal{K}$, enabling the model to follow low-noise tokens for denoising without retraining. However, naive autoregressive scheduling is slow due to twisted frame-by-frame denoising, while full-sequence generation, though faster, lacks reactivity to changes. Therefore, we use a pipelined strategy to denoise multiple frames simultaneously, distributing the cost within the chunk. In pyramidal scheduling, the chunk length scales with the number of denoising steps. Each iteration introduces new frame tokens while removing fully denoised ones, allowing continuous scene refinement and output once the chunk is saturated. Trapezoidal scheduling enables bidirectional token updates, where tokens enter and exit from both ends of the chunk, reinforcing goal-conditioned trajectory synthesis. The goal's guidance can propagate to other agents through spatial-temporal attention within the Nexus.

Behavior alignment via classifier guidance. To align generated scenes with realistic driving behavior, we incorporate classifier guidance inspired by dynamic thresholding [45], adjusting noise levels at each step to refine agent interactions. We refine Eq. (3) by applying $f(\mathbf{x}^t, t)$ as a corrective function, adjusting agent trajectories iteratively to enforce behavior constraints at each denoising step. Practically, we separate overlapping agents along their centerline's opposite direction to avoid collisions, smooth trajectories, and pull agents toward the nearest lanes for on-road driving. The formula detail is provided in Appendix C.3.

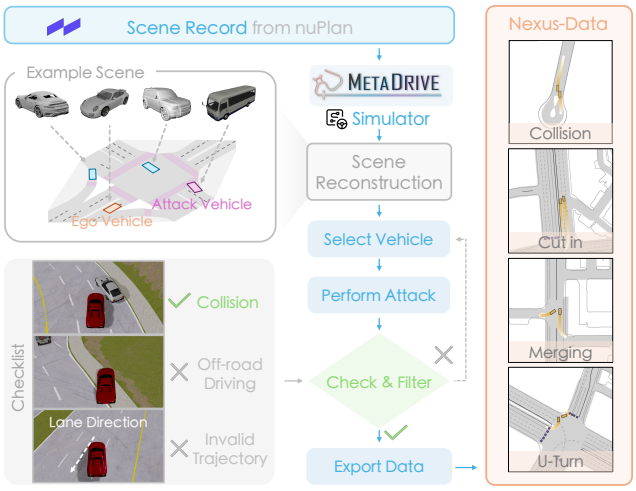

Figure 5. **Nexus-Data construction.** Nexus-Data employs scene records from the nuPlan dataset to reconstruct maps and agents in a simulator to ensure scene realism. It selects a neighbor vehicle to generate attack trajectories by adversarial learning [61] and filters out unrealistic cases.

3.3. Nexus-Data for Generalization in Risky Scenes

Existing datasets predominantly feature safe driving behaviors, limiting exposure to rare corner cases and leading to trajectory discontinuities. To address this, we construct **Nexus-Data**, using MetaDrive [27], to enhance generalization in safety-critical scenarios by changing the motions of the logged agents with adversarial traffic generation [61].

Scene layout construction. We utilize ScenarioNet [26] to transform scenes into a unified description format suitable for simulators, known as *scene records*, logging the agent and map information. As illustrated by the example scene in Fig. 5, loading *scene records*, MetaDrive [27] can reconstruct lanes, roadblocks, and intersections and place corresponding 3D models based on the recorded positions and orientations. By doing so, the digital twin scenario can be faithfully reconstructed in the simulator.

Creation of safety-critical data. As collisions naturally lead to hazardous situations, we use CAT [61] to generate risky scenes initialized from real-world layouts. Specifically, a candidate pool with distance d of the ego vehicle is defined, and we randomly select one as the attack vehicle. Using adversarial learning [61], we generate high-risk interactions by selecting the most collision-prone trajectory for the attack vehicle, forcing the ego vehicle to ex-

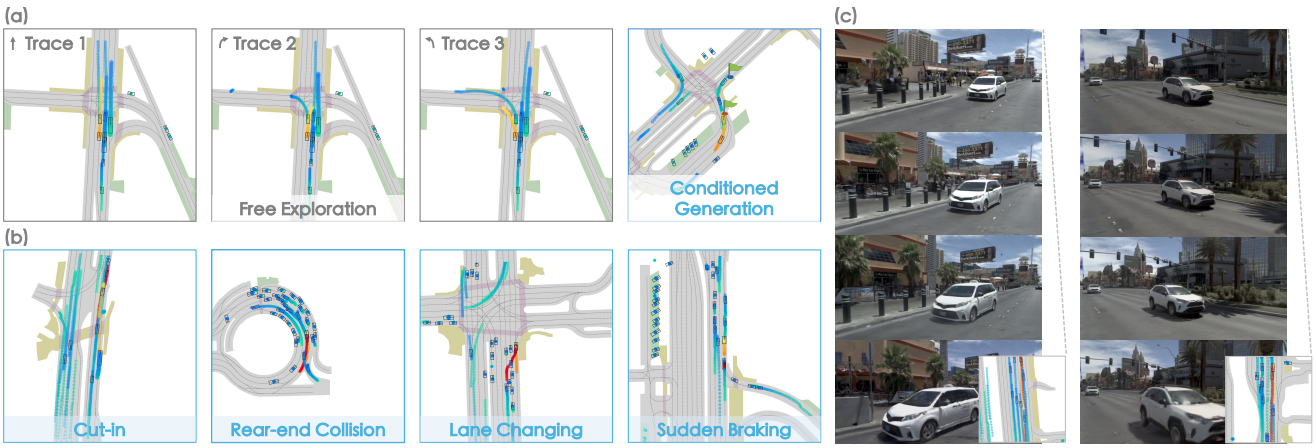

Figure 6. **Visualization of Nexus.** **(a)** Free exploration generates diverse future scenarios from initialized history, while conditioned generation synthesizes scenes based on predefined goal points. **(b)** Setting the attacker's goal as the ego's waypoint enables adversarial scenarios. **(c)** Neural radiance fields provide Nexus-generated scenes with realistic, risky driving visuals. Color transitions represent motion. Ego: yellow to orange. Attacker: red to magenta. Others: green to blue.

ecute avoidance maneuvers under realistic constraints. After K rounds, the attack trajectory is generated. Despite full-stack automation, our statistical study shows that only 36.9% of scenarios result in valid collisions, let alone reasonable ones. Thus, we introduce a checklist to filter out non-collision cases, off-road driving, and invalid trajectories (*e.g.*, in-place U-turns, and lateral shifts), using assert mesh detection and rules. If no valid attacker is found, we continue iterating until the pool is depleted. Eventually, we collect 540 hours of high-quality driving scenarios, covering risky behaviors like sudden braking, crossroad meeting, merging, and sharp turning.

4. Experiments

Setup and protocols. Nexus is trained on nuPlan [3], Waymo [49], and our self-collected Nexus-Data, ensuring exposure to both standard and safety-critical driving scenarios. The performance is evaluated based on controllability, interactivity, and kinematics. We assess trajectory accuracy using average displacement error (ADE), measuring the deviations from ground truth trajectories. The offroad rate measures the proportion of agents that deviate from the centerline beyond a threshold. Collision rate quantifies the safety of generated trajectories, assessing how well Nexus models interactions between agents while avoiding crashes. Metrics related to velocity and angular change describe the trajectory instability of agents' movement.

4.1. Comparison to State-of-the-arts

We compare Nexus with the recently available and reproduced data generation approaches trained on the nuPlan dataset. Conditioned generation and free exploration tasks are used for evaluation. The former probabilistically provides a goal point, while the latter generates an 8-second future freely, with both using the first two seconds as context. Tab. 1 shows that Nexus surpasses all previous methods in controllability (ADE), interactivity (R_{road} and R_{col}), and kinematics (M_{k}). Specifically, Nexus significantly improves ADE by **-4.71** compared to SceneDiffuser [25] while reducing generation time by **-2.55** seconds. It also excels in collision rate (**1.56%**) and trajectory instability, showcasing Nexus's multi-agent interaction modeling. Fig. 6 (a) shows Nexus generating diverse, realistic futures (trace 1-3) from the same initial conditions. Besides, it highlights Nexus's goal adherence and controllability in the conditioned generation for driving simulation.

Table 1. **Generation controllability, interactivity, and kinematics compared to nuPlan experts.** The tasks predict 8-second futures from 2-second history, with or without a goal. ADE: displacement error, R_{road} and R_{col}: off-road and collision rate (%), M_{k}: instability. *Full*: model with Nexus-Data and classifier guidance. gray : main metric. **bold**: best results.

Method	Conditioned Generation				Free Exploration		Time
	ADE ↓	R_{road} ↓	R_{col} ↓	M_{k} ↓	R_{col} ↓	M_{k} ↓	(Sec)
IDM [56]	10.52	9.85	10.17	6.30	12.10	6.02	2.16
D. Policy [6]	7.80	13.9	14.92	12.71	16.88	9.30	6.59
SceneD. [25]	5.99	8.53	11.78	9.64	13.59	6.16	5.34
CTG [63]	3.33	7.79	4.80	6.86	10.67	4.57	-
GUMP [17]	1.93	7.73	7.85	16.18	10.23	14.30	5.59
D. Planner [62]	1.36	8.01	3.47	**2.07**	6.93	**1.73**	-
Nexus	1.28	6.89	1.62	4.63	2.61	3.23	**2.79**
Nexus-*Full*	**1.12**	**6.25**	**1.56**	3.17	**2.10**	2.14	2.93

To boost controllability and corner case generation, we introduce Nexus-*Full*, which incorporates extra Nexus-Data and classifier guidance. This improves scene controllability, reducing ADE by **-0.18** while maintaining interactive realism with minimal time increase from guidance. Nexus-*Full* exhibits strong generalization ability across scenarios (Fig. 6 (b)). It covers safety-critical driving behaviors, including cut-in, collision, lane changing, *etc*.

Table 2. **Comparisons on scheduling strategies.** In addition to task performance, we report the sampling steps, reaction time to changes, and overall time to generate an 8-second scene. `A.R.`: autoregressive. †: updating histories by receiving interactions.

Scheduling	Conditioned Generation				Steps	React Time	Overall Time
	ADE ↓	R_{road} ↓	R_{col} ↓	M_{k} ↓			
A.R.	1.48	9.95	1.98	**4.58**	512	4.96	79.36
Full-sequence	1.28	9.63	1.62	4.63	**32**	4.96	**4.96**
Pyramidal	1.53	9.85	1.80	4.74	48	**0.16**	7.68
Trapezoidal	1.39	9.70	1.92	4.63	40	**0.16**	6.20
Trapezoidal †	**1.17**	**9.54**	**1.71**	4.89	40	**0.16**	6.20

Table 3. **Ablation on designs in Nexus.** `P.E.`: positional embedding with physical and denoising time. All proposed designs contribute to the final performance.

Method	Conditioned Generation				Free Exploration	
	ADE ↓	R_{road} ↓	R_{col} ↓	M_{k} ↓	R_{col} ↓	M_{k} ↓
Baseline	7.53	9.74	13.52	11.47	15.79	6.02
+ Noise Masking	3.42	9.16	3.01	8.19	3.48	5.23
+ P.E.	1.44	8.17	2.52	6.20	3.17	3.42
+ Nexus-Data	1.32	7.53	1.92	4.87	2.76	3.35
+ Classifier Guidance	**1.25**	**6.73**	**1.47**	**4.02**	**1.38**	**3.28**

Beyond nuPlan, we also evaluate the unconditional generation of Nexus on the Waymo Open Sim Agent *val* set, scoring **61.9** on the composite metric without any post-processing, outperforming the state-of-the-art competitor SceneDiffuser [25] (55.8).

4.2. Ablation Study

Comparison on scheduling strategies. The ablation is conducted by training each variant of our model on nuPlan with 30K steps. Tab. 2 compares the impact of different scheduling strategies of Nexus on performance and speed. Traditional scheduling strategies introduce significant latency, requiring **4.96** seconds to respond to environmental changes, making them impractical for real-time scene adaptation. Notably, the pyramidal and trapezoidal scheduling strategies respond to changes during each denoising step, reducing response time by **-4.80** seconds without performance loss. When Nexus updates the historical state in real-time using feedback from agents, ADE improves by **+0.11** compared to full-sequence scheduling. Further, by applying step skipping [48], the sampling steps can be reduced to 18, generating 8-second future scenes in just **2.79** seconds.

Ablation on each component. To validate each component's effectiveness, we gradually introduce our proposed components and conditions, starting with a Diffusion Policy baseline [6]. As Tab. 3 shows, noise-masking training with independent noise states reduces ADE by **-4.11**, enabling the model to follow low-noise cues via sequence reconstruction. Likewise, encoding physical and denoising time boosts performance by **-1.98**, aligning with our independent noise design. Integrating Nexus-Data improves ADE by **-0.12**, enhancing controllability in diverse scenarios. Lastly, adding classifier guidance from human behavior constraints boosts collision (**-0.45%**) and kinematic metrics (**-0.85**). Fig. 7 shows how Nexus-Data enhances corner case generation. Besides, adding constraints improves safe distancing, on-road driving, and trajectory stability.

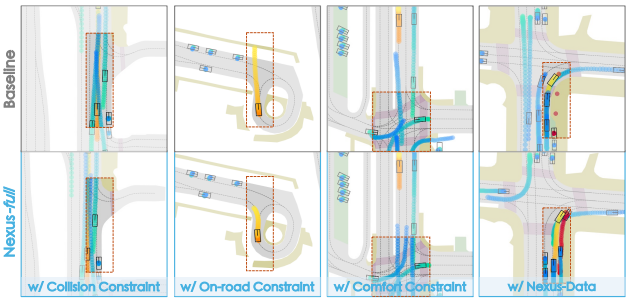

Figure 7. **Ablation on classifier guidance and Nexus-Data.** The designs improve collision avoidance, on-road driving, stability, and corner case generation. Yellow is the ego agent, blue is the others, and red is the attacker.

4.3. Discussion on Application

World generator for closed-loop driving. Nexus enables closed-loop scene generation, acting as an interactive environment for autonomous agents. The agent uses generated scenes for planning, while Nexus updates them in real-time based on the agent's actions. To assess the realism of the scene generator, we set up an evaluation on the nuPlan closed-loop Val14 set (Tab. 4). The generator predicts the next scenario using history and agent actions. A baseline agent, Diffusion Planner [62], predicts future waypoints based on the generated scenes, with more realistic scenes yielding higher metrics in the nuPlan closed-loop evaluation. Nexus surpasses all baselines in reactive closed-loop evaluation by **+15.8**, highlighting its ability to generate interactive, realistic driving environments for closed-loop planning and policy learning.

Data augmentation via synthetic data. Nexus can generate diverse future scenarios from fixed history, making it a possible data engine for augmenting planning model training. In a preliminary experiment, we sample 3 hours of nuPlan logs, generate synthetic data with Nexus, and train a lightweight planner [6] on real and augmented data. Tab. 5 indicates that blending generated data with real-world data improves reactive closed-loop score to **57.86**, a **+20%** improvement over real-data-only models. This demonstrates that high-quality synthetic data can enhance planner robustness and generalization. Models trained on small real datasets tend to slow down, raising collision scores while worsening other metrics. We also find that limited synthetic data (3×) degrades performance, likely due to scene noise

Table 4. **Evaluation of a generation model as a world generator.** The scene generator serves as the interactive world model response to the baseline planner's actions, with nuPlan closed-loop metrics reflecting its realism. `Oracle`: ground truth environment of nuPlan evaluation. S_{col}: collision score, S_{p}: progress score.

Method	Reactive Eval.			Non-reactive Eval.		
	Score ↑	S_{col} ↑	S_{p} ↑	Score ↑	S_{col} ↑	S_{p} ↑
Oracle	82.8	89.5	97.0	89.2	91.6	100.0
Diffusion Policy [6]	61.6 (-21.2)	81.9	90.2	47.2 (-42.0)	67.0	89.8
SceneDiffuser [25]	57.2 (-25.6)	74.7	91.6	50.1 (-39.1)	66.3	89.5
Nexus	**73.0** (-9.8)	**84.9**	**95.0**	**68.1** (-21.1)	**77.7**	**96.6**

Table 5. **Comparison involving data augmentation using synthetic data.** Nexus serves as a data engine, expanding sampled scenes to train the planner [6] at varying scales. The nuPlan closed-loop evaluation demonstrates the performance gains from data augmentation. `Synth.`: synthetic.

Training Data	Reactive Eval.			Non-reactive Eval.		
	Score ↑	S_{col} ↑	S_{p} ↑	Score ↑	S_{col} ↑	S_{p} ↑
Real Data	48.11	**80.81**	83.41	46.39	72.54	88.80
w/ 3× Synth. Data	46.61	75.78	86.25	43.69	66.93	89.15
w/ 30× Synth. Data	56.46	78.92	**92.55**	53.39	69.86	**94.55**
w/ 60× Synth. Data	**57.86**	79.50	91.96	**56.42**	**76.49**	93.32

affecting planner learning. A 30× increase brings significant gains, but further expansion leads to model saturation. This shows that sufficient data scaling benefits the driving model, further validating Nexus as a reliable synthetic data generator for driving model training.

Attempts to visual world models. Nexus is not designed for visual synthesis. Yet, high-quality traffic layout generation opens new possibilities for controllable visual scene rendering, which is essential for realistic interactive driving simulations and closed-loop testing. We have an initial attempt to render nuPlan driving scenes using neural radiance fields. We modify an open-source autonomous driving neural reconstruction method [55] to support Nexus-generated scenes, allowing control over agent positions and behaviors through novel layouts. This shows a promising result of action-conditioned video generation for safety-critical scenarios in Fig. 6 (c). This application is impossible with existing world models, which are only trained and conditioned on a given static real-world dataset that lacks records of dangerous driving behaviors. This brings new opportunities for closed-loop data generation capabilities.

5. Related Work

Diffusion models for conditioned generation. Diffusion models have advanced policy generation [6, 23, 31, 36] and scene synthesis [19, 29, 44, 66]. Some works explore their use in ego-motion planning [21, 24, 30, 50], while Diffusion-ES [59] integrates evolutionary search for optimizing non-differentiable trajectories. Diffusion Planner [62] jointly predicts ego and surrounding vehicle trajectories, merging motion prediction with closed-loop planning. Other efforts focus on full-scene generation as a world generator [7] or controllable traffic simulation via user-defined trajectories [64]. SceneDiffuser [25] refines diffusion denoising for efficient simulation with hard constraints and LLM-driven scene generation. However, these approaches rely on static dataset layouts, limiting their reactivity to dynamically evolving traffic conditions and safety-critical scenario synthesis. Nexus overcomes this by integrating decoupled diffusion with noise-aware scheduling for real-time reaction. Despite amortized optimizations, existing diffusion models struggle with goal-oriented planning due to uniform noise treatment, which hinders precise control over agent trajectories. Additionally, their fixed-sequence denoising process limits timely reactivity to environmental changes.

Transformers for reactive generation. Closed-loop agent state prediction requires reaction mechanisms for simulation dynamics and realism [13, 15, 40, 43, 53]. Decisions are influenced by the historical and current system state, including other agents' actions. MotionLM [46] and TrajEnglish [39] use autoregressive GPT-like models, while GUMP [17] improves response speed with key-value pair tokenization and state-space quantization, enabling flexible handling of agent disappearance and emergence. Unlike autoregressive models, which rely on discrete updates, Nexus enables real-time scene adaptation by integrating noise-aware scheduling with goal-conditioned diffusion.

Conventional traffic simulation. Rule-based traffic simulation methods have demonstrated success in controlled settings [9, 11, 51, 56] , but their rigid structure prevents them from adapting to novel interactions or emergent behaviors. While imitation learning methods [5, 8, 18] excel in behavior cloning, they lack adaptability to unseen scenarios, as their reliance on predefined rules leads to brittle, unsafe outputs in edge cases [20, 32, 57, 65]. In contrast, Nexus learns flexible, data-driven representations that capture diverse driving behaviors, generalizing beyond predefined rule sets and enabling controllable scene generation.

6. Conclusion

We introduce Nexus, a decoupled diffusion model that generates diverse driving scenarios using independent noise states for real-time reaction and goal-oriented control. This enables dynamic evolution while maintaining precise trajectory conditioning. We curate Nexus-Data, a large dataset of safety-critical scenarios, helping Nexus better model edge cases and high-risk interactions. By integrating noise-masking training and noise-aware scheduling, Nexus achieves goal-driven scene synthesis with improved fidelity and diversity, improving autonomous driving simulations.

Acknowledgements

We extend our gratitude to Shenyuan Gao, Li Chen, Chonghao Sima, Carl Lindström, Adam Tonderski, and the rest of the members from OpenDriveLab and Zenseact for their profound discussions and supportive help in writing. This work is supported by the National Key Research and Development Program of China (2024YFE0210700) and NSFC (62206172).

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 13

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 14

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 2, 6, 13, 14, 15

[4] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024. 4

[5] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023. 8

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2023. 1, 2, 6, 7, 8, 17

[7] Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *European Conference on Computer Vision*, pages 57–74. Springer, 2025. 8

[8] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2022. 8

[9] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. *arXiv preprint arXiv:2306.07962*, 2023. 8

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 13

[11] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018. 8

[12] Francesca Favaro, Laura Fraade-Blanar, Scott Schnelle, Trent Victor, Mauricio Peña, Johan Engstrom, John Scanlon, Kris Kusano, and Dan Smith. Building a credible case for safety: Waymo's approach for the determination of absence of unreasonable risk. *arXiv preprint arXiv:2306.01917*, 2023. 13

[13] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575. IEEE, 2023. 2, 8

[14] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *arXiv preprint arXiv:2405.17398*, 2024. 1

[15] Zhiming Guo, Xing Gao, Jianlan Zhou, Xinyu Cai, and Botian Shi. Scenedm: Scene-level multi-agent trajectory generation with consistent diffusion models. *arXiv preprint arXiv:2311.15736*, 2023. 8

[16] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 14, 17

[17] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. In *European Conference on Computer Vision*, pages 386–404. Springer, 2025. 2, 6, 8, 17

[18] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 8

[19] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023. 8

[20] Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3903–3913, 2023. 8

[21] Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. *arXiv preprint arXiv:2410.05582*, 2024. 8

[22] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 4, 14

[23] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 8

[24] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9644–9653, 2023. 8

[25] Chiyu Max Jiang, Yijing Bai, Andre Cornman, Christopher Davis, Xiukun Huang, Hong Jeon, Sakshum Kulshrestha, John Lambert, Shuangyu Li, Xuanyu Zhou, et al. Scenediffuser: Efficient and controllable driving simulation initialization and rollout. *arXiv preprint arXiv:2412.12129*, 2024. 2, 3, 4, 6, 7, 8, 17

[26] Quanyi Li, Zhenghao Peng, Lan Feng, Chenda Duan, Wenjie Mo, Bolei Zhou, et al. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *arXiv preprint arXiv:2306.12241*, 2023. 5

[27] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022. 2, 5, 13, 14

[28] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023. 1

[29] Zuoyue Li, Zhenqiang Li, Zhaopeng Cui, Marc Pollefeys, and Martin R Oswald. Sat2scene: 3d urban scene generation from satellite images with diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7150, 2024. 8

[30] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024. 8

[31] Tenglong Liu, Jianxiong Li, Yinan Zheng, Haoyi Niu, Yixing Lan, Xin Xu, and Xianyuan Zhan. Skill expansion and composition in parameter space. *arXiv preprint arXiv:2502.05932*, 2025. 8

[32] William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *European Conference on Computer Vision*, pages 161–177. Springer, 2024. 8

[33] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 2018. 14

[34] Enhui Ma, Lijun Zhou, Tao Tang, Zhan Zhang, Dong Han, Junpeng Jiang, Kun Zhan, Peng Jia, Xianpeng Lang, Haiyang Sun, et al. Unleashing generalization of end-to-end autonomous driving with controllable long video generation. *arXiv preprint arXiv:2406.01349*, 2024. 1

[35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 13

[36] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023. 8

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3, 4

[38] Zhenghao Peng, Wenjie Mo, Chenda Duan, Quanyi Li, and Bolei Zhou. Reward-free policy learning through active human involvement. 2022. 13

[39] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 8

[40] Cheng Qian, Di Xiu, and Minghao Tian. The 2nd place solution for 2023 waymo open sim agents challenge. *arXiv preprint arXiv:2306.15914*, 2023. 8

[41] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. NuScenes-QA: A multi-modal visual question answering benchmark for autonomous driving scenario. *arXiv preprint arXiv:2305.14836*, 2023. 13, 14

[42] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[43] Luke Rowe, Roger Girgis, Anthony Gosselin, Bruno Carrez, Florian Golemo, Felix Heide, Liam Paull, and Christopher Pal. CtRL-sim: Reactive and controllable driving agents with offline reinforcement learning. In *8th Annual Conference on Robot Learning*, 2024. 8

[44] Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *CVPR*, 2025. 8

[45] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 5, 17

[46] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *ICCV*, 2023. 8

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 17

[48] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 7

[49] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 6, 14

[50] Qiao Sun, Shiduo Zhang, Danjiao Ma, Jingzhe Shi, Derun Li, Simian Luo, Yu Wang, Ningyi Xu, Guangzhi Cao, and Hang Zhao. Large trajectory models are scalable motion predictors and planners. *arXiv preprint arXiv:2310.19620*, 2023. 8

[51] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21371–21382, 2022. 8

[52] Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Krähenbühl, and Marco Pavone. Promptable closed-loop traffic simulation. *arXiv preprint arXiv:2409.05863*, 2024. 2

[53] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Krähenbühl. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*, 2023. 2, 8

[54] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety Administration, 2018. 13

[55] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 8, 19

[56] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 6, 8, 17

[57] Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 897–904. IEEE, 2022. 8

[58] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. *arXiv preprint arXiv:2311.16813*, 2023. 1

[59] Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *arXiv preprint arXiv:2402.06559*, 2024. 1, 3, 8

[60] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 13

[61] Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *7th Annual Conference on Robot Learning*, 2023. 2, 5, 13, 15

[62] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025. 6, 7, 8

[63] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, pages 144–177. PMLR, 2023. 6

[64] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566. IEEE, 2023. 8

[65] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monoef: Extrinsic parameter free monocular 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10114–10128, 2021. 8

[66] Yunsong Zhou, Michael Simon, Zhenghao Mark Peng, Sicheng Mo, Hongzi Zhu, Minyi Guo, and Bolei Zhou. Simgen: Simulator-conditioned driving scene generation. *Advances in Neural Information Processing Systems*, 37:48838–48874, 2024. 1, 8