RareOcc: Controllable 4D Occupancy and LiDAR Generation for Long-Tail Driving Scenes

Arthur Jakobsson[1,2], Yu-Rou Tuan[1,2], Leron Julian[1], Shawn Hunt[1], Kenta Suzuki[1], Shinya Tanaka[1], Mahdi Bandegi[1], Jay Gloomis[1], Hironobu Fujiyoshi[3], Kris Kitani[2], and Jeffrey Ichnowski[2]

[1] DENSO INTERNATIONAL AMERICA, INC, Pittsburgh Innovation Lab,
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[3] DENSO CORPORATION

Abstract. Autonomous vehicle behavior in emergency and rare situations is a well documented challenge, with some of the most concerning instances being ambulance obstruction. Yet the sensor data needed to study these long-tail events at scale is exactly what their rarity makes scarce. We propose RareOcc (ROCC), a system that generates realistic 4D occupancy and LiDAR for customizable long-tail situations and documented vehicular crashes; to our knowledge RareOcc is the first pipeline to jointly support crash-grounded 4D generation, editable BEV authoring, occupancy generation, and LiDAR generation. ROCC uses a bird's-eye-view (BEV) layout as a shared intermediary that three interchangeable methods write to (CIREN-based crash recreation, LLM-based behavior specification, and CompoSIA-based real-scene editing) and lifts it to occupancy with an entity-grounded generator that places each agent from its specified box and replaces the diffusion stage of prior occupancy-centric pipelines. Evaluated on the nuPlan-Occ mini validation dataset, our model outperforms the state-of-the-art UniScene-v2 generator by 22.6 points in occupancy mIoU, and materializes requested scene edits across all seven foreground classes, rare categories included. Generated examples from all three creation routes are browsable at `https://rareocc-anon.github.io/`.

Keywords: Autonomous Driving · Safety-Critical Scenarios · 4D Occupancy Generation

1 Introduction

"Waymo is frequently now blocking our fire stations from access... Their default is to freeze," reports Deputy Chief of Operations Patrick Rabbitt of the San Francisco Fire Department [40]. His account is not isolated. Since April 2025 San Francisco firefighters have filed at least 31 internal reports of robotaxis obstructing emergency operations, including four cases in which crews met stalled vehicles while responding to life-threatening calls [5]. These are emblematic of a broader pattern: autonomous vehicles faltering in rare, dangerous situations, coming to a standstill rather than resolving the conflict, or misreading a scene

whose structure they have never seen. Whether such a failure originates in perception or in the decision layer above it, both are trained and evaluated on sensor data, and sensor data of precisely these situations is what does not exist. One incident poses the problem this paper addresses most sharply: an active post-crash scene with an injured pedestrian on the ground [29] is the configuration no driving log contains, and what no log records, no reconstruction-based simulator can re-render. For autonomous vehicle perception and decision making, such long-tail events are hardest to collect, hardest to reproduce, and therefore hardest to train and evaluate against. Data collection would be a challenging instrument for this regime. Accumulated nominal driving data collection by construction contains little of the tail: the rarer a situation, the less data exists for it. Generation decreases that dependence. If a rare situation can be specified and synthesized at sensor level, the long tail of the training distribution can be extended deliberately rather than waited for, and generated occupancy and LiDAR of this kind already improve downstream perception and decision making [20]. We therefore seek an end-to-end pipeline that produces the LiDAR and occupancy of situations no fleet can be asked to record. Prior work approaches the parts but not the whole. Occupancy has emerged as a central representation for perception, for 4D forecasting of how a scene will evolve [30], and as a grounding signal for reward and closed-loop evaluation, while a scenario-generation line reconstructs driving situations from real CIREN [27, 28] crash records [26], stopping at abstract layouts or trajectories rather than sensor output and full 3D representations. Our method bridges these threads, using a bird's-eye-view (BEV) layout as a shared intermediary connecting rare-situation *description* to sensor-level *generation*. We generate occupancy and LiDAR here; photorealistic multi-camera output is left to future work, but prior generators show it is achievable with enough compute, and our occupancy interface is designed to support it. Our contributions are:

- **RareOcc (ROCC)**, an end-to-end system for creating 3D occupancy and LiDAR of long-tail and safety-critical driving scenes, including real recorded crashes.
- An **improved BEV-to-occupancy stage** that preserves scene layout and agents where prior generators degrade under rare-class layout edits.
- A **unified editable-BEV interface** that integrates three interchangeable scene-creation methods: CIREN-based crash recreation, LLM-based behavior specification, and CompoSIA-based real-scene editing.
- An **evaluation** of generation fidelity, controllability, and scenario integrity on nominal and created scenes.

2 Related Work

Controllable driving scene generation. Camera-centric methods [11, 12, 39, 55] render multi-view video from BEV layouts and 3D boxes, and driving world models [13, 22] produce action-conditioned rollouts. Occupancy-centric methods instead treat 3D occupancy as the primary generative substrate [4, 30, 36, 38, 46, 56]: UniScene [20, 21], AnyScene [50], and DrivingSphere [42] decode BEV

layouts into occupancy, multi-view video, and LiDAR, with demonstrated downstream benefit [20]. Dedicated LiDAR generators [16, 31, 59, 60] skip the occupancy stage, and SLEDGE [8] generates the lane graphs and agent layouts themselves – a natural upstream source for our interface. All are trained on nominal driving distributions, and none is designed for the layout edits a simulator needs (Sec. 4). ROCC keeps the occupancy-centric factorization but grounds layouts in real crash records and OOD scenarios, and improves BEV-to-occupancy so that layout and agent fidelity survives in long-tail, safety-critical scenes.

Entity-centric latent scene representations. Our occupancy stage builds on the LatentWorld of Park *et al.* [30], which represents a scene as a sparse set of *grounded latents* – one per foreground actor, many for background – each decoded into semantic 3D Gaussians [17] and splatted to an occupancy grid; related object-centric formulations appear in [4]. We inherit this representation and its autoencoder, and differ in where the layout comes from: LatentWorld *generates* the layout with a dedicated diffusion stage, whereas ROCC *receives* an authored BEV and learns the mapping from a 2D raster to grounded latents (Sec. 3.4), porting the representation to nuPlan-Occ [21].

Simulation and re-simulation. Driving simulators [9,15,24,25] offer scripted scenarios and free labels, but render camera and LiDAR from a hand-authored asset library, so appearance diversity is bounded and a sim-to-real gap persists. Log re-simulation [35, 43, 44, 47, 58] inherits real sensor statistics and supports novel views and actor edits, but stays anchored to the logged scene.

Safety-critical and crash scenario generation. Language and LLM methods [34,51,57] write out multi-agent trajectories or simulator scripts, and adversarial methods [32, 33] search for collisions and near-misses. A crash-grounded line rebuilds scenes from real NHTSA crash reports and sketches [23, 26–28] or generates crash videos [14,52]; DeepAccident [37] renders simulated accidents to multi-view camera and LiDAR, but from hand-authored assets. Dashcam accident anticipation [2, 7, 10, 48] studies crashes only as 2D video, detecting rather than creating them. These methods give the rare behavior we want, but as trajectories, scripts, or pixels – not the 3D occupancy and LiDAR a self-driving stack actually reads. ROCC reuses SAFE's [26] crash extraction as one of its three BEV front-ends and carries it all the way to sensor data (Sec. 3.2).

Long-tail and out-of-distribution perception. OOD occupancy prediction and anomaly injection [53, 54], and robustness under corrupted or missing sensors [18,41], motivate both the need for controllable long-tail data and the question of whether generated scenes are genuinely out-of-distribution yet realistic.

3 Method

We introduce ROCC, an end-to-end pipeline from BEV composition to occupancy and LiDAR (Fig. 1). Two prior methods, SAFE [26] and CompoSIA [49], are integrated behind a unified editable-BEV interface, alongside a new LLM-based creation of BEV scenes from free-text description. A grounded-latent gen-

erator (Sec. 3.4) lifts each BEV, together with its agent boxes, to a 3D occupancy grid, improving layout and agent fidelity over UniScene-v2's occupancy generation, and LiDAR is rendered from that occupancy with UniScene-v2's occupancy-to-LiDAR model (Sec. 3.5).

3.1 The Unified Editable-BEV Interface

ROCC routes every scene through a single editable bird's-eye-view (BEV) tensor, $\mathcal{B}$, that every creation method writes and every synthesis stage reads. Fixing this shared interface is what makes the creation methods interchangeable: the occupancy and LiDAR generators never see how a scene was created, only $\mathcal{B}$.

Representation (adopted from prior work). $\mathcal{B} \in \{0,1\}^{18\times400\times400}$ is an ego-centric binary raster at 0.25 m/px over a $\pm$50 m range, in the channel layout and class taxonomy of **nuPlan** [6] exactly as used by UniScene-v2's [21] occupancy benchmark, so a BEV from any creation method is format-identical to the data the occupancy model was trained on. The 18 channels group into eight static map layers (0–7), seven dynamic agent classes (8–14), and three divider layers (15–17); seven auxiliary channels carry height and elevation, and the BEV encoder reads all 25 (Sec. 3.4); the binary definition above covers the 18 class channels only. A class mask fixes an agent's ground-plane footprint but not its vertical extent, so each scene also carries one 3D box per agent, giving its center (x, y, z), extents (L, W, H), yaw, and class. Every foreground latent is instantiated from one box and conditioned on its extents (Sec. 3.4), so the third dimension is *specified* by the creation method rather than inferred from the raster, and the box doubles as the control surface for stylization: a tall, wide box decodes toward an ambulance-like vehicle rather than a car.

Editability. Because the interface is a small stack of class masks plus a list of boxes, a scene can be inspected and edited directly. Map edits, such as redrawing the drivable region, are local edits to a single channel; agent edits, such as adding a cyclist or moving a vehicle, are edits to that agent's box, from which the dynamic channels are re-rasterized, keeping raster and boxes consistent by construction (the protocol of App. E). Three writers target this interface: crash-grounded extraction (Sec. 3.2) and the free-text and real-clip-editing routes (Sec. 3.3).

3.2 Crash-Grounded Scene Extraction (CIREN → SAFE → BEV)

The first creation route recreates real recorded crashes. Each case in the NHTSA CIREN [27, 28] database pairs a hand-drawn collision sketch with a natural-language summary, and we compile the two into an editable BEV scene. We reuse SAFE's [26] multimodal extraction without modification: a vision-language model reads the sketch and summary and emits a structured scenario comprising a meta description (road type, actors, maneuvers) and a DSL, which we render into $\mathcal{B}$ as in Sec. 3.1. A crash sketch is abstract and has no ground-truth map to query, so the static and divider layers are synthesized procedurally from the DSL road type; the DSL actors are placed on this schematic map and rasterized under the same metric-to-grid convention as real data, with planar

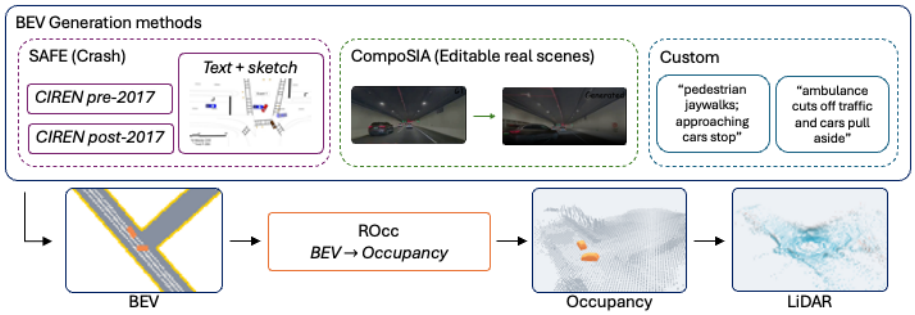

Fig. 1: ROCC overview. Three interchangeable creation routes write the same editable BEV interface $\mathcal{B}$ (Sec. 3.1): crash-grounded extraction from CIREN sketches and summaries via SAFE (Sec. 3.2), free-text behavior specification, and CompoSIA-based editing of real clips (Sec. 3.3). We then run our grounded latent occupancy generation to create an occupancy grid (Sec. 3.4), which UniScene-v2's frozen occupancy-to-LiDAR model renders into ego LiDAR sweeps (Sec. 3.5).

size and height drawn from class-conditioned priors indexed by vehicle type, so synthesized scenes depart from real data in content, not in form. We serve this extraction with a self-hosted Qwen3-VL [1] model in place of the proprietary API used by the original SAFE pipeline; the same model serves the text-driven creation of Sec. 3.3. We augment the extracted scenario with a deterministic conflict descriptor recording the crash structure, and run a short kinematic rollout that stages the collision and latches the impact frame. Because a crash involves several vehicles, we rasterize the rollout once per vehicle taken as ego, so one case yields several ego-centric clips observing the same event from complementary viewpoints. We will release this corpus of per-ego BEV and occupancy clips; App. B provides details and Sec. 4.1 its scale.

3.3 Editable and Custom OOD Scenes

Beyond recorded crashes, ROCC creates novel out-of-distribution scenes through two further routes into the same interface.

Free-text behavior specification. The first route synthesizes a scene from free text through a hierarchical procedure separating *what the user intends* from *how each agent realizes it*: a text-only call parses the description into a structured scene schema, then a second stage assigns each agent a compact, scenario-agnostic goal (a target speed, a lateral action, an optional stopping condition, and a turn). Because these primitives compose, one vocabulary expresses behaviors as varied as yielding to an emergency vehicle or halting for a crossing animal, with no scenario-specific logic. A universal controller advances the scene under a shared car-following and no-overlap model and rasterizes every frame into $\mathcal{B}$ by class. This route reuses SAFE's road geometry and rollout primitives and adds the intent-to-goal planning; App. C gives the prompt structure, the goal vocabulary, and the two planning modes.

Editing real scenes. The second route edits real scenes with CompoSIA [49], which manipulates a real driving clip through disentangled control of scene structure, agent appearance, and dynamics, and emits per-frame 3D bounding boxes. A thin bridge renders that stream into $\mathcal{B}$: boxes are aligned from CompoSIA's convention to nuPlan's and rasterized into the dynamic channels, classes are remapped to the nuPlan taxonomy (unrecognized ones assigned to the generic-object channel rather than dropped), and the static and divider layers are borrowed from the real frame. The conversion is lossless: on real tokens the bridge reproduces the reference BEV at a mean intersection-over-union of 1.0, so an edited clip (displacing an agent, inserting a cut-in, or altering the ego's path) retains the realism of its underlying map and agents.

What "out-of-distribution" means here. Our created scenes are out-of-distribution *behaviorally*, ordinary agent classes in configurations nominal driving logs do not contain, while remaining in-distribution *representationally*, since every route emits the same nuPlan-format tensor the occupancy model was trained on. That combination is the property Sec. 3.4 exploits. We do not do 3D class-level OOD: an unrecognized class is written to the generic-object channel and is represented in occupancy as a generic object rather than a novel category.

3.4 BEV → Occupancy

Given the unified BEV tensor $\mathcal{B}$ and the agent boxes that populate it, this stage synthesizes the dense 3D semantic occupancy $\mathcal{O} \in \{0, \ldots, 8\}^{400\times400\times32}$ that the downstream LiDAR renderer consumes. Prior occupancy-centric generators treat this as volumetric denoising: a latent diffusion model reconstructs a compressed occupancy volume from BEV conditioning that is concatenated to the noisy latent [20, 21]. We instead adopt an *entity-centric* formulation. The scene is a sparse set of *grounded latents*, one anchored to each agent, together with a background lattice, decoded to occupancy by semantic-Gaussian splatting. This design follows the grounded-latent generation from Park *et al.* [30], and it fits the safety-critical regime: because each agent is placed explicitly from its box rather than sampled from noise, every specified agent is explicitly instantiated in the latent representation; it need only render an object whose position, class, and pose are already given. Figure 2 illustrates the pipeline.

Grounded latent set. The scene is encoded as a fixed-size set of Q latents partitioned into foreground and background. Each *foreground* latent is instantiated from a single agent box and positioned at the box center, carrying that agent's class, yaw, and extents; the number of foreground latents varies with the scene and is padded to Q. The *background* latents form a regular 3D lattice restricted to the drivable footprint of the map, representing static structure. Very large vehicles are split into a few sub-latents along the heading so that extended objects remain covered.

Semantic-Gaussian occupancy decoder. A shared decoder maps the latent set to occupancy. Each latent is processed by a transformer and emits a budget of anisotropic 3D Gaussians, each with a predicted offset, scale, and

opacity [17], which are splatted into the voxel grid and accumulated into per-voxel class logits; free space and semantics are decoded on separate channels so that emptiness competes directly with occupancy. Classes are allotted different Gaussian budgets, matching representational capacity to class geometry (App. A). This decoder is shared between the autoencoder that pretrains it and the BEV-conditioned generator that reuses it (below), so that generation inherits a decoder already capable of high-fidelity occupancy and need only learn to place latents correctly.

Latent autoencoder. The decoder is pretrained within a variational autoencoder that compresses ground-truth occupancy into the grounded latent set and reconstructs it, providing both the pretrained decoder and a class-structured latent space for the generator to target. To make that space a clean target we regularize it with a prototype-based contrastive loss (PBCL) adapted from ProOOD [54], applied only during autoencoder training (App. A).

BEV-conditioned generation. The generator learns the mapping from the BEV input to the grounded latents and then decodes with the shared decoder. A 2D convolutional encoder embeds $\mathcal{B}$, including its auxiliary height channels, into a spatial feature map, and each latent *samples* its own conditioning feature from this map at its anchored (x, y) position. This grounded feature sampling, rather than concatenating a single global BEV embedding to every latent, ties each latent's appearance to the local layout at its own location. For foreground latents we additionally condition on the box extents, height included, so that the layout can steer object geometry (the stylization control of Sec. 3.1). The nuPlan BEV provides no explicit structure channel for buildings or vegetation, so background must be inferred from the map layout alone. We add a background head that predicts background occupancy from the layout and applies it as a gated residual (App. A). The generator is trained end-to-end from the warm-loaded decoder with a Lovász-softmax objective [3], which directly optimizes intersection-over-union, together with a class-weighted cross-entropy that up-weights rare classes.

3.5 Occupancy → LiDAR

We render each generated occupancy grid into LiDAR sweeps for the nuPlan sensor rig using the pretrained, frozen occupancy-to-LiDAR model of UniScene-v2 [21]. We neither modify nor retrain it, and we apply it identically to every method and to all three scene-creation routes of Sec. 3.1. Because the renderer is shared and frozen, differences in LiDAR fidelity isolate the upstream occupancy of Sec. 3.4 rather than the renderer itself (Sec. 4.2).

4 Experiments

4.1 Setup

nuPlan-Occ. Our nominal-scene experiments build on nuPlan-Occ [21], the dense semantic occupancy annotation of nuPlan [6]: each frame is an ego-centered 100×100 m scene voxelized to a $400 \times 400 \times 32$ grid at 0.25 m planar and 0.4 m

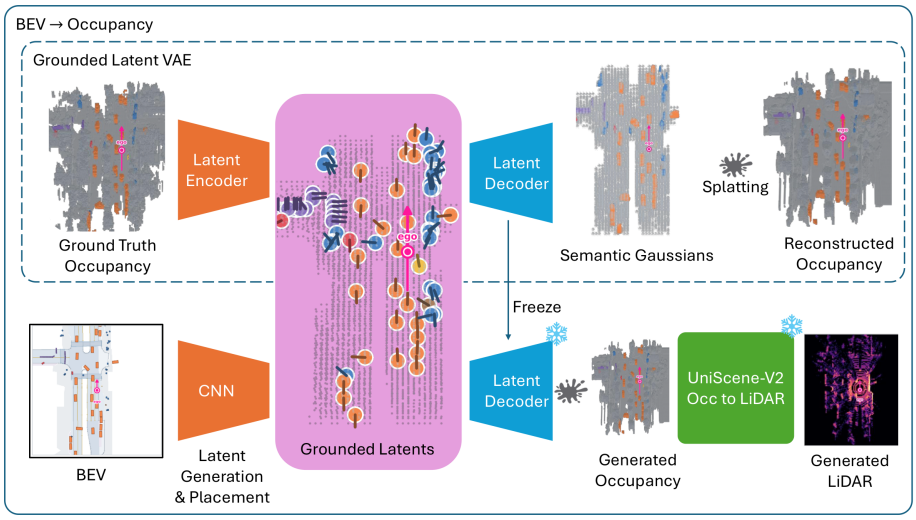

Fig. 2: BEV to occupancy and LiDAR. (Top) The grounded-latent VAE, following Park *et al.* [30], compresses occupancy into a latent set and decodes it by semantic-Gaussian splatting. (Bottom) At generation time a CNN encodes the BEV, each grounded latent samples its conditioning feature at its anchored position, and the shared decoder produces occupancy, which UniScene-v2's [21] occupancy-to-LiDAR model renders into sweeps. The snowflake marks frozen models used for inference only.

vertical resolution, labeled with the nine classes of Sec. 3.4 (free, background, vehicle, bicycle, pedestrian, cone, barrier, construction zone, generic object). The generator is trained on a 200k-token subset (nuPlan indexes frames by unique tokens) of the nuPlan training split. We evaluate on two nested held-out views of the official validation split:

- Mini-Val (43,644 frames): every annotated frame of the official mini validation set of nuPlan-Occ, the set on which UniScene-v2 reports its published results [21]. It is the complete split and our primary set for comparison.
- Key-457 (457 frames): one key frame per validation clip.

Our models train on the 200k train-split subset, and UniScene-v2's released checkpoint is trained on the full train split. The generator consumes the BEVs of Sec. 3.1 together with the frame's agent boxes; both are produced identically for real nuPlan frames and for custom scenes.

CIREN. From each NHTSA CIREN case we use only the hand-drawn collision sketch and the natural-language summary; by selection, these cases are the severe long-tail events driving logs do not contain. We process 2,180 cases, drawn from the current and legacy CIREN case series [27, 28], into 3,854 ego-centric clips and 65,518 frames. Custom OOD scenes (Sec. 3.3) have no paired ground truth and are scored qualitatively. Quantitative analysis of out-of-distribution scene feasibility and generation quality is future work.

Implementation details. We first train the grounded-latent autoencoder on nuPlan-Occ; the prototype loss (PBCL) is enabled during autoencoder training after a short warmup. We then train the BEV-conditioned generator (the BEV encoder, feature head, and decoder), initializing its decoder from the pretrained autoencoder decoder and fine-tuning end to end. The generator trains for 12 epochs with Adam at learning rate 10^{-4} under cosine decay, batch size 2 per GPU, in mixed precision; the reported checkpoint is selected at epoch 11. Full hyperparameters are in App. A. Inference is a single deterministic forward pass per frame. The UniScene-v2 baseline runs from its released checkpoint at the noise-prior value λ=0.05.

Metrics. We report per-class IoU accumulated over the dataset under two averaging conventions plus a binary criterion, using a single shared scorer for both methods on identical ground truth and tokens. **fg-mIoU**, our primary metric, is the mean IoU over the seven foreground/agent classes, excluding `free` and `background`. **mIoU**$_{\text{US}}$ is UniScene-v2's mIoU metric, reported so our numbers can be read against theirs. We lead with fg-mIoU, which removes the easy `free` mass and measures what a scene generator prioritizes, the foreground actors of the scene. **occ-IoU** is the binary occupied-vs-free IoU.

4.2 Fidelity on Nominal Scenes

This experiment measures how faithfully each method realizes ordinary, in-distribution driving scenes, where paired ground truth exists. All methods are conditioned on the ground-truth BEV layouts of the validation split and scored against the corresponding ground-truth occupancy and LiDAR. Our primary comparison is a controlled head-to-head against the released UniScene-v2 occupancy model: identical ground truth, identical frames, and a single shared scorer (Sec. 4.1).

Occupancy Control (BEV → Occupancy)

Table 1 compares our grounded generator with the released UniScene-v2 occupancy model; App. D breaks the comparison down by class.

The method comparison Both methods solve BEV-to-occupancy from the same nuPlan tensor $\mathcal{B}$. ROCC additionally receives the per-agent 3D boxes, but those belong to the *specification* rather than to the target: the dynamic channels of $\mathcal{B}$ already rasterize every agent's footprint, and the box adds only the vertical extent and pose that a 2D raster cannot express (Sec. 3.1). In all three creation routes the boxes are authored rather than measured. We provide a comparison in the results with a CNN (Dense CNN) that does not explicitly place the latents with the BEV boxes.

Results. On Mini-Val, ROCC reaches 49.45 fg-mIoU, 52.11 mIoU$_{\text{US}}$ and 28.21 occ-IoU, against 20.19, 29.53 and 22.24 for the released UniScene-v2 checkpoint: margins of 29.3, 22.6 and 6.0 points. The two systems target different objectives, ours being controllable per-timestep generation from BEV and theirs diffusion-style generation over multiple timesteps. The gap is smaller under mIoU$_{\text{US}}$ only because free space dominates that average. The advantage holds for every class,

Metric	UniScene-v2	Dense CNN	**ROCC (ours)**
fg-mIoU ↑	20.19	33.55	**49.45**
mIoU_{US} ↑	29.53	38.09	**52.11**
occ-IoU ↑	22.24	24.74	**28.21**

Table 1: Occupancy generation on Mini-Val. One shared per-class IoU accumulator scores every column on identical ground truth and tokens. UniScene-v2 is the released checkpoint at its default inference settings. The dense CNN shares our BEV input and supervision but has no boxes, latents, or Gaussian decoder.

and is largest for the rare agents a long-tail generator most needs to preserve: construction-zone markers (41.03 vs. 2.73), bicycles (48.71 vs. 8.19) and barriers (52.03 vs. 25.22) (App. D). We score UniScene-v2 ourselves under the shared accumulator and measure 29.53 mIoU_{US}; the value its authors report for this split is 32.22 [21], and the comparison holds under either.

Both systems share a comparable ceiling. A generator can only be as good as the latent space it decodes from. On Key-457, the 457-frame key-frame subset, our grounded latent set reconstructs at 61.70 mIoU_{US}, and UniScene-v2 reports 62.4 for the latent its diffusion model samples in [21]. Neither system wins on the codec. What differs is how much of that ceiling each generator recovers: ours reaches 83% (51.14/61.70), against 52% (32.22/62.4) for UniScene-v2, and 87% (48.38/55.73) on our primary metric. Each ratio pairs a generation score with the ceiling measured alongside it on the same set. Because the boxes supply placement, the generator's remaining task is to complete the geometry of agents whose position, class and pose are already fixed.

Temporal Behavior of 4D Occupancy

Both systems emit a 4D volume and every frame is conditioned on its own BEV, so neither forecasts. The question is whether quality is uniform along a clip or concentrated in one privileged frame. We partition Mini-Val into the 8,602 non-overlapping five-frame clips its 218 validation scenes admit (43,010 frames, 0.1 s spacing), and each model generates all five frames of a clip in one invocation under the recipe of Sec. 4.2.

Results. ROCC maintains consistently high quality throughout the clip, while UniScene-v2 degrades rapidly away from the reference frame. This difference follows from their conditioning: UniScene-v2 injects a single-frame ground-truth latent into the entire clip, which becomes increasingly stale as the scene evolves, whereas ROCC generates each frame from its corresponding BEV and boxes. ROCC therefore avoids reference-frame bias and provides uniform per-frame fidelity, but this stability is not learned temporal coherence.

LiDAR Fidelity

The renderer is frozen and identical for every row, so the differences in Tab. 3 come from the upstream occupancy alone. The gap between the two generators is wider here than in Tab. 1, and the reason lies in the sensor model rather than

	fg-mIoU ↑		mIoU_{US} ↑		occ-IoU ↑	
Frame	UniScene-v2	**ROCC**	UniScene-v2	**ROCC**	UniScene-v2	**ROCC**
t_0	19.93	**49.46**	29.30	**52.11**	22.17	**28.26**
t_1	8.44	**49.46**	20.93	**52.12**	18.42	**28.25**
t_2	5.89	**49.46**	18.56	**52.12**	16.39	**28.25**
t_3	4.71	**49.45**	17.48	**52.11**	15.34	**28.24**
t_4	4.07	**49.49**	16.91	**52.14**	14.73	**28.25**
Mean	8.61	**49.46**	20.63	**52.12**	17.41	**28.25**

Table 2: Quality along the clip. All three metrics are resolved by frame index over the 8,602 five-frame clips of Mini-Val, one shared scorer, same recipe as Tab. 1.

Method	MMD (10^{-5}) ↓	JSD ↓
Ground-truth occupancy (renderer oracle)	0.457	0.028
UniScene-v2, end to end (λ=0.05)	33.710	0.186
ROCC (ours)	**1.714**	**0.057**

Table 3: LiDAR fidelity on nuPlan-Occ Mini-Val (43,644 frames). One frozen occupancy-to-LiDAR model [21] renders every row, so the upstream occupancy generator is the only variable. The first row renders ground-truth occupancy: the ceiling this renderer imposes, not a competing method. Bold marks the best end-to-end result.

in the generators. A ray stops at the first surface it hits, so a single misplaced surface deflects every ray behind it, while an occupancy metric counts only the voxels of that surface. LiDAR fidelity therefore penalises structural error more heavily than volume overlap, and ROCC's advantage over the diffusion baseline widens accordingly. The oracle row bounds what any occupancy generator can recover through this renderer.

4.3 Controllability

Controllability is what turns a scene generator into a simulator: the user must change one agent and get exactly that change. Following the editing protocols of X-Scene and SemCity [19, 45], we apply *insert*, *delete* and *move* to 1,000 random Mini-Val scenes and score *box coverage*, the fraction of a requested box's ground-plane columns that carry the target class. We insert every foreground class, since preserving rare agents is the point of long-tail generation. Coverage cannot reach 1: LiDAR-derived occupancy captures surfaces, not filled boxes, so agents already present in an unedited scene reach only 0.81, which we use as the practical ceiling. Because each foreground latent is instantiated from its box, ROCC materializes edited agents. Insertion averages 0.70 over the seven classes, and an inserted vehicle reaches 0.93 against 0.92 for a vehicle already in the scene, so an edit is generated no differently from an ordinary scene. A moved agent reaches 0.74 at its new pose while leaving 0.04 at the old, and deletion clears 0.97. The remaining gap sits in the smallest classes, cones and

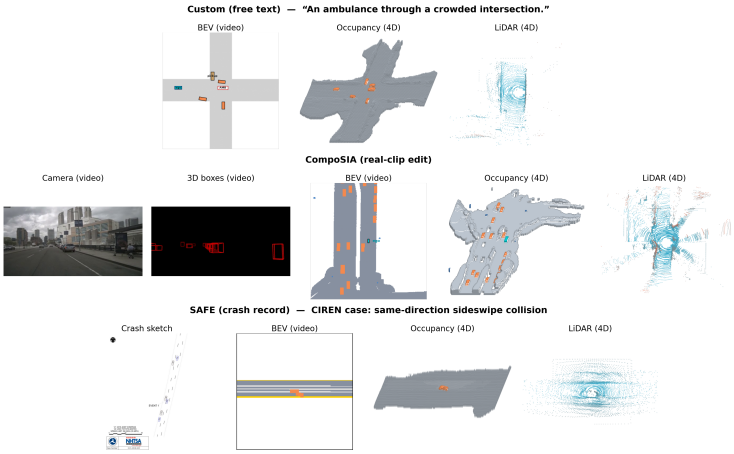

Fig. 3: Qualitative results across the three creation routes. Top: a free-text scene (*"an ambulance through a crowded intersection"*) compiled to BEV, occupancy, and LiDAR. Middle: a CompoSIA edit of a real clip (*"a vehicle cuts into the lane to the right of the ego"*), with the source camera frame and edited 3D boxes that produce the BEV. Bottom: a CIREN crash case, from the NHTSA crash sketch to the reconstructed same-direction sideswipe collision. All three routes share the same BEV interface and generator.

construction-zone markers, whose footprints make coverage sensitive to small localization errors. Per-class results and the protocol are in App. E.

4.4 Qualitative Results

Figure 3 shows one worked example per creation route, each carried from its input specification to occupancy and LiDAR. Across all three the same BEV-conditioned generator produces the sensor data, so route choice trades grounding against authorial control without changing the output format. Each requested change lands: an inserted agent is materialized, a deleted one is cleared from its pose, and a moved one appears at the new pose and not the old. Generated examples from each creation route, including animated sequences and failure cases, are browsable at `https://rareocc-anon.github.io/`; static versions are in App. F for custom generation.

4.5 Ablations

Every ablation is a single-variable change from the reported model: same autoencoder lineage, same training tokens, same schedule, same three-GPU batch. Results are in Tab. 4. The dense CNN is the one exception, having no latent decoder to warm-start, and is trained from scratch.

How precise must the conditioning be? Because the BEV already carries per-class agent mask channels, the question is not whether layout conditioning helps but how exact it must be, and the dense CNN answers it by consuming the same BEV under the same supervision while never anchoring a latent to

Variant	fg-mIoU ↑	mIoU_{US} ↑	occ-IoU ↑
ROCC, calibrated bg head (reported)	**49.45**	**52.11**	**28.21**
no background head	49.25	51.65	27.25
naive background head	45.62	49.01	24.19
no PBCL in the autoencoder	49.35	51.74	27.24

Table 4: Single-variable ablations on Mini-Val, all trained under one protocol. The no-PBCL model is trained without a background head, so it is read against the no-head row.

a box. Removing that anchoring costs 15.9 fg-mIoU on Mini-Val, from 49.45 to 33.55, but the dense CNN also forgoes the latent set, the Gaussian decoder and the autoencoder pretraining, so this figure bounds the value of grounding from above. The per-class breakdown in Tab. 5 shows the gap concentrated in the rare classes, where the dense CNN reaches 16.1 on bicycle, 22.8 on cone and 18.9 on construction zone against our 48.7, 49.1 and 41.0, while trailing on more frequent classes such as pedestrian and generic by only 6.1 and 3.5.

Background head and prototype loss. Neither of the remaining two components moves the result far, but they behave differently. Because the BEV has no structure channel, background must be inferred from the map layout, and how that prediction is combined matters more than whether it is made: a naive head that adds background logits directly costs 3.63 fg-mIoU and 3.06 occ-IoU, whereas the calibrated head of Sec. 3.4 gains 0.20 fg-mIoU, 0.46 mIoU_{US} and 0.96 occ-IoU. That gain sits in background itself (+1.07) and in the static classes bordering it (construction zone +0.97, barrier +0.82, cone +0.74), against −1.13 on generic. Removing PBCL from the autoencoder, with the background head disabled on both sides, *raises* fg-mIoU by 0.10 and mIoU_{US} by 0.09 and leaves occ-IoU unchanged, so the prototype loss makes little impact on this model. PBCL improves class separation in the latent space, which we expect to help a generator that samples latents; here every foreground latent is instantiated with its class already fixed, so that separation is not exercised. The reported model keeps PBCL.

5 Discussion and Limitations

The two limitations we consider most important follow from the same source. The BEV determines what a user is able to specify, and the occupancy stage can only render what its channels are able to express. The nuPlan BEV provides no channel for static structure, which leaves background as our weakest class at 25.67 IoU and constrains the range of scenes we can author more than it constrains the accuracy of the scenes we do author. A scene whose event involves background, such as a vehicle leaving the road and striking a tree, cannot be specified at all, since there is no channel in which the tree can be placed. A related boundary applies to agents. Every vehicle seen during training sits on drivable surface, so an agent placed off the road tends to acquire road beneath it,

because the generator resolves the ambiguity toward its training prior. Adding an explicit structure channel to the BEV would address both cases.

Temporal behavior. ROCC generates each frame independently, and temporal coherence is therefore inherited from the conditioning. Because latents are anchored one-to-one to boxes, an agent's latent persists and translates with its box across frames, and any region of the BEV that is left unchanged decodes identically. The one mechanism that can perturb content the user did not touch is the reallocation of the fixed-size background budget when the agent count changes. We leave a quantitative temporal-consistency metric to future work.

Future work. We see two directions as the most promising. Generated nominal occupancy and LiDAR are known to transfer to perception and planning [20,22], and establishing the same for crash-grounded and out-of-distribution scenes would require training a rare-class perception model with ROCC augmentation and stress-testing a planner on the CIREN corpus. Separately, the kinematic rollout latches the frame of deepest overlap, so extending the model to deformation and post-impact dynamics would carry crash grounding past the moment of impact.

6 Conclusion

We presented RareOcc, an end-to-end system that turns a specification of a rare driving situation, whether a documented crash, a sentence, or an edit to a real clip, into 4D semantic occupancy and LiDAR. Its organizing idea is a single editable BEV tensor that all three creation routes write to and every synthesis stage reads from, which allows an out-of-distribution scene to reach the occupancy model in an in-distribution representation. Our BEV-to-occupancy stage replaces volumetric denoising with an entity-grounded formulation in which each agent is instantiated from its box and rendered by semantic-Gaussian splatting. On the complete mini validation split it reaches 49.45 foreground occupancy mIoU against 20.19 for the released UniScene-v2 generator under one shared scorer, and it recovers 83% of a reconstruction ceiling that the two autoencoders share, against 52% for the baseline. The three creation routes are complementary, since recorded crashes anchor the tail to events that did happen while edits and free text extend it to events that have not.

Acknowledgements

This work was carried out during a summer internship by Arthur Jakobsson and Yu-Rou Tuan at the DENSO Pittsburgh Innovation Lab, DENSO INTERNATIONAL AMERICA, INC.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Bao, W., Yu, Q., Kong, Y.: Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In: ACM MM (2020)
3. Berman, M., Triki, A.R., Blaschko, M.B.: The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR (2018)
4. Bian, H., Kong, L., Xie, H., Pan, L., Qiao, Y., Liu, Z.: DynamicCity: Large-scale 4D occupancy generation from dynamic scenes. In: ICLR (2025)
5. Blough, J.: Delays, detours, and passed-out riders: Waymos keep blocking SF firefighters. The San Francisco Standard (Jul 2026), `https://sfstandard.com/2026/07/10/waymo-robotaxi-emergency-response/`
6. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. In: CVPR (2021), aDP3 Workshop
7. Chan, F.H., Chen, Y.T., Xiang, Y., Sun, M.: Anticipating accidents in dashcam videos. In: ACCV (2016)
8. Chitta, K., Dauner, D., Geiger, A.: SLEDGE: Synthesizing driving environments with generative models and rule-based traffic. In: ECCV (2024)
9. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conf. Robot Learn. (2017)
10. Fang, J., Yan, D., Qiao, J., Xue, J., Yu, H.: DADA: Driver attention prediction in driving accident scenarios. IEEE Trans. Intell. Transp. Syst. (2022)
11. Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q.: MagicDrive: Street view generation with diverse 3D geometry control. In: ICLR (2024)
12. Gao, R., Chen, K., Xie, E., Hong, L., Li, Z., Yeung, D.Y., Xu, Q.: MagicDrive-V2: High-resolution long video generation for autonomous driving with adaptive control. arXiv preprint arXiv:2411.13807 (2024)
13. 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. In: NeurIPS (2024)
14. Gosselin, A., Luo, G.Y., Lara, L., Golemo, F., Nowrouzezahrai, D., Paull, L., Jolicoeur-Martineau, A., Pal, C.: Ctrl-Crash: Controllable diffusion for realistic car crashes. arXiv preprint arXiv:2506.00227 (2025)
15. Gulino, C., Fu, J., Luo, W., Tucker, G., Bronstein, E., Lu, Y., Harb, J., Pan, X., Wang, Y., Chen, X., Co-Reyes, J.D., Agarwal, R., Roelofs, R., Lu, Y., Montali, N., Mougin, P., Yang, Z., White, B., Faust, A., McAllister, R., Anguelov, D., Sapp, B.: Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In: NeurIPS (2023), datasets and Benchmarks Track
16. Hu, Q., Zhang, Z., Hu, W.: RangeLDM: Fast realistic LiDAR point cloud generation. In: ECCV (2024)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3D Gaussian splatting for real-time radiance field rendering. ACM TOG **42**(4) (2023)
18. Kong, L., Liu, Y., Li, X., Chen, R., Zhang, W., Ren, J., Pan, L., Chen, K., Liu, Z.: Robo3D: Towards robust and reliable 3D perception against corruptions. In: ICCV (2023)
19. Lee, J., Lee, S., Jo, C., Im, W., Seon, J., Yoon, S.E.: SemCity: Semantic scene generation with triplane diffusion. In: CVPR (2024)

20. Li, B., Guo, J., Liu, H., Zou, Y., Ding, Y., Chen, X., Zhu, H., Tang, F., Zhang, C., Wang, K., et al.: UniScene: Unified occupancy-centric driving scene generation. In: CVPR (2025)
21. Li, B., Jin, X., Zhu, H., Liu, H., Li, R., Guo, J., Cai, K., Ma, C., Jin, Y., Zhao, H., Yang, X., Zeng, W.: Scaling up occupancy-centric driving scene generation: Dataset and method. arXiv preprint arXiv:2510.22973 (2025)
22. Li, B., Ma, Z., Du, D., Peng, B., Liang, Z., Liu, Z., Guo, X., Zhu, Z., Ma, C., Jin, Y., Jin, X., Zhao, H., Zeng, W.: OmniNWM: Omniscient driving navigation world models. In: ECCV (2026)
23. Li, M., Ding, W., Lin, H., Lyu, Y., Yao, Y., Zhang, Y., Zhao, D.: CrashAgent: Crash scenario generation via multi-modal reasoning. arXiv preprint arXiv:2505.18341 (2025)
24. Li, Q., Peng, Z., Feng, L., Liu, Z., Duan, C., Mo, W., Zhou, B.: ScenarioNet: Open-source platform for large-scale traffic scenario simulation and modeling. In: NeurIPS (2023), datasets and Benchmarks Track
25. Li, Q., Peng, Z., Feng, L., Zhang, Q., Xue, Z., Zhou, B.: MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. IEEE TPAMI (2022)
26. Luo, S., Zhang, Y., Deng, Y., Liang, L., Zheng, X.: SAFE: Harnessing LLM for scenario-driven ADS testing from multimodal crash data. In: Int. Conf. Softw. Eng. (2026)
27. National Highway Traffic Safety Administration: Crash injury research and engineering network (CIREN): Legacy case series. `https://crashviewer.nhtsa.dot.gov/ciren`, u.S. Department of Transportation
28. National Highway Traffic Safety Administration: Crash injury research and engineering network (CIREN): 2017 case series. `https://crashviewer.nhtsa.dot.gov/ciren` (2017), u.S. Department of Transportation
29. National Highway Traffic Safety Administration: Consent order with Cruise after company failed to fully report crash involving pedestrian. NHTSA (2024), `https://www.nhtsa.gov/press-releases/consent-order-cruise-crash-reporting`
30. Park, J., Sanghvi, N., Weng, E., Hunt, S., Tanaka, S., Fujiyoshi, H., Kitani, K.: Grounded latents for entity-centric 4D scene generation. In: CVPR (2026)
31. Ran, H., Guizilini, V., Wang, Y.: Towards realistic scene generation with LiDAR diffusion models. In: CVPR (2024)
32. Rempe, D., Philion, J., Guibas, L.J., Fidler, S., Litany, O.: Generating useful accident-prone driving scenarios via a learned traffic prior. In: CVPR (2022)
33. 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. In: Conf. Robot Learn. (2024)
34. Tan, S., Ivanovic, B., Weng, X., Pavone, M., Kraehenbuehl, P.: Language conditioned traffic generation. In: Conf. Robot Learn. (2023)
35. Tonderski, A., Lindström, C., Hess, G., Ljungbergh, W., Svensson, L., Petersson, C.: NeuRAD: Neural rendering for autonomous driving. In: CVPR (2024)
36. Wang, L., Zheng, W., Ren, Y., Jiang, H., Cui, Z., Yu, H., Lu, J.: OccSora: 4D occupancy generation models as world simulators for autonomous driving. arXiv preprint arXiv:2405.20337 (2024)
37. Wang, T., Kim, S., Wenxuan, J., Xie, E., Ge, C., Chen, J., Li, Z., Luo, P.: DeepAccident: A motion and accident prediction benchmark for V2X autonomous driving. In: AAAI (2024)

38. Wei, J., Yuan, S., Li, P., Hu, Q., Gan, Z., Ding, W.: OccLLaMA: An occupancy-language-action generative world model for autonomous driving. arXiv preprint arXiv:2409.03272 (2024)
39. 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: CVPR (2024)
40. Emergency first responders say waymos are getting worse. WIRED (2026), https://www.wired.com/story/emergency-first-responders-say-waymos-are-getting-worse/
41. Xie, S., Kong, L., Zhang, W., Ren, J., Pan, L., Chen, K., Liu, Z.: Benchmarking and improving bird's eye view perception robustness in autonomous driving. IEEE TPAMI (2025)
42. Yan, T., et al.: DrivingSphere: Building a high-fidelity 4D world for closed-loop simulation. In: CVPR (2025)
43. 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: ECCV (2024)
44. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: EmerNeRF: Emergent spatial-temporal scene decomposition via self-supervision. In: ICLR (2024)
45. Yang, Y., Liang, A., Mei, J., Ma, Y., Liu, Y., Lee, G.H.: X-Scene: Large-scale driving scene generation with high fidelity and flexible controllability. arXiv preprint arXiv:2506.13558 (2025)
46. Yang, Y., Mei, J., Ma, Y., Du, S., Chen, W., Qian, Y., Feng, Y., Liu, Y.: Driving in the occupancy world: Vision-centric 4D occupancy forecasting and planning via world models for autonomous driving. In: AAAI (2025)
47. Yang, Z., Chen, Y., Wang, J., Manivasagam, S., Ma, W.C., Yang, A.J., Urtasun, R.: UniSim: A neural closed-loop sensor simulator. In: CVPR (2023)
48. Yao, Y., Wang, X., Xu, M., Pu, Z., Wang, Y., Atkins, E., Crandall, D.J.: DoTA: Unsupervised detection of traffic anomaly in driving videos. IEEE TPAMI (2022)
49. Zhan, Y., Chen, Z., Wang, Q., He, Z., Niu, M., Guo, X., Yin, W., Ren, W., Zhang, Q., Zheng, Y.: Composing driving worlds through disentangled control for adversarial scenario generation. In: ECCV (2026)
50. Zhang, H., Zhou, J., Jiang, F., Li, J., Guo, Z., Dai, P., Dai, J., Xie, Y., Zhu, B.: AnyScene: Towards highly controllable driving scene generation at anywhere and beyond. arXiv preprint arXiv:2605.26113 (2026)
51. Zhang, J., Xu, C., Li, B.: ChatScene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In: CVPR (2024)
52. Zhang, X., Zhang, Q., Han, L., Qu, Q., Chen, X., Cai, W.: AccidentSim: Generating vehicle collision videos with physically realistic collision trajectories from real-world accident reports. arXiv preprint arXiv:2503.20654 (2025)
53. Zhang, Y., Duan, M., Peng, K., Wang, Y., Liu, R., Teng, F., Luo, K., Li, Z., Yang, K.: Out-of-distribution semantic occupancy prediction. arXiv preprint arXiv:2506.21185 (2025)
54. Zhang, Y., Duan, M., Peng, K., Wang, Y., Wen, D., Paudel, D.P., Gool, L.V., Yang, K.: ProOOD: Prototype-guided out-of-distribution 3D occupancy prediction. In: CVPR (2026)
55. Zhao, G., Wang, X., Zhu, Z., Chen, X., Huang, G., Bao, X., Wang, X.: DriveDreamer-2: LLM-enhanced world models for diverse driving video generation. arXiv preprint arXiv:2403.06845 (2024)

56. Zheng, W., Chen, W., Huang, Y., Zhang, B., Duan, Y., Lu, J.: OccWorld: Learning a 3D occupancy world model for autonomous driving. In: ECCV (2024)
57. Zhong, Z., Rempe, D., Chen, Y., Ivanovic, B., Cao, Y., Xu, D., Pavone, M., Ray, B.: Language-guided traffic simulation via scene-level diffusion. In: Conf. Robot Learn. (2023)
58. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: DrivingGaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes. In: CVPR (2024)
59. Zyrianov, V., Che, H., Liu, Z., Wang, S.: LidarDM: Generative LiDAR simulation in a generated world. In: ICRA (2025)
60. Zyrianov, V., Zhu, X., Wang, S.: Learning to generate realistic LiDAR point clouds. In: ECCV (2022)

A Implementation Details

Autoencoder training. The grounded-latent autoencoder trains with Adam at learning rate 3×10^{-5} under cosine decay in mixed precision (fp16). Its loss is a class-frequency-weighted cross-entropy (weight clip 25, with per-class boosts) plus Lovász-softmax plus a KL term weighted 0.0015. PBCL is enabled after a 500-step warmup with EMA momentum 0.9, temperature 0.1, and loss weight 0.3.

Architecture. The latent set has Q=4096 latents of 128 channels, up to 512 of them foreground. The shared decoder has 384 channels, 6 transformer blocks, and 6 heads; each latent emits up to 96 anisotropic Gaussians with offsets bounded to 7.0 voxels and a minimum scale of 0.25.

Per-class Gaussian budget. Each class activates a different number of its per-latent Gaussians, matching representational capacity to class geometry: vehicles receive 96, mid-size agents 48, background latents 32, and small agents 24; only active Gaussians are splatted.

Generator training. The generator (BEV encoder, feature head, and decoder) initializes its decoder from the pretrained autoencoder and fine-tunes end to end with the Lovász-softmax and class-weighted cross-entropy objectives of Sec. 3.4, using Adam at 10^{-4} under cosine decay for 12 epochs, batch 2 per GPU on 3 GPUs (effective batch 6), in mixed precision; the reported checkpoint is epoch 11. Inference is a single deterministic forward pass per frame.

Prototype-based contrastive loss (PBCL). PBCL maintains an exponential-moving-average bank of per-class latent prototypes, updated with ground-truth labels and no gradient. After the warmup, a supervised contrastive objective pulls each L2-normalized latent toward its own class prototype and repels the others, with the intent of a class-separable latent space and hence a more predictable mapping for the generator to learn. PBCL is applied only during autoencoder training and is enabled in the autoencoder we report. We ablate it in Sec. 4.5; at matched configuration, removing it is neutral to marginally positive.

Background head We add a small convolutional head that predicts a background logit volume from the shared BEV features and folds it into the decoder output as a residual, moved from the free channel to the background channel so the agent logits are untouched. The head is zero-initialised, so it starts as an identity and grows only where it helps; background is supervised directly by an auxiliary cross-entropy term; and a foreground gate suppresses the residual wherever an agent logit already dominates a voxel, so structure is never written over a placed object. Sec. 4.5 gives the gain: foreground unchanged, occ-IoU +0.96 on Mini-Val and +1.05 on Key-457, concentrated in background and the static classes bordering it. The reported model uses this head; the naive variant that adds background logits without the gate loses 3.63 fg-mIoU.

B Crash-Grounded Scene Extraction

We augment the scenario extracted by SAFE with a deterministic conflict descriptor recording the crash structure: the at-fault vehicle, the struck vehicle, the impact type, and the point of impact. A single additional vision-language call produces this descriptor, which we merge into the scenario without altering any field the extraction produced. From the augmented scenario we run a short kinematic rollout that stages the collision: the at-fault vehicle follows a smooth intercept trajectory toward the struck vehicle, and the frame of deepest overlap is detected and latched so later frames hold the post-impact configuration. Producing the conflict descriptor as a separate deterministic pass is deliberate. Re-invoking the stochastic extraction stage to add it would re-sample the scenario and can perturb an already-faithful scene; keeping the extraction fixed and layering the descriptor on top preserves the recorded crash while making the staged collision reproducible. Because a crash involves several vehicles, we rasterize the rollout once per vehicle taken as ego, so a single case yields several ego-centric clips that observe the same event from complementary viewpoints.

C Free-Text Behavior Specification

The free-text route parses a description into a structured scene schema through a text-only call: the road type, an agent roster with semantic classes, and a short brief naming the intended event. Interactive sliders for density, randomness, and seed override the schema deterministically, so the user retains direct control over the scene. Given the placed scene, a second stage assigns each agent a goal. We describe every agent to the model in words, its class, heading, and position relative to the junction, and a "traffic director" prompt returns a compact, scenario-agnostic goal: a target speed, a lateral action (hold lane, change lane, pull to the shoulder, or move away from a named agent), an optional stopping condition (at the stop line, or behind a named agent), and a turn. Because these primitives compose, one vocabulary expresses behaviors as varied as yielding to an emergency vehicle, stopping behind stalled debris, or halting for a crossing animal, with no scenario-specific logic. Planning runs in two modes: a single call that assigns all goals jointly, or a hierarchical per-agent mode in which agents are decided in turn with the previously decided goals in context, so that reactions

stay mutually consistent, the vehicle ahead of the ambulance easing aside while cross-traffic holds at the line. A universal controller then advances the scene, each agent driving its assigned goal under a shared car-following and no-overlap safety model, and every frame is rasterized into $\mathcal{B}$ by class.

D Per-Class Results

Class	**ROCC**	no-bg head	no PBCL	UniScene-v2	CNN
free	95.44	95.46	95.43	96.93	94.18
background	**25.67**	24.60	24.61	21.09	22.68
vehicle	48.36	**48.54**	48.26	21.43	41.12
bicycle	48.71	**48.90**	48.16	8.19	16.07
pedestrian	**56.49**	56.08	56.20	36.26	50.40
cone	**49.12**	48.38	48.68	24.39	22.82
barrier	**52.03**	51.21	51.24	25.22	38.56
construction zone	41.03	40.06	**41.34**	2.73	18.90
generic	50.42	51.55	**51.58**	23.10	46.97
fg-mIoU ↑	**49.45**	49.25	49.35	20.19	33.55
mIoU_{US} ↑	**52.11**	51.65	51.74	29.53	38.09
occ-IoU ↑	**28.21**	27.25	27.24	22.24	24.74

Table 5: Per-class IoU ↑ on MINI-VAL (43,644 frames). **ROCC** is the reported model (calibrated background head); no-bg head and no PBCL are the ablations of Sec. 4.5, and the no-PBCL run also has no background head, so it is read against the no-bg-head column. UniScene-v2 is at λ=0.05; CNN is the dense-CNN baseline. mIoU_{US} is free-inclusive, averaging free, background, and classes up to construction zone following UniScene-v2's convention (which omits the generic class); fg-mIoU averages the seven agent classes.

E Controllability

This appendix provides the full edit protocol and per-class results behind Sec. 4.3.

Edit protocol. Each edit modifies only the agent box list. We then re-rasterize the dynamic and auxiliary BEV channels from the edited boxes while keeping the static map unchanged. Re-rasterizing the original boxes reproduces the dataset BEV at IoU 1.000, validating the boxes as the source of truth for these edits. We sample 1,000 held-out scenes with a fixed seed and apply three operations: *insert* one agent of each foreground class, *delete* one agent between 5 and 30 m, and *move* one agent laterally by up to 3.5 m.

Insertion. Inserted agents are sampled from real boxes of the target class, preserving their heading and 3D extents. A placement is accepted only if it lies on a valid surface for that class and does not overlap an existing agent. Inserted-agent geometry therefore follows the real data distribution, but its range distribution may differ because valid collision-free placements are not uniform.

Edit adherence	Box coverage
Insert: requested agent, by class ↑	
vehicle	0.93
pedestrian	0.78
bicycle	0.72
barrier	0.68
generic object	0.63
construction zone	0.59
traffic cone	0.58
mean	0.70
Delete and move	
delete: residue at vacated pose ↓	0.03
move: residue at old pose ↓	0.04
move: coverage at new pose ↑	0.74
Reference: agents already in the scene	
existing vehicle, no edit ↑	0.92
existing agent, no edit ↑	0.81

Table 6: Full controllability results on 1,000 held-out scenes. Insert results are reported separately for all foreground classes; the main paper reports their mean. The reference rows score unedited agents under the same metric: 0.92 for vehicles specifically, 0.81 averaged over classes.

Metric. Box coverage counts ground-plane columns inside the requested box holding the target class within its vertical extent. Insert and move are scored at the requested pose; delete and move also score residue at the vacated one. The reference row applies the same metric to unedited agents, restricted to isolated instances so a neighbour cannot supply coverage. Delete and move results are averaged over classes with at least 30 instances.

Per-class behavior. Insertion succeeds across all foreground classes, but coverage is lower for small objects such as traffic cones and construction-zone markers. Their small footprints make the metric more sensitive to minor localization or shape errors. Moving an agent produces similar coverage at the new pose to insertion while leaving little residual occupancy at the original pose, indicating that ROCC responds to both the addition and removal implied by the edit.

F Additional Qualitative Results

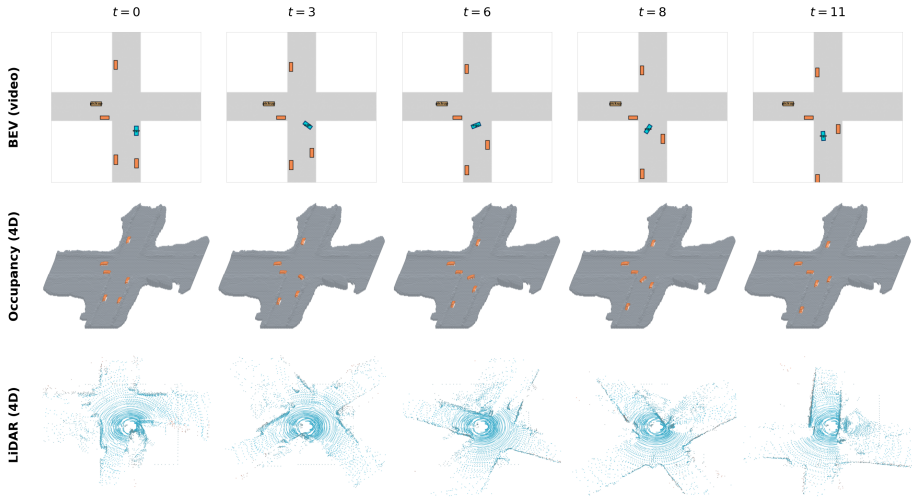

Fig. 4: Worked custom scene. *"A car makes a U-turn"*, specified in free text, unrolled over time (t=0 to t=11). Rows: the authored BEV video, the generated 4D occupancy, and the generated 4D LiDAR at the same timesteps. The turning car (cyan in the BEV) sweeps through the intersection while the queued cross traffic holds.

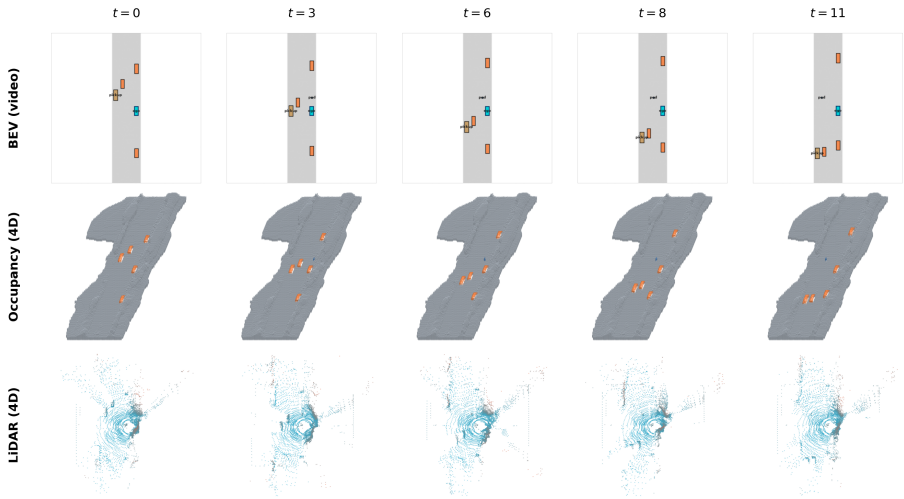

Fig. 5: Worked custom scene. *"A pedestrian crosses mid-block"*, unrolled over time (rows as in Fig. 4). The pedestrian (*ped* in the BEV) crosses between intersections while the pickup and surrounding traffic slow behind it.

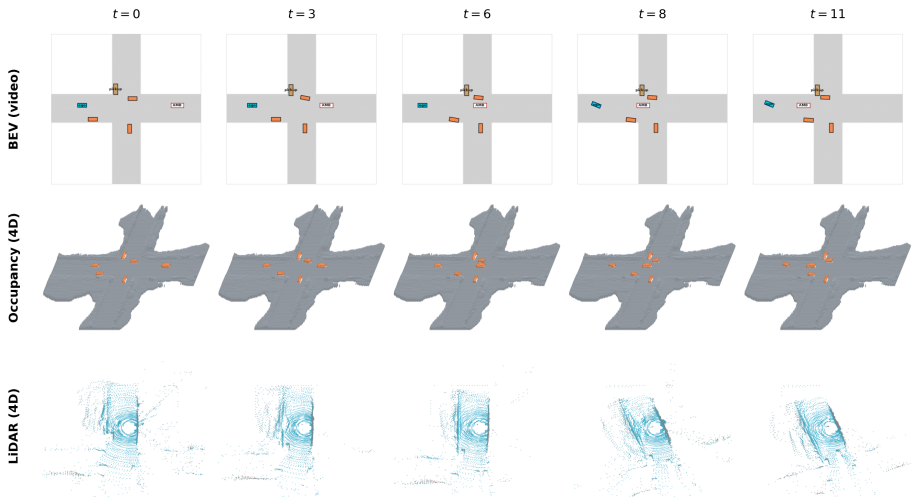

Fig. 6: Worked custom scene. *"An ambulance through a crowded intersection"*, unrolled over time (rows as in Fig. 4); the same scene appears as a single frame in Fig. 3. The ambulance (*AMB* in the BEV) crosses while the surrounding vehicles hold at the intersection.

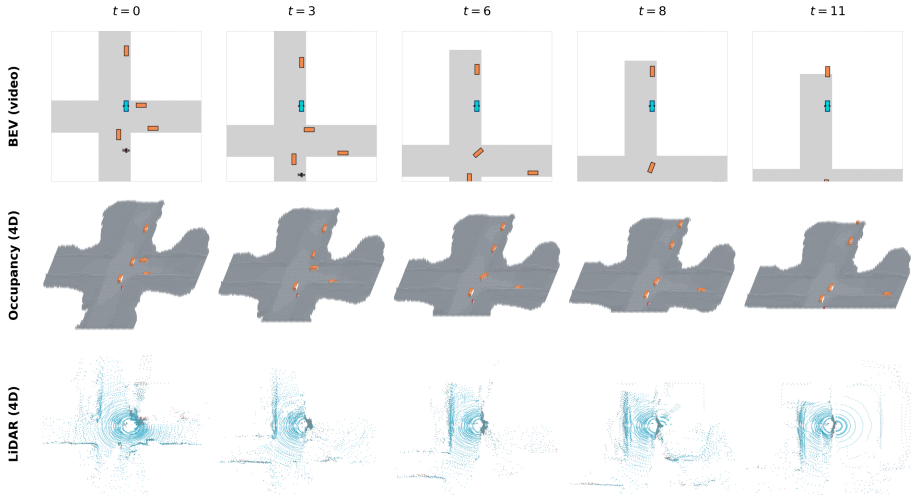

Fig. 7: Failure case: rare-class under-materialization. *"A cyclist waits at the red light between two stopped cars"*, unrolled over time (t=0 to t=11; rows as in Fig. 4). The BEV places the cyclist (black marker ahead of the cyan ego) between the queued vehicles, but the generator never materializes it in occupancy or LiDAR: the specified arrangement is lost exactly for the rare class the prompt is about.