SimForge: Generalization in Autonomous Driving through City-Scale Digital Twins and Simulation

Dibyendusekhar Goswami, Sourang Sri hari, Abhishek Shinde, Michael Vu, Ayush Gupta, Mayank Gupta, Anuj Gupta
SimForge, Richmond, CA, USA
{dib, sourang, abhishek, michael, ayush, mayank, anuj}@simforge.ai

***Abstract*—The critical bottleneck limiting AV generalization lies not in architectural capacity but in the breadth and coherence of training distributions across geographies, conditions, and failure modes [1]. This paper presents SimForge, a scalable data generation engine for safety critical and adversarial driving scenarios, addressing three coupled problems: world coverage, scenario relevance, and controlled diversity. SimScene reconstructs simulation ready digital twins and HD maps from aerial and dashcam imagery, grounded in real world risk through an accident and near miss abstraction loop. SimCloud enables scenario authoring through a road relative representation allowing reinstantiation across digital twins, and expands individual scenarios into dataset families by systematically varying environment conditions. The pipeline is paradigm agnostic: it produces distributional infrastructure to empirically test generalization under modular, end to end, or hybrid architectures.**

***Index Terms*—Autonomous driving, simulation, digital twins, scenario generation, dataset expansion, generalization.**

I. Introduction

Autonomous vehicle development depends fundamentally on simulation. The behaviors that determine whether an AV system is safe, such as multiagent conflicts at unprotected intersections, occluded vulnerable road users stepping into traffic, sudden infrastructure failures, degraded visibility in adverse weather, are precisely the behaviors that are rare, costly, and dangerous to collect at scale on public roads [2], [3]. No fleet, however large, can encounter the full tail of safety critical situations through naturalistic driving alone. Simulation offers the only tractable path to systematically generating, repeating, and varying these scenarios. Yet simulation does not automatically confer generalization. A system trained on a narrow set of hand authored virtual worlds, populated with ad hoc edge cases under fixed appearance conditions, inherits the same distributional blind spots as a log replay pipeline, only in synthetic form [1], [4]. The bottleneck, as recent empirical work confirms, is not model architecture but the breadth and coherence of the driving distributions a model actually sees during training [1]. This observation holds regardless of paradigm: whether the downstream consumer is a modular perception and planning stack or a monolithic end to end controller [5], [6], its generalization ceiling is set by the diversity and relevance of its training data.

Three coupled problems explain why current simulation workflows fail to deliver that diversity. First, producing simulation ready digital twins and HD maps remains labor intensive and tied to a handful of hand authored environments [4], [7], so training distributions are biased toward the few geographies that happen to have been built, and AV performance degrades sharply when deployed on unseen road geometries [8]. Second, scenario authoring is difficult to scale in a way that respects road topology and the statistical structure of real world incidents [9], [10]; without grounding in observed collision and near miss distributions, synthetic data risks overrepresenting common interactions while missing the rare failure modes that dominate actual crash reports [11], [12]. Third, even when good scenarios exist, converting them into large, structured, multimodal datasets with traceable configuration is typically disjoint from the tools used to author and run simulation [13], [14], making it difficult to audit which regions, failure modes, and operating conditions a model has been exposed to, and making reproducible cross paradigm benchmarking impractical, despite recent efforts to derive reactive closed loop evaluation from real world traces [15]. Existing tools address fragments of this problem: mature simulators like CARLA [4], AirSim [16], and LGSVL [17] provide physics and sensing but rely on hand authored worlds; map prior extensions [18], [19] widen spatial coverage but stop short of full reconstruction; neural scene representations [20]–[22] push fidelity but are not codesigned with OpenDRIVE maps or closed loop execution; and scenario languages [23], [24], fuzzing methods [25], and LLM based generation tools [26] improve authoring but keep scenario logic, map semantics, and batch execution weakly linked. The integration gap, from reconstruction through grounded, portable scenarios to traceable dataset families, remains open.

SimForge is designed to close that gap. The system comprises two tightly coupled subsystems. SimScene reconstructs urban scale digital twins and OpenDRIVE HD maps from aerial and dashcam imagery, augments them with semantic, risk, and visibility metadata, and grounds parameterized scenarios in real world accident and near miss patterns, ensuring that generated situations reflect observed failure distributions rather than arbitrary edge cases. SimCloud authors scenarios through a road relative intermediate representation that decouples scenario logic from world specific coordinates, so a single authored scenario can be reinstantiated across different digital twins without manual rework; it then expands each scenario into dataset families through systematic environmental sweeps, with every run governed by a configuration manifest that binds scenario, sensors, and conditions to outputs for full

reproducibility. A single tile can host dozens of road relative scenarios (e.g. 84 on one deployed Palo Alto asset in Sec. IV-B), each realized under many independent manifests, yielding orders of magnitude more multimodal, traceable realizations than fixed single condition or hand tuned simulation runs. The pipeline is deliberately paradigm agnostic: it produces the distributional infrastructure needed to empirically test whether broader geographic coverage, scenarios grounded in incidents, and systematic condition variation actually close the generalization gap, regardless of whether the model consuming that data is modular, end to end, or hybrid.

II. SimScene

A. Digital Twin Generation

SimScene reconstructs urban scale, simulation ready digital twins from widely available imagery, such as aerial photography (orthographic, nadir, oblique) and dashcam or street level sources, rather than proprietary fleet data [27], [28]. The pipeline proceeds through seven stages: **(i) Multi view photogrammetry** converts registered aerial and dashcam imagery to dense, georeferenced point clouds [29]; **(ii) Geometric cleanup and ground separation** isolates building clusters via sequential multiplane RANSAC and statistical outlier removal; **(iii) LOD2 building reconstruction** recovers watertight CityGML meshes with planar facades, α-shape roofs, and closed ground polygons, from aerial point clouds; **(iv) HD map extraction** derives OpenDRIVE lane networks and GeoJSON features coregistered with the 3D layout [30]; **(v) Facade and prop enrichment** adds doors, windows, and streetscape assets via selected manual refinement; **(vi) Procedural vegetation** populates sidewalks and medians; and **(vii) Asset graph export** packages the scene as OpenUSD/FBX for direct ingestion by engines such as CARLA [4]. The resulting twin is intended to be both *geometrically* and *photometrically* faithful to the imaged site: metric structure and HD maps from photogrammetry and coregistration, with facade, prop, and texture enrichment for plausible closed loop rendering. Maps are further augmented with structured metadata encoding semantic, risk, visibility, and language aligned annotations to support scenario placement and dataset balancing [31].

Stage (iii) is the geometric core. The key challenge is that aerial scanners illuminate rooftops densely but facades sparsely, so naive surface reconstruction fails. Ground separation runs sequential multiplane RANSAC (iteratively fitting and removing planes) and selects the lowest elevation result as terrain, guaranteeing reliable isolation regardless of point density. Above ground points still contain trees, vegetation, and parked vehicles alongside building structures, which are separated before footprint extraction. *Trees and vegetation* are distinguished using a planarity filter on local point neighborhoods: building facades and rooftops form locally planar patches with high planarity, whereas unstructured canopy and shrub geometry yields low planarity values and is removed. *Vehicles* are filtered via a cluster size threshold on the remaining above ground points: cars, trucks, and similar objects form small, spatially compact clusters whose point counts fall well below those of building clusters, so a fixed lower bound on cluster size cleanly retains buildings and discards vehicles. Footprint extraction projects roof edge points to an axis aligned frame, fills a morphologically closed occupancy grid, and classifies shapes as rectangular (4 vertices) or L-shaped ($\geq$6 vertices). Facade panels are synthesized per footprint edge with column heights clamped against a KD tree lookup over roof points to prevent facade and roof penetration; an α-shape roof mesh and fan triangulated ground polygon complete the watertight building shell. The output is a coherent world package with HD maps, building meshes, props, and simulator ready scene files, enabling new geographic regions to be onboarded without manual rebuilding.

B. Grounding Scenario Generation with Real World Data

Synthetic scenarios decoupled from real failure statistics may overrepresent common interactions while missing the rare failure modes that dominate actual crash reports [11], [12], [25]. SimScene closes this gap through a four layer grounding architecture.

Safety Event Memory. Raw driving events, such as crashes, AV disengagements, and near miss reports [11], [32], are ingested and normalized through a progressive abstraction chain: *RawEvent* $\rightarrow$ *ScenarioPrimitive* $\rightarrow$ *ScenarioTemplate* $\rightarrow$ *MapScenarioIndex*. Each template captures interaction type, actor kinematics, and parameter ranges while discarding location specific coordinates, and is tagged with provenance and hotspot priors derived from the source corpus. This produces a reusable, evidence backed scenario library grounded in observed failure distributions. In practice, the ingested events serve as structured templates that inform downstream LLM assisted scenario authoring (Sec. III-A) along two complementary axes: *scenario families* grouped by interaction type and *scenario locations* derived from the map references attached to each event. From each ingested record, contextual hints such as the number of involved actors, prevailing weather and traffic conditions, and the geometry of the interaction (for example, the angle of a collision) are extracted and attached to the corresponding template, providing structured priors that the authoring layer subsequently consumes when composing a scenario.

Map Intelligence substrate. The HD map is augmented with a canonical scene graph encoding roads, lanes, junctions, crosswalks, curb zones, and drivability signals. Primitive feature detectors populate per region candidate anchors and pooled scenario zones, and third party enrichment (OpenStreetMap, agency feeds) is merged with confidence and freshness metadata. The result is a queryable world model against which scenario templates can be placed.

Scenario Grounding / MapRAG. A retrieval layer bridges user intent and the world model. Given a scenario request, expressed as natural language or a structured incident query, the layer extracts environment, actors, dynamics, and hard constraints, then executes a hybrid search over map zones, candidate anchors, and scenario templates. Deterministic geometric and semantic filters are applied first; the remaining shortlist is then scored to identify which locations on the

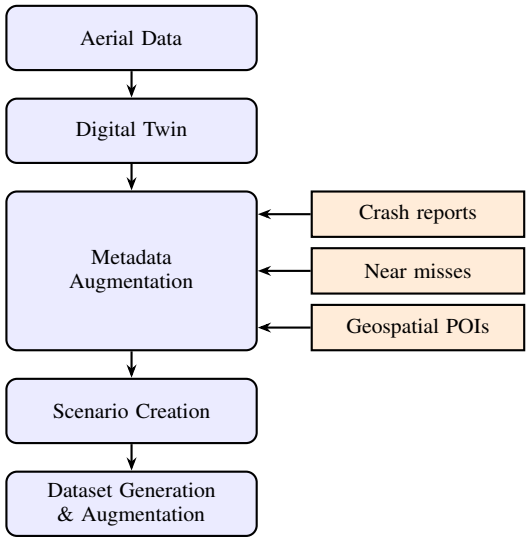


Fig. 1. SimCloud workflow: aerial assets and a digital twin (SimScene); metadata augmentation from crash reports, near misses, and geospatial POIs; scenario creation in simulation; dataset generation and augmentation governed by manifests.

map are most likely to host which types of scenarios. This scoring step is *not* a trained ML model; instead, accident and near miss datasets whose entries reference real map locations are ingested, and embeddings of each referenced location are extracted jointly with embeddings of its associated scenario type. These paired (location, scenario type) embeddings provide an evidence grounded signal: at query time, candidate map zones are ranked by their similarity to the embedding cluster of locations historically associated with the requested scenario type, yielding explainable candidate anchors backed by real incident evidence rather than learned parameters. The output is a grounded *ScenarioSpec* with actors anchored to road IDs, environment constraints resolved, ready for direct emission in SimCloud's road relative format (Sec. III-A).

Evaluation feedback loop. Retrieved and generated scenarios are evaluated against the Safety Event Memory in a shared scenario signature space: retrieval acceptance, simulation outcomes, and realism scores are compared against real world accident statistics. Reviewed outcomes feed back into template ranking and coverage calibration, closing the loop between synthetic generation and observed failure distributions.

III. SimCloud

SimCloud consumes simulation ready twins and HD maps from SimScene (Sec. II). Fig. 1 sketches the operational pipeline: geospatial imagery supports twin reconstruction; the twin is augmented with metadata derived from incidents and maps (crash and near miss abstractions, POI alignment) so scenarios can be grounded in real risk structure rather than ad hoc layouts. From that substrate, road relative scenarios are instantiated in simulation (e.g. CARLA) under manifests, then expanded into multimodal dataset families; the two subsections below detail authoring and environmental expansion.

A. Scenario Authoring

SimCloud decouples scenario logic from world specific coordinates through a road relative intermediate representation: each actor is anchored to a road ID, lane ID, and normalized longitudinal position. At simulation time, anchors resolve against the target twin's OpenDRIVE geometry, enabling a single authored scenario to be reinstantiated across different twins without coordinate rework. Actor behavior is encoded as declarative clip timelines selecting motion primitives with parameters. The realism of actor behavior at simulation time is delegated to mature simulator backends, principally CARLA and SUMO, which model vehicle and pedestrian dynamics, traffic flow, and signal logic; the road relative authoring layer issues motion primitives that these backends resolve into low level actor trajectories. LLM assisted authoring maps natural language descriptions to actors and behaviors constrained by map metadata [26], or supports interactive refinement through tool calls against road relative state. The currently ingested incident corpus is not yet of sufficient volume to train a dedicated, data driven behavior model on top of these simulator backends, and a planned next step is to use SimForge generated simulations in a closed loop to train such models. Each run is governed by a structured configuration manifest binding scenario, sensors, and environment conditions to outputs, ensuring reproducibility and coverage auditability.

B. Dataset Expansion

SimCloud expands each base scenario into a *family* of dataset realizations by sweeping exogenous factors encoded in each manifest, such as illumination and sun angle, atmospheric and precipitation state, seasonal material appearance, traffic and pedestrian density, and sensor pose, while freezing logical actors, road anchors, and motion programs. Parameters are assigned discrete levels or bounded intervals so that factorial or stratified designs yield reproducible coverage over the environmental axis of the generalization problem rather than ad hoc screenshots. Each realization is a separate job with its own manifest, emitting aligned multimodal outputs (RGB, depth, LiDAR, segmentation, trajectories, event logs) for traceable auditing and downstream training under declared distributional assumptions. Engine specific presets (e.g. CARLA weather and sun/sky bundles) and paired transfer or relighting stages (e.g. COSMOS Transfer [33]) are invoked only as backends: the contract to the dataset consumer is the manifest record, not a particular engine string. The generative transfer stage (COSMOS Transfer [33]) applies conditional relighting, material appearance, and weather and surface edits to simulator renders, narrowing the photometric and textural gap between synthetic and fleet captured imagery, a practical sim to real step for perception and prediction models trained on SimForge outputs. Pairing of source and transferred frames under each manifest preserves labels and calibration, so domain shift mitigation remains auditable alongside pure simulation runs.

IV. Results

A. Digital Twin Generation

Fig. 2 summarizes the SimScene LOD2 path in Sec. II-A. (a) Dense reconstruction from registered aerial and dashcam imagery yields a textured mesh over the site. (b) Geometric cleanup and ground separation suppress terrain and clutter so building clusters are isolated for reconstruction. (c) A single building aerial point cloud feeds the LOD2 routine (photogrammetry or fused LiDAR). (d) Footprint extraction recovers the base polygon and eave ring (the seam where vertical facades meet the roof) used to anchor panels in (e). (e) Facade calibration estimates outward facing panels and normals prior to meshing. (f) The classified point cloud separates roof, facade, and eave adjacent samples, reducing roof and facade penetration before (g) merging roof, facade, and ground into a watertight manifold mesh. The automatic LOD2 shell is followed by *selected* manual refinement: artists add higher LOD openings (doors, windows), props, and texture where simulation fidelity requires it. Concretely, manual human effort in the pipeline is currently concentrated in two stages: (1) promoting LOD2 shells to a higher LOD level by adding fine geometry such as windows and doors, and (2) texturing the resulting assets to achieve photometrically faithful appearance. All other stages, including photogrammetry, geometric cleanup, building isolation, LOD2 reconstruction, HD map extraction, procedural vegetation placement, and asset graph export, are automated; full automation of the remaining LOD enrichment and texturing stages is an explicit roadmap goal for SimScene.

Scaling to city wide reconstructions demands lightweight meshes. Dense surface methods such as Poisson reconstruction [34] typically yield on the order of 8–10 MB per building, prohibitive when a map tile aggregates hundreds of structures. The LOD2 pipeline (Sec. II-A) instead emits a parametric shell: planar facades, a piecewise roof, and a closed ground polygon.

Table I summarizes representative mesh file sizes. Simple rectangular footprints (4 vertices) compress to roughly 1.2 KB, while L-shaped or otherwise complex footprints ($\geq$6 vertices) remain in the 2–3 KB range, roughly three to four orders of magnitude smaller, keeping scene graphs tractable at scale.

TABLE I
LOD2 mesh sizes vs. dense Poisson reconstruction (per building).

Structure type	Poisson	LOD2 (ours)
Rectangular (4 vertices)	8–10 MB	$\sim$1.2 KB
L-shape / complex ($\geq$6 vertices)	8–23 MB	2–3 KB

Validated against terrestrial LiDAR over a Palo Alto road corridor, SimScene HD maps achieve $\sim$10 cm longitudinal MAE and sub 4 cm transverse MAE, sufficient for lane level scenario placement without survey grade infrastructure [27]. At that accuracy, scenarios anchored to lanes and junctions inherit real world spatial constraints at intersection scale rather than idealized placeholders. Representative twins have been imported into CARLA: closed loop runs with spawned actors and multimodal sensors (e.g. RGB and depth) are executed, confirming simulation readiness beyond offline mesh inspection.

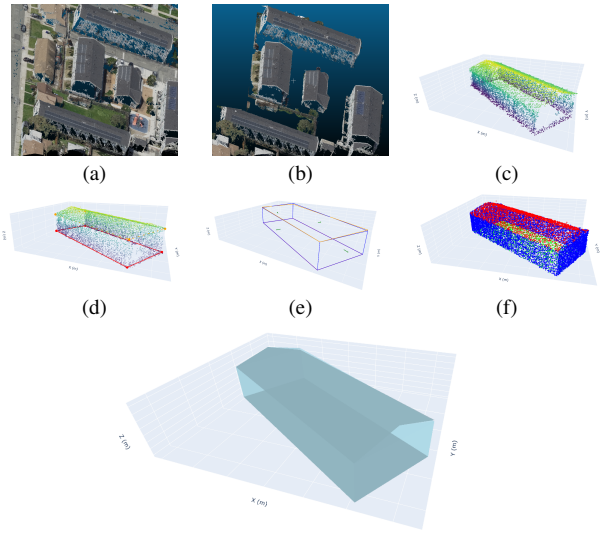


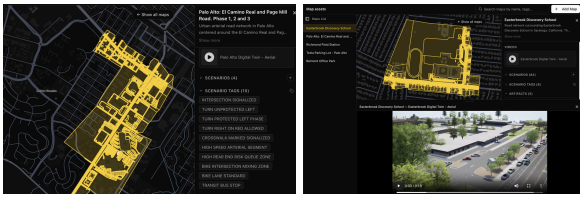
Fig. 2. SimScene digital twin generation (Sec. II-A). From site level photogrammetric mesh and denoised building isolation through footprint/eaves detection, facade calibration, labeled points, to a watertight LOD2 shell; optional manual pass adds fine geometry and textures.

Producing high fidelity, geometrically accurate simulation worlds is a central aim of SimScene, and the LiDAR comparison above is used as the primary fidelity reference for this objective. Since the comparison is grounded in physical LiDAR scans collected at the same location, the resulting accuracy figures are intended to be informative across paradigms: both end to end systems consuming rendered sensor streams and modular stacks whose perception, prediction, or planning components depend on lane level geometric correctness can be examined against the same metric reference. The present validation is limited to a single Palo Alto corridor, and broader assessment across additional locations and terrain types, including dense urban, suburban, highway, and varied topography, remains an open direction and is part of the planned fidelity evaluation effort for SimScene.

B. Grounding Scenario Generation

Fig. 3 shows maps augmented with metadata used for grounding (Sec. II-B). (a) On the Palo Alto El Camino Real twin, ten scenario tags, such as unprotected left turns, high rear end risk queue zones, bike and vehicle mixing, index conflict prone structure for map metadata alignment. (b) The map asset library indexes six deployed tiles; one tile (Easterbrook Discovery School) links 84 authored scenarios and eight tags to HD map geometry, illustrating how families grounded in incidents attach to candidate regions.

Limitation. While the grounding architecture is designed to align authored scenarios with real world failure distributions through the evaluation feedback loop (Sec. II-B), quantitative comparison numbers between generated scenario distributions and real accident statistics are not reported in this version of the paper. A full quantitative comparison, including category level



(a) Tags (b) Asset library

Fig. 3. Grounding: semantic tags on a twin (a) and per tile scenario/tag counts (b).

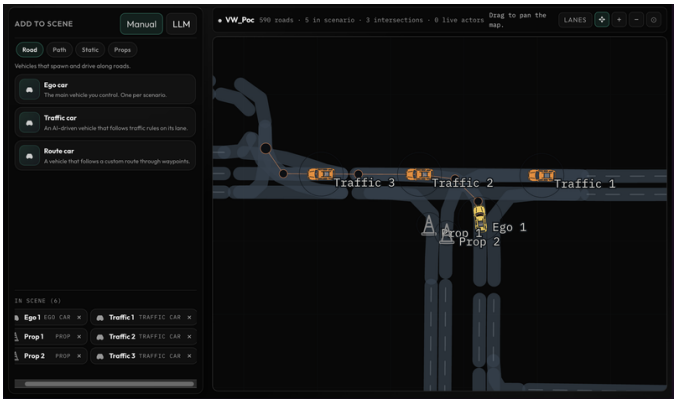


Fig. 4. Scenario authoring: lane anchored editor with actors anchored to road IDs prior to export governed by a manifest.

prevalence (rear end conflicts, unprotected turns, VRU exposure, etc.) and divergence metrics between the authored scenario distribution and ingested accident and near miss corpora, is deferred to future work alongside expansion of the Safety Event Memory ingestion pipeline.

C. Scenario Authoring

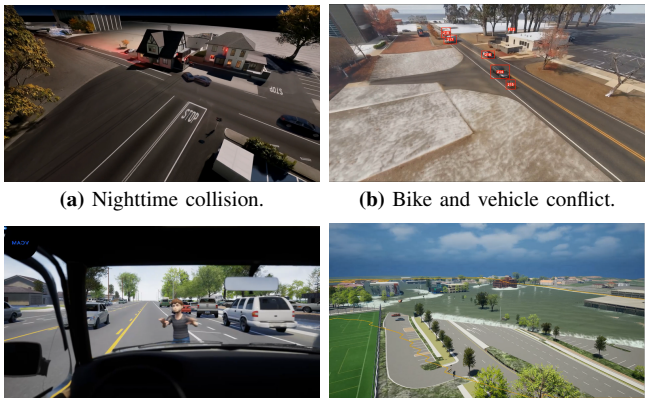
Fig. 4 shows the road relative authoring interface (Sec. III-A). The editor places ego, traffic, and props on OpenDRIVE lanes with Manual or LLM mode; actors are anchored to road IDs before export. For each run, config generation, simulation outputs, and COSMOS Transfer [33] emit artifacts under the same manifest, giving end to end traceability (e.g. Palo Alto parking lot scenario, CARLA). In practice, building one scenario takes approximately five minutes end to end, since the road relative authoring and LLM assisted scene construction are automated and only require lightweight human review of the resulting *ScenarioSpec* before manifest export.

D. Dataset Expansion

Under manifest governed sweeps (Sec. III-B), scenario logic and map geometry stay fixed while appearance level factors vary: paired illumination and sky state combinations in CARLA, complemented by lighting, surface, and weather bundles applied under generative transfer (COSMOS Transfer [33]) for sim to real alignment of RGB and paired modalities. Fig. 5 shows one authored scenario under varied viewpoints and environmental stresses (e.g. reduced sun elevation, snow cover), expanding sensing diversity without respecifying actors or topology.

Fig. 6 complements these sweeps with *interaction centric* stress tests spanning distinct failure families: (a) a low light

Fig. 5. Dataset expansion: environmental and viewpoint sweeps for one scenario (each panel a separate run governed by its manifest).



(a) Nighttime collision. **(b)** Bike and vehicle conflict.

(c) Jaywalking exposure. **(d)** Flooded surface.

Fig. 6. Dataset expansion: qualitative RGB samples spanning lighting, weather, and interaction adversity (each run under its own manifest).

forward collision, (b) a bicycle and vehicle conflict, (c) a partially occluded vulnerable road user entering the corridor, and (d) degraded tire coupling under standing water; each panel is a qualitative sample from an independent manifest, isolating interaction level adversity orthogonal to appearance decorrelation.

Intended application and ongoing validation. SimForge is intended for the generation of structured, multimodal driving datasets that benchmark open source autonomous driving models, covering both modular and end to end approaches. A quantitative benchmarking effort is in progress; detailed performance results on unseen real world and simulated environments are deferred to a forthcoming companion study. The present contribution is positioned as a reproducible, manifest governed dataset generation pipeline supporting such benchmarking across the research community.

Comparative evaluation against existing tools. A systematic comparison against existing simulation and scenario generation tools, including the CARLA Digital Twin Tool, OSM2World, and related dataset generation frameworks, is planned as part of the same study, with evaluation axes covering scenario diversity, photometric and geometric realism, and downstream task performance.

V. Conclusion

Looking ahead, a central goal is to construct a large scale, richly annotated metadata corpus of adversarial driving events spanning the full spectrum of safety critical failure modes: collision kinematics, vulnerable road user exposure, adverse weather and surface interactions, and multiagent conflict at ambiguous right of way boundaries. This corpus is intended to serve as a shared evaluation and training substrate for a broad class of downstream applications, including trajectory prediction under distribution shift, rare scenario fine tuning of deployed AV stacks, safety case validation for regulatory certification, and closed loop stress testing of modular and end to end controllers alike. By coupling SimForge's generative pipeline with this adversarial metadata layer, the objective is to provide the community with a principled, reproducible path from real world incident evidence to the distributional diversity that safe AV generalization demands. As part of this roadmap, quantitative alignment between generated scenario distributions and real accident and near miss statistics is planned for future reporting; such results are not included in the current submission and remain an explicit limitation of this work. Sweeps governed by manifests are designed to scale data volume superlinearly in the number of environmental and interaction axes: each additional authored scenario or tile compounds into large families of labeled runs, addressing the core "use more data" bottleneck without abandoning traceability. A public release of selected map tiles, scenario manifests, and expanded dataset families is scheduled for the end of June 2026, to support community driven evaluation and cross paradigm benchmarking aligned with open tooling in autonomous driving.

References

[1] C. Xu *et al.*, "A survey on end-to-end autonomous driving training from the data, strategy, and platform perspectives," TechRxiv, 2024, https://www.techrxiv.org/users/1002954/articles/1363381.

[2] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" RAND Corporation, Santa Monica, CA, USA, Tech. Rep. RR-1478-RC, 2016.

[3] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," 2017.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn. (CoRL)*, vol. 78, 2017, pp. 1–16.

[5] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," 2024.

[6] H. Shao, Y. Hu, L. Wang *et al.*, "LMDrive: Closed-loop end-to-end driving with large language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 15 120–15 130.

[7] ASAM e.V., "ASAM OpenDRIVE," Standard specification, version 1.8, 2023, https://www.asam.net/standards/detail/opendrive/.

[8] H. Caesar, V. Bankiti, A. H. Lang *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 11 621–11 631.

[9] S. Riedmaier, T. Ponn, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.

[10] U. Eichberger, S. Kock, and M. Aeberhard, "Scenario-based safety assessment for automated vehicles: A pathway toward practical deployment," *ATZ Worldwide*, vol. 122, no. 6, pp. 52–57, 2020.

[11] World Health Organization, "Global status report on road safety 2018," Geneva, Switzerland, 2018.

[12] National Highway Traffic Safety Administration, "Traffic safety facts: 2020 data," U.S. Department of Transportation, Tech. Rep. DOT HS 813 532, 2022.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 3354–3361.

[14] B. Wilson, W. Qi, T. Agarwal *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," 2023.

[15] J. You, X. Jia, Z. Zhang *et al.*, "Bench2drive-r: Turning real world data into reactive closed-loop autonomous driving benchmark by generative model," 2024.

[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robot.*, 2018, pp. 621–635.

[17] G. Rong, B. H. Shin, H. Tabatabaee *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," 2020.

[18] P. Cai, Y. Lee, Y. Lee *et al.*, "SUMMIT: A simulator for urban driving in massive mixed traffic," 2019.

[19] H. Zhang, W. Zhou, and B. Li, "MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning," 2022.

[20] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, 2023.

[21] M.-Y. Shen *et al.*, "Driveenv-nerf: A nerf-based autonomous driving environment," 2024.

[22] B. Mildenhall, P. P. Srinivasan, M. Tancik *et al.*, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 405–421.

[23] ASAM e.V., "ASAM OpenSCENARIO," Standard, https://www.asam.net/standards-detail/openx/openscenario/.

[24] D. J. Fremont, T. Dreossi, S. Ghosh *et al.*, "Scenic: A language for scenario specification and scene generation in simulation and testing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, 2019, pp. 1–8.

[25] Z. Deng *et al.*, "Semantic-guided fuzzing for autonomous driving testing," *J. Syst. Softw.*, vol. 215, p. 112314, 2024.

[26] Y. Wei, Z. Wang, Y. Lu *et al.*, "Editable scene simulation for autonomous driving via collaborative LLM-agents," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 15 077–15 087.

[27] F. Rottensteiner, G. Sohn, J. Jung *et al.*, "Results of the ISPRS benchmark on urban object classification, 3d building reconstruction and semantic labeling," in *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. II-3, 2014, pp. 293–298.

[28] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 5000–5009.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 5534–5543.

[30] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HDMapNet: An online HD map construction and evaluation framework," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022, pp. 4628–4634.

[31] A. Radford, J. W. Kim, C. Hallacy *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021, pp. 8748–8763.

[32] V. L. Neale, T. A. Dingus, S. G. Klauer *et al.*, "An overview of the 100-car naturalistic study and results," National Highway Traffic Safety Administration, U.S. Department of Transportation, Tech. Rep. DOT HS 810 593, 2005.

[33] NVIDIA Research, "Cosmos-transfer: Conditional world generation with multimodal control," 2025.

[34] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Symp. Geom. Process. (SGP)*, 2006, pp. 61–70.