LEARNING A STRUCTURED LATENT SPACE FOR VECTORIZED HD MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vectorized HD maps provide structured geometric and semantic descriptions essential for autonomous driving. As autonomous systems advance toward generative world models and simulation platforms, maps are increasingly required to function as learnable, generative interfaces rather than static offline assets. However, they naturally take the form of variable-length, heterogeneous, and unordered sets of geometric primitives, such as lane polylines and road polygons. Integrating such irregular vector collections into modern continuous generative models presents a fundamental structural challenge. Existing approaches typically struggle with a trade-off: imposing heuristic one-dimensional sorting on primitives induces semantic drift that collapses outputs into dataset-average straight corridors, whereas uncoordinated set diffusion fails to organize global scene layouts, yielding trivial, mutually parallel paths. To resolve this dilemma, we propose MapVAE, a hierarchical autoencoding framework that decouples local geometric modeling from global scene aggregation into a compact continuous latent space with canonical slot identities. In the first stage, a unified Vector Autoencoder projects heterogeneous primitives onto a shared hyperspherical manifold using canonical local coordinates, faithfully preserving fine-grained curvatures without category-specific backbones. In the second stage, a Map Autoencoder aggregates these element-level features into a fixed-length bottleneck via learnable queries, establishing structured slot representations that anchor global layout and directional diversity for downstream continuous generation. Crucially, distinct from existing works that decode raw geometries, we formulate the Map Autoencoder as a DETR-style set reconstruction directly in feature space, which unifies heterogeneous primitive recovery into a shared metric and improves optimization stability via unit-hypersphere constraints, a discriminative triplet loss, and spherical geodesic denoising. Extensive experiments demonstrate that MapVAE achieves superior reconstruction fidelity and enables our diffusion model, MapLDM, to synthesize diverse, geometrically precise HD maps across complex urban scenes.

1 INTRODUCTION

Vectorized HD maps provide structured geometric and semantic descriptions for autonomous systems, supporting perception, planning, and simulation Liao et al. (2023); Jiang et al. (2023); Ren et al. (2025). As autonomous driving moves toward generative world models and simulation platforms Sun et al. (2024), map representations must evolve. Beyond serving as static reference data, maps are increasingly required to function as compact, continuous, and learnable latent interfaces that can be integrated directly into generative models.

A persistent dilemma in map modeling is balancing fine-grained geometric precision with global structural consistency. Unlike images that reside on uniform pixel grids, HD maps naturally take the form of variable-length, heterogeneous, and unordered sets of geometric primitives, such as open lane polylines and closed road object polygons. Early rasterized bird's-eye-view representations Bansal et al. (2019) offer regular grid layouts suitable for standard convolutions, yet they introduce discretization errors and computational redundancy. Conversely, graph-based vectorized representations Gao et al. (2020) preserve exact geometries, but they lack a unified continuous manifold for generative modeling. Recent vectorized autoencoders, such as Scenario Dreamer Rowe et al. (2025), retain element-level vector fidelity by autoencoding individual road primitives independently. How-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ever, representing an entire road network simply as an unaligned set of tokens creates a challenge for continuous generative models like Diffusion Transformers Peebles & Xie (2023).

Specifically, current vectorized map generation faces a fundamental trade-off between local geometric accuracy and global layout coordination. On one hand, diffusion models operate most effectively on fixed-dimensional representations with stable slot semantics. To impose an ordering on variable-sized sets of primitives, existing pipelines often rely on heuristic one-dimensional sorting, such as ordering elements by spatial coordinates Rowe et al. (2025). In complex urban scenes, slight variations in layout density or coordinate bounds induce drastic index shifts. This semantic drift creates conflicting supervisory targets for the continuous denoiser, frequently collapsing outputs into dataset-average straight corridors. On the other hand, if one removes token ordering and diffuses primitives purely as an uncoordinated set, the model can synthesize local curves without sorting artifacts, but it loses global structural context. Without a scene-level representation to coordinate spatial relations, such models struggle to organize global layouts, frequently producing fragmented segments or trivial, mutually parallel paths with poor directional diversity.

These observations motivate the central question of this work: *can we learn a continuous latent representation that preserves native vector geometry while organizing the entire scene into a structured, globally coordinated latent space?* Our goal is to move beyond both fragile token sorting and uncoordinated set diffusion, providing a compact scene-level interface for continuous generation.

To this end, we propose MapVAE, a hierarchical autoencoding framework that decouples local geometric modeling from global scene aggregation. In the first stage, a unified Vector Autoencoder processes heterogeneous map elements into discriminative features on a shared unit hypersphere. By parameterizing each entity within its canonical local coordinate frame, this stage faithfully preserves high-frequency road curvatures without requiring category-specific sub-networks. In the second stage, a Map Autoencoder aggregates these element-level features into a fixed-length bottleneck via learnable queries. Collecting scene context through fixed query positions establishes structured slot identities, preventing both sorting-induced drift and layout degradation, thereby providing the downstream generator with stable reference roles to coordinate global layout and diversity.

Crucially, distinct from existing works that decode raw, unbounded coordinates, we formulate the scene decoder as a DETR-style set reconstruction directly in the unit hyperspherical feature space. This design unifies the recovery of diverse, heterogeneous primitives into a single metric and circumvents the ill-conditioned optimization of jointly regressing raw point sequences and semantic logits. To stabilize feature-domain bipartite matching, we combine unit-hypersphere constraints, a discriminative triplet loss, and a spherical geodesic denoising objective along tangent paths. Together, these components transform irregular map primitives into a compact continuous latent interface ready for generative modeling without losing detailed geometry.

Our contributions are summarized as follows: (1) We propose MapVAE, a hierarchical autoencoding framework that decouples local geometric modeling from global scene aggregation, compressing variable-length, heterogeneous map elements into a fixed-length continuous latent space; (2) We address the trade-off in vectorized map generation between sorting-induced collapse and uncoordinated set fragmentation, providing stable slot representations that anchor global layout and directional diversity for continuous diffusion models; (3) We design a unified element encoder that projects heterogeneous map entities into a shared hyperspherical manifold using canonical local coordinate frames, capturing fine-grained curvature without category-specific backbones; (4) We formulate map decoding as a DETR-style set reconstruction directly in feature space, achieving a unified recovery formulation across heterogeneous entities and ensuring training stability through unit-hypersphere constraints, a discriminative triplet loss, and spherical geodesic denoising. Extensive evaluations demonstrate that MapVAE achieves high reconstruction fidelity and enables our downstream diffusion model (MapLDM) to synthesize diverse, geometrically precise HD maps across complex urban scenes.

2 Related Work

Maps in Driving Models and Simulators. Across modern autonomous driving stacks and simulation environments, HD maps serve as the foundational geometric and semantic basis Hu et al. (2023); Jiang et al. (2023); Ren et al. (2025). Downstream modules rely on them to provide struc-

tured context fused into agent features Gao et al. (2020); Liang et al. (2020); Zhou et al. (2023), anchor geometric priors such as goal candidates and reference lines for motion planning Gu et al. (2021); Shi et al. (2022); Dauner et al. (2023), and condition background agent placement and sensor simulation Tan et al. (2021); Feng et al. (2023); Zhong et al. (2023); Wu et al. (2024); Gao et al. (2024). Yet across all these roles, maps are almost universally treated as static, pre-existing inputs rather than generative assets. Simulators traditionally replay logged sessions from recorded datasets Caesar et al. (2021); Gulino et al. (2023); Li et al. (2023), rely on small catalogues of hand-crafted virtual towns Dosovitskiy et al. (2017), or procedurally assemble layouts from rule-based grammars Li et al. (2022b), leaving learned generative mapping as an emerging frontier Mi et al. (2021); Sun et al. (2024); Chitta et al. (2024); Rowe et al. (2025). This reliance on static maps introduces a bottleneck: since agent interactions, behavioral policies, and sensor streams all branch conditionally from the road layout, the diversity of the map directly governs the behavioral coverage and generalization bounds of the entire autonomous stack. Addressing this constraint demands a shift toward generative map modeling, where novel, geometrically precise, and topologically consistent layouts can be synthesized on demand in standard vectorized formats.

Evolution of Map Representations. To enable effective map modeling, the underlying representation must balance geometric fidelity with structural regularity. Early pipelines rasterized map elements into multi-channel BEV images Bansal et al. (2019); Philion & Fidler (2020). While structurally regular and convenient for standard neural architectures, raster formats suffer from discretization errors, redundant computation over empty space, and the loss of discrete instance identity. Vectorized formulations overcome these limitations by modeling map elements as polyline subgraphs Gao et al. (2020) or structured lane graphs Liang et al. (2020). However, these approaches often rely on complex graph neural networks or handcrafted connectivity heuristics tailored separately to specific road categories. Furthermore, while online mapping methods resolve set unorderedness during perception via Hungarian matching Liao et al. (2023), this matching acts solely as an assignment strategy during training; the latent activations remain variable-sized query sets across scenes. Merely padding these sets achieves a uniform sequence length but fails to establish canonical semantic roles across tokens. Consequently, existing representations fail to provide a compact, coordinate-free latent interface that can support continuous generative models.

Structured Latents for Map Generation. Generative modeling of driving environments typically adopts a two-stage latent diffusion paradigm: an autoencoder first establishes a compact latent space, upon which a diffusion model learns the scene distribution Rombach et al. (2022); Peebles & Xie (2023). The inductive bias of this latent space fundamentally bounds generative quality. Unlike images that naturally reside on uniform pixel grids, vectorized maps possess no canonical continuous coordinate space, presenting a distinct representational dilemma. SLEDGE Chitta et al. (2024) projects vectorized lane graphs into low-resolution raster latent grids, performing diffusion over these regular cells before reconstructing polylines with a set decoder Zhu et al. (2023). While this ensures a fixed latent layout, global graph connectivity must be heuristically recovered post hoc Farag et al. (2026). Conversely, Scenario Dreamer Rowe et al. (2025) retains a fully vectorized representation by assigning an individual latent token to each map element. To apply standard sequence-based denoisers, it introduces a heuristic one-dimensional spatial ordering on element tokens. However, in complex urban environments, projectively sorting multi-branch topologies along a single axis induces severe index drift across scenes. This slot ambiguity disrupts continuous diffusion training, biasing the generator toward empirical training averages and producing repetitive straight roads. On the other hand, removing token ordering and diffusing map elements as a purely unordered set avoids sorting drift, yet it deprives the generative model of global structural coordination. Without a scene-level representation to anchor relative spatial relationships, purely unconstrained set diffusion produces fragmented topologies, missing junctions, and trivial parallel paths.

We address this representational gap through a hierarchical formulation. Instead of relying on fragile coordinate sorting or unconstrained element sets, our Map Autoencoder compresses variable-length map elements into a fixed set of canonical bottleneck tokens. Crucially, by shifting set reconstruction from raw unbounded coordinates to a shared hyperspherical feature space, this architecture unifies heterogeneous element recovery into a well-conditioned optimization target. This design decouples local geometric modeling from global scene aggregation, providing continuous diffusion models with stable slot identities to synthesize diverse, geometrically precise road layouts without layout collapse.

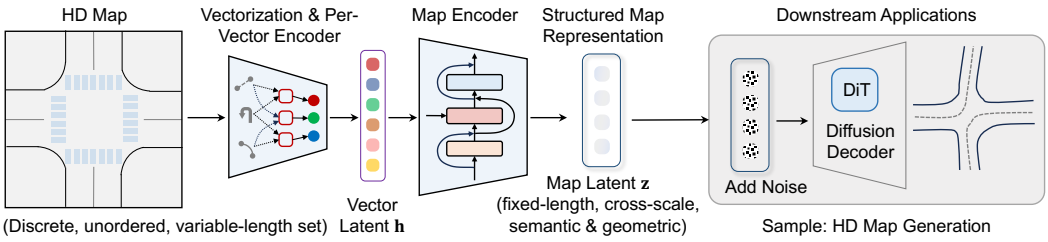

Figure 1: Overview of the MapVAE framework. A variable-length vectorized HD map is first encoded into compact element-level vector latents $\boldsymbol{h}$ on a shared hypersphere. A scene-level map encoder then aggregates these features into a fixed-length bottleneck $\boldsymbol{z}$ with canonical slot identities. Downstream, a latent DiT operates directly on this canonical space, generating diverse HD Maps.

3 METHODS

MapVAE learns a continuous latent space for vectorized HD maps. To bridge variable-length, unordered map elements with standard generative models, our framework compresses complex road layouts into a compact scene latent $\boldsymbol{z} \in \mathbb{R}^{K \times D}$ while preserving geometric detail (Fig. 1).

We train the model in three sequential stages, with each stage training on the frozen representations from the previous one. First, the Vector Autoencoder encodes individual map elements into normalized vector latents. Second, the Map Autoencoder aggregates these unordered latents into the global scene latent $\boldsymbol{z}$ using a Transformer bottleneck. Finally, we train a Diffusion Transformer with v-prediction on this frozen latent space to generate new maps conditioned on scene complexity.

3.1 CONCEPTS AND FRAMEWORK

Problem Definition. We represent a vectorized HD map as a set $\mathcal{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n\}$. Each element $\boldsymbol{m}_i$ contains an ordered sequence of waypoints $\boldsymbol{p}_i \in \mathbb{R}^{N \times 3}$ and a semantic label $\boldsymbol{c}_i \in \{0,1\}^C$. This label is formed by concatenating B attribute blocks, $\boldsymbol{c}_i = [\boldsymbol{c}_i^1 \parallel \cdots \parallel \boldsymbol{c}_i^B]$, where each block represents a mutually exclusive attribute and $\parallel$ denotes concatenation, so that $C = \sum_{b=1}^{B} C_b$ and $\|\boldsymbol{c}_i^b\|_1 = 1$. Our objective is to train an encoder f_ϕ that compresses $\mathcal{M}$ into a fixed-length representation $\boldsymbol{z} \in \mathbb{R}^{K \times D}$, alongside a decoder f_θ that reconstructs the scene $\hat{\mathcal{M}}$, where $\phi = \{\phi_1, \phi_2\}$ and $\theta = \{\theta_1, \theta_2\}$ collect the parameters of the two stages.

Hierarchical Reconstruction in Feature Space. Following the standard latent diffusion paradigm, we first train an autoencoder on a scene reconstruction task to establish the latent representation $\boldsymbol{z}$. However, directly reconstructing raw map entities from a compact scene bottleneck $\boldsymbol{z}$ poses two primary optimization challenges. First, accommodating a variable number of map instances requires bipartite matching during training, which is prone to assignment oscillations and training instability in unconstrained geometric spaces Carion et al. (2020); Liao et al. (2023). Second, jointly recovering continuous waypoints $\boldsymbol{p}$ and discrete categorical labels $\boldsymbol{c}$ forces the decoder to optimize an entangled, high-dimensional target. In our setup, for instance, each map polyline consists of $N = 20$ waypoints spanning $C = 40$ fine-grained classes (Table 2). Recovering raw point coordinates requires the decoder to satisfy strict sequential ordering constraints and maintain geometric continuity along the path, while simultaneously learning discriminative features for classification. To circumvent these difficulties, we decouple local geometric modeling from scene-level aggregation, adopting the hierarchical formulation illustrated in Fig. 2.

3.2 AUTOENCODERS

3.2.1 VECTOR AUTOENCODER (STAGE ONE)

The first stage establishes a unified feature space across individual map elements. To eliminate global spatial variance, we decouple each element's extrinsic pose from its intrinsic shape. Specifically, an element is parameterized by its spatial endpoints $\boldsymbol{u}_i \in \mathbb{R}^{2\times 3}$, an orientation vector $\boldsymbol{r}_i \in \mathbb{R}^2$, and a canonical point sequence $\tilde{\boldsymbol{p}}_i \in \mathbb{R}^{N\times 3}$ defined in its normalized local coordinate frame. The

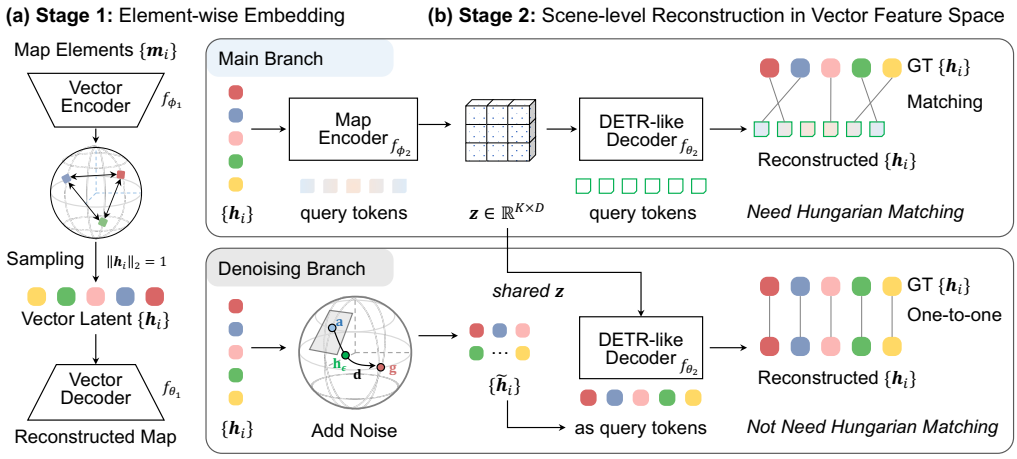

Figure 2: Hierarchical reconstruction framework for learning the map latent $\boldsymbol{z}$. Stage 1 translates raw maps into element-wise vector embeddings $\boldsymbol{h}$. Stage 2 performs scene-level reconstruction within this feature space via a DETR-like architecture enhanced by a denoising branch. Finally, it compresses discrete, variable-length elements into a unified fixed-length latent representation $\boldsymbol{z}$.

absolute point geometry is analytically recovered via $\boldsymbol{p}_i = \mathcal{T}(\boldsymbol{u}_i, \boldsymbol{r}_i, \tilde{\boldsymbol{p}}_i)$, as detailed in App. A.1. Ablation experiments in Table 4 indicate that this canonical decomposition lowers reconstruction errors for both open polylines (e.g., lane dividers) and closed polygons (e.g., road objects), while visibly improving the fidelity of reconstructed road curvature.

We initially fuse pose, shape, and semantic attributes into an element token $\boldsymbol{e}_i$ (App. A.2). Rather than using specialized graph neural networks, we contextualize these tokens using a standard Transformer encoder. Global self-attention naturally captures inter-element relational dependencies and layout structure, avoiding complex edge formulations while benefiting from highly optimized, hardware-accelerated primitives like flash attention Dao et al. (2022). The resulting representations are normalized to produce direction vectors $\bar{\boldsymbol{h}}_i \in \mathbb{S}^{d-1}$. To bound the target space and introduce reconstruction tolerance for Stage Two, we draw vector latents from a Power Spherical posterior De Cao & Aziz (2020), $\boldsymbol{h}_i \sim \mathcal{S}(\bar{\boldsymbol{h}}_i, \kappa)$, with concentration κ. An element-wise MLP then reconstructs the raw attributes from $\boldsymbol{h}_i$. This angular perturbation forces the decoder to recover geometries from local geodesic neighborhoods, providing a built-in error margin for downstream feature matching. As validated in Table 6, this spherical sampling introduces a crucial trade-off: while it incurs a marginal drop in standalone Stage 1 reconstruction fidelity, the resulting tolerance to downstream latent errors yields substantial AP gains in Stage 2.

Geometric and Semantic Objectives. We supervise the reconstruction on both the absolute geometry and its individual components:

$$\mathcal{L}^v_{geo} = \frac{1}{n}\sum_{i=1}^{n}\left(\|\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i\|_1 + \|\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i\|_1 + \|\boldsymbol{r}_i - \hat{\boldsymbol{r}}_i\|_1 + \|\tilde{\boldsymbol{p}}_i - \hat{\tilde{\boldsymbol{p}}}_i\|_1\right), \tag{1}$$

where variables with a hat ($\hat{\cdot}$) denote the predicted outputs from the decoder, and n is the total number of map elements in the scene. For semantics ($\mathcal{L}^v_{sem}$), we apply a block-wise softmax cross-entropy loss along with a binary indicator for road objects. During decoding, we take the argmax within each block to assign one class per attribute. To maintain structural accuracy, we add a curvature-weighted smoothness term $\mathcal{L}^v_{smo}$ that forces the model to pay more attention to curved road segments. Because road objects (like crosswalks) are closed polygons and lack a canonical starting point, we compute their loss over the optimal relabeling $\pi^\star_i$ of the vertex sequence. We also include a connectivity term $\mathcal{L}^v_{conn}$ that pulls predicted endpoints toward the midpoint of ground-truth lane junctions, ensuring the map topology makes sense. Full definitions are available in App. A.2.

Discriminative Manifold Objective. To prevent representation collapse and ensure that distinct entities remain separable during downstream matching, we regularize the hyperspherical latent space

with a triplet objective:

$$\mathcal{L}^{v}_{\text{triplet}} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \max\left(0,\, \delta(\boldsymbol{h}_i, \boldsymbol{h}_i^{+}) - \delta(\boldsymbol{h}_i, \boldsymbol{h}_j) + \alpha_m\right), \tag{2}$$

where $\delta(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|_2$ denotes the chordal distance on $\mathbb{S}^{d-1}$, and α_m specifies the target separation margin. For each anchor, we obtain the positive representation $\boldsymbol{h}_i^{+}$ by applying spatial jitter to the raw attributes $\boldsymbol{m}_i$, while treating all other co-occurring elements in the scene as negative samples $\boldsymbol{h}_j$. Table 7 ablates the margin α_m. A small α_m yields weak separation and ambiguous matching costs, while increasing it enforces distinct angular margins on $\mathbb{S}^{d-1}$, enhancing target discriminability and improving Stage Two AP. The final Stage one objective is $\mathcal{L}_1 = \sum_k \lambda_k \mathcal{L}^{v}_{k}$, aggregated as described in App. A.2.

3.2.2 Map Autoencoder (Stage Two)

In the second stage, we aggregate the vector latents $\mathcal{H} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n\}$ into a global bottleneck $\boldsymbol{z} \in \mathbb{R}^{K \times D}$. We introduce K learnable global tokens $\boldsymbol{Q}$ that attend to the set $\mathcal{H}$ via a Transformer. To parameterize the posterior, we explicitly decompose the latent mean into a radial magnitude $r \geq 0$ and a normalized direction:

$$\boldsymbol{\mu} = r \cdot \frac{\bar{\boldsymbol{\mu}}}{\|\bar{\boldsymbol{\mu}}\|_2}, \quad \boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{3}$$

During inference, we drop the stochastic noise and deterministically set $\boldsymbol{z} = \boldsymbol{\mu}$. Under the standard Gaussian prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, this polar decoupling reduces the mean penalty in the KL divergence $\mathcal{L}_{\text{KL}}$ directly to r^2. The KL objective thus serves as an intrinsic structural sparsity regularizer: it allows redundant query slots to be deactivated by collapsing $r \to 0$, while safeguarding the directional semantics of active tokens against metric distortion. App. B.3.2 shows an ablation of the polar parameterization, and App. A.3 gives more details of the detailed network and KL divergence derivations.

Anchor Initialized Queries and Feature Matching. The Map Decoder reconstructs the latents $\hat{\mathcal{H}}$ using q queries $\boldsymbol{Q}_d$. We initialize these queries from q anchors $\mathcal{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_q\}$, which form a data-driven covering of the hypersphere. We resolve the associations using the Hungarian algorithm, based on a cost function that combines feature distance and foreground confidence logits (App. A.3).

Spherical Geodesic Denoising. To mitigate the optimization instability of bipartite matching, we design a manifold-aware denoising auxiliary task Li et al. (2022a). For each target $\boldsymbol{h}_i$, we select a neighboring anchor $\boldsymbol{a}_i$ and synthesize a perturbed query within its tangent space:

$$\tilde{\boldsymbol{h}}_i = \Pi\left(\boldsymbol{a}_i + \lambda_i \boldsymbol{d}_i + \boldsymbol{\epsilon}_i^{\perp}\right), \quad \lambda_i \sim \mathcal{U}\left(0,\, \tfrac{1}{\boldsymbol{h}_i^{\top} \boldsymbol{a}_i}\right), \tag{4}$$

where $\boldsymbol{d}_i$ points along the geodesic toward $\boldsymbol{h}_i$, $\boldsymbol{\epsilon}_i^{\perp}$ introduces orthogonal tangent perturbation, and $\Pi(\boldsymbol{v}) = \boldsymbol{v}/\|\boldsymbol{v}\|_2$ projects the perturbed vector back onto $\mathbb{S}^{d-1}$. These perturbed latents $\tilde{\boldsymbol{h}}_i$ serve directly as decoder query tokens $\boldsymbol{Q}_d$. Unlike regular matching queries whose targets fluctuate dynamically during training, each $\tilde{\boldsymbol{h}}_i$ possesses a hard-wired, deterministic assignment: it is supervised to directly reconstruct its corresponding feature $\boldsymbol{h}_i$, bypassing Hungarian matching altogether. Conversely, negative queries sampled from distant anchors are trained to predict the background slot. We concatenate the denoising queries along the batch dimension to strictly prevent label leakage into the matching stream. Ablation results in Table 9 confirm that spherical denoising plays a pivotal role in boosting the overall reconstruction capability. The overall Stage Two objective is:

$$\mathcal{L}_2 = \lambda_{\text{feat}} \mathcal{L}^{s}_{\text{feat}} + \lambda_{\text{cls}} \mathcal{L}^{s}_{\text{cls}} + \lambda_{\text{kl}} \mathcal{L}_{\text{KL}} + \lambda_{\text{dn}} \mathcal{L}_{\text{dn}}, \tag{5}$$

with loss formulations detailed in App. A.3.

3.3 Latent Map Generation

With the continuous bottleneck $\boldsymbol{z} \in \mathbb{R}^{K \times D}$ established, generating variable-length maps readily translates into standard latent diffusion. We train a MapLDM based on DiT Peebles & Xie (2023) directly on standardized latents using the v-prediction parameterization Salimans & Ho (2022). To

explicitly control scene complexity, we embed the target counts of lane dividers and road objects alongside diffusion timesteps and inject them via adaLN. At inference time, the denoised latent is sequentially unrolled by the Stage Two slot decoder and Stage One vector decoder to recover the full vector map. Training hyperparameters and scheduler configurations are deferred to App. A.4.

4 EXPERIMENTS

We first analyze the latent manifold learned by MapVAE to confirm its structured properties, and then present map generation results built upon this representation. Comprehensive results regarding reconstruction fidelity and ablation experiments are provided in App. B.1 and B.3, respectively.

4.1 IMPLEMENTATION DETAILS

Dataset. To evaluate our method under realistic conditions, we use a large-scale HD map dataset spanning over 2,300 km^2 across multiple metropolitan areas with diverse urban layouts. The dataset comprises 149,428 training and 29,263 validation frames, each extracted as an expansive 500×500 m map tile sampled at 100-meter intervals along real driving trajectories. During training, we randomly crop 128×128 m regions from these large tiles (Fig. 4). Sourcing from an area substantially larger than the training window enables rich, boundary-free geometric augmentations via random translations and planar rotations, which prevents edge-truncation bias and discourages overfitting to artificial frame borders. Across all frames, the benchmark encompasses 57.7 million lane polylines and 16.0 million road objects, orders of magnitude larger than existing public benchmarks. It also exhibits high semantic granularity, covering 12 lane marking types, 5 paint colors, and 23 road object categories (see Fig. 5 and Table 2 for detailed taxonomy and statistics).

Experimental Setup. We benchmark our framework against Scenario Dreamer Rowe et al. (2025), which serves as the representative state-of-the-art baseline for our task. Scenario Dreamer models driving scenes by autoencoding road primitives into latent tokens and learning scene distributions with continuous diffusion. This two-stage formulation makes it structurally comparable to ours. For fair evaluation under identical computational budgets, we train Scenario Dreamer's autoencoder for 360k global steps and its generative model for approximately 200k global steps. Symmetrically, our Stage One (Vector AE) and Stage Two (Map AE) are trained for 360k steps, and our DiT generator is trained for 200k steps.

4.2 LATENT STRUCTURE OF MAPVAE

(a) Token sparsity. We analyze the latent geometry via the posterior mean tokens $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^D$ (D=24, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$), gated by polar radii $r_i = \|\boldsymbol{\mu}_i\|$. As shown in Figure 3(a), the latent matrix $\boldsymbol{\mu}$ exhibits pronounced sparsity induced by KL regularization, leaving redundant tokens near zero. **(b) Pairwise slot discrepancy.** To verify that the latent bottleneck establishes structured slot identities, we compute the pairwise RBF-MMD across all tokens over the validation set. As shown in Figure 3(b), the distance matrix exhibits pronounced off-diagonal blocks and cross-stripes. Rather than behaving as interchangeable, permutation-symmetric tokens, individual slots develop distinct marginal distributions. This emergent functional specialization provides the downstream generator with stable reference roles to coordinate multi-directional scene layouts, preventing both sorting-induced drift and trivial parallel corridor collapse. **(c) Directional separation.** Evaluating MMD on normalized directions $[\boldsymbol{\mu}_i/r_i, \boldsymbol{\mu}_j/r_j]$ in Figure 3(c) reveals that tokens remain widely separated on the unit sphere even without radial magnitude. This indicates that individual slots are anchored to distinct geometric subspaces. **(d) Capacity allocation.** The active token count $N_{\text{act}}(\tau) = \sum_{i=1}^{K} \mathbb{I}[r_i \geq \tau]$ increases smoothly with scene complexity (Figure 3(d)). Meanwhile, the effective dimensionality $D_{\text{eff}} = \exp(-\sum_{i=1}^{d} p_i \log p_i)$ stabilizes at $\sim$18 of 24 dimensions, showing that the model scales by recruiting more tokens rather than inflating per-token capacity. **(e) Intrinsic dimensionality.** We compute the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ of the second-moment matrix $\boldsymbol{S} = \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top \in \mathbb{R}^{D\times D}$, with normalized eigenvalues $p_i = \lambda_i / \sum_j \lambda_j$. As shown in Figure 3(e), the variance spectra remain remarkably consistent across scene density tiers, confirming that MapVAE maintains a stable internal latent geometry regardless of scene complexity. **(f) Per-token radius distribution.** Figure 3(f) reveals distinct radii distributions across different tokens:

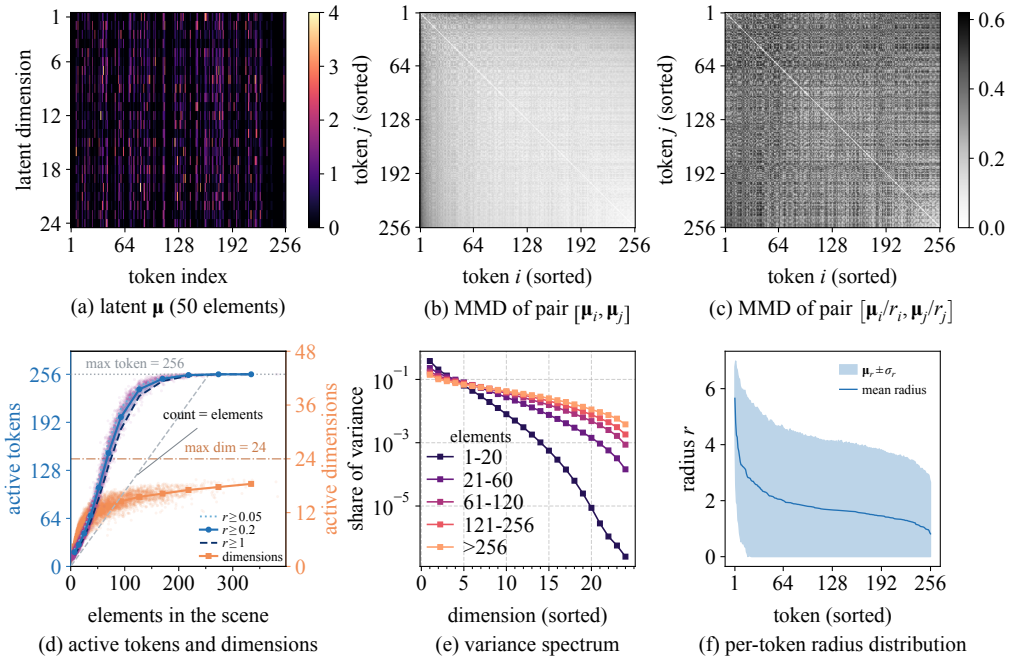

Figure 3: **Latent space analysis of MapVAE on the validation set.** (a) Latent matrix $\boldsymbol{\mu}$ for a scene with 50 elements, exhibiting marked token sparsity. (b, c) Pairwise RBF-MMD distance matrices across sorted tokens for unnormalized means $[\boldsymbol{\mu}_i, \boldsymbol{\mu}_j]$ and normalized directional vectors $[\boldsymbol{\mu}_i/r_i, \boldsymbol{\mu}_j/r_j]$. The structured divergence confirms emergent functional specialization across slots ($p(\boldsymbol{\mu}_i) \neq p(\boldsymbol{\mu}_j)$). (d) Both active token count and effective dimensionality scale smoothly with scene element counts. (e) Variance spectra across complexity tiers, showing stable intrinsic dimensionality. (f) Per-token radius distribution ($r = \|\boldsymbol{\mu}\|$), showing distinct radial profiles across tokens.

leading tokens reliably maintain large radius to anchor primary road structures, while trailing tokens decay gradually to provide adaptive capacity in denser scenes.

4.3 Evaluation of Map Generation

Evaluation metrics. We assess map generation across three complementary dimensions: lane geometry, scene layout, and scene diversity. For each dimension, we compute the Fréchet distance between generated and ground-truth distributions. Lane geometry evaluates lane lengths, maximum turning angles, and curvature distributions. Scene layout measures lane counts and spatial coverage. Scene diversity evaluates the range and entropy of lane directions, alongside relative angles between adjacent lanes. Road object counts are evaluated separately because Scenario Dreamer cannot synthesize objects.

Quantitative evaluation on map generation. Tab. 1 presents the distributional fidelity of generated HD maps on the validation split. *Comparison with the Baseline.* Compared to Scenario Dreamer Rowe et al. (2025), our framework achieves substantial gains across all metrics. Scenario Dreamer exhibits high Fréchet distances in geometry ($F_{\text{geo}} = 3.65$) and diversity ($F_{\text{div}} = 1.52$), alongside a lower spatial coverage of 0.14 ± 0.07 versus 0.20 ± 0.11 for the ground truth. This stems from its heuristic sorting, which suffers from severe semantic drift in complex scenes and biases samples toward repetitive straight roads. Moreover, our method accurately reproduces realistic object distributions (10.60 ± 13.55 versus 10.43 ± 13.85), a capability absent in Scenario Dreamer. *Impact of the Stage 2 Bottleneck.* Diffusing uncompressed Stage 1 vector tokens directly (DiT without Stage-2, 24 layers, 461 MB trainable parameters) achieves a competitive F_{geo} of 0.63, confirming that removing 1D coordinate sorting immediately liberates local curve generation. However, it underperforms in scene diversity ($F_{\text{div}} = 1.09$) and yields a collapsed nearest-neighbor angle of only 0.03 rad (Table 15), showing that pure set diffusion takes the shortcut of generating mutually parallel

Table 1: Quantitative comparison of lane geometry, directional diversity, and scene layout on validation split of the internal dataset. F_{geo}, F_{div}, and F_{lay} denote the corresponding Fréchet distances, where lower is better. The layout statistics are reported as mean $\pm$ standard deviation. Road-object statistics are excluded from F_{lay} because Scenario Dreamer does not generate road objects. Details of the statistics of geometry and diversity are given in Table 14 and Table 15.

Method	Parameters	Geometry	Diversity	Scene Layout			
		$F_{\text{geo}}\downarrow$	$F_{\text{div}}\downarrow$	$F_{\text{lay}}\downarrow$	Lanes	Coverage	Objects
Ground truth	–	–	–	–	48.80 ± 39.47	0.20 ± 0.11	10.43 ± 13.85
Scenario Dreamer (B)	377M	3.65	1.52	0.69	48.78 ± 39.37	0.14 ± 0.07	-
DiT without Stage-2	461M	**0.63**	1.09	0.28	48.85 ± 39.73	0.22 ± 0.09	10.38 ± 13.48
MapLDM (B)	385M	1.23	<u>0.72</u>	0.29	44.79 ± 31.92	0.18 ± 0.11	9.66 ± 12.25
MapLDM (L)	762M	1.21	<u>0.72</u>	<u>0.22</u>	46.02 ± 33.10	0.19 ± 0.11	10.15 ± 12.82
MapLDM (L), $K = 400$	762M	<u>1.18</u>	**0.71**	**0.13**	47.38 ± 35.85	0.19 ± 0.12	10.60 ± 13.55

lines without global coordination. In contrast, the Stage 2 Map AE compresses primitive sets into a unified scene latent $\boldsymbol{z}$, offloading macro-level layout coordination from the diffusion backbone. This lowers F_{div} to 0.72 and restores realistic directional arrangements. *Model Scalability and Bottleneck Capacity.* Scaling from the Base model (20 layers, 385M parameters) to the Large model (40 layers, 762M parameters) steadily improves layout fidelity, lowering F_{lay} from 0.29 to 0.22. Increasing the bottleneck token capacity to $K = 400$ achieves the best overall performance, reaching $F_{\text{geo}} = 1.18$, $F_{\text{div}} = 0.71$, and $F_{\text{lay}} = 0.13$. Furthermore, generated lane counts, spatial coverage, and object counts closely match the ground-truth distribution, validating the scalability of our hierarchical latent architecture. *Controllability.* When conditioned on target element counts, MapLDM faithfully regulates lane and object densities with minimal count deviation (Table 1), validating its effectiveness for steerable scene simulation.

Qualitative analysis on map generation. Fig. 9 illustrates visual comparisons across different setups. *Baseline behavior under complex scenes.* As shown in the first column, Scenario Dreamer primarily generates simple parallel corridors. Heuristic sorting functions adequately on small-scale patches with merged centerlines (e.g., nuPlan 64×64 m), but becomes unstable at our 128×128 m industrial scale where elements vary widely in length and relative position, biasing the generator toward dominant straight modes. *Role of the Stage 2 bottleneck.* Generating Stage 1 tokens directly as an unordered set (second column) removes sorting artifacts and introduces diverse orientations. However, without a scene-level bottleneck, the denoiser must independently handle both local geometry and scene-level layout from scratch, frequently producing fragmented lane segments and misplaced objects. *Diversity and scalability of MapVAE.* Our framework (third and fourth columns) addresses these issues by compressing irregular primitive sets into a compact latent $\boldsymbol{z}$. As a result, MapVAE reliably synthesizes realistic road layouts, including curved ramps, multi-way intersections, and multi-lane corridors. Scaling from the Base model to the Large model further refines lane continuity and object arrangements, confirming the effectiveness of our hierarchical representation.

5 Conclusion

We have presented MapVAE, a hierarchical framework that compresses heterogeneous, variable-length vectorized HD maps into a compact continuous latent space with canonical slot identities. By training diffusion models MapLDM over this structured bottleneck, our approach avoids fragile heuristic coordinate sorting and coordinates global scene layouts, reliably synthesizing diverse road environments. Despite these results, our current framework has several limitations. First, synthesized scenes occasionally omit minor lane segments in extremely dense intersections, suggesting that incorporating explicit relational graphs could further improve fine-grained local connectivity. Second, the generative stage adopts a standard DiT backbone, leaving ample room for specialized network architectures and further parameter scaling. Finally, the present model generates isolated map scenes without temporal continuity. Extending MapLDM to history-conditioned sequential rollout is a natural next step, as our compact, fixed-slot scene bottleneck provides a foundation for autoregressive modeling in long-horizon simulation and world model applications.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the manuscript's writing, grammar refinement, and assisting with code refactoring to improve formatting and adherence to coding standards. We have not used generative AI tools for core concept ideation, theoretical derivations, experimental design, or primary algorithm implementation, and data generation tasks are not applicable to this work. Additionally, we used generative AI tools for minor text readability enhancements and routine code formatting. We have reviewed all AI-assisted work. Specifically, all polished text was manually checked to ensure it accurately conveys our original technical meaning without hallucination or distortion, and all AI-suggested code modifications were strictly reviewed, verified, and unit-tested by the authors to guarantee functional correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We deeply value the open-source philosophy, as our work is built upon and inspired by the contributions of the open-source community. To ensure reproducibility during the review process, we have provided comprehensive implementation details, network architectures, loss formulations, and training hyperparameters in Section B and Appendix. We will make code available upon publication. Furthermore, we are also exploring the possibility of releasing a subset of the data, subject to applicable legal and privacy constraints.

REFERENCES

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science and Systems XV*, 2019.

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *European Conference on Computer Vision*, pp. 57–74. Springer, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pp. 1268–1281. PMLR, 2023.

Nicola De Cao and Wilker Aziz. The power spherical distribution. *arXiv preprint arXiv:2006.04437*, 2020.

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

Marwan Farag, Steffen Wäldele, and Yu Yao. Controlmap: Controllable high-definition map generation for traffic scenario simulation. *arXiv preprint arXiv:2606.15930*, 2026.

Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3567–3575, 2023. doi: 10.1109/ICRA48891.2023.10160296.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11525–11533, 2020.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, volume 2024, pp. 22841–22860, 2024.

Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15283–15292. IEEE, 2021.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36: 7730–7742, 2023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13619–13627, 2022a.

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

Quanyi Li, Zhenghao Mark Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36:3894–3920, 2023.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European conference on computer vision*, pp. 541–556. Springer, 2020.

Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *The Eleventh International Conference on Learning Representations*, 2023.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4227–4236, June 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pp. 194–210. Springer, 2020.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, et al. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models. *arXiv preprint arXiv:2506.09042*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17207–17218, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr-a: 1st place solution for 2022 waymo open dataset challenge–motion prediction. *arXiv preprint arXiv:2209.10033*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Shuo Sun, Zekai Gu, Tianchen Sun, Jiawei Sun, Chengran Yuan, Yuhang Han, Dongen Li, and Marcelo H Ang. Drivescenegen: Generating diverse and realistic driving scenarios from scratch. *IEEE Robotics and Automation Letters*, 9(8):7007–7014, 2024.

Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 892–901, June 2021.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. *Advances in Neural Information Processing Systems*, 37: 114048–114071, 2024.

Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 3560–3566. IEEE, 2023.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17863–17873, 2023.

Xiyue Zhu, Vlas Zyrianov, Zhijian Liu, and Shenlong Wang. Mapprior: Bird's-eye view map layout estimation with generative models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8194–8205. IEEE, 2023.

A METHOD DETAILS

This appendix expands on the components introduced in Sec. 3.

A.1 ELEMENT PARAMETERIZATION

We decompose each map element into a pose and a shape. The pose includes a starting point $\boldsymbol{q}_i$ and a heading θ_i (derived from the first two points of the sequence), which we encode as a normalized pair $\boldsymbol{r}_i = [\sin\theta_i, \cos\theta_i]$, so that $\theta_i = \operatorname{atan2}(r_{i,1}, r_{i,2})$. The shape is represented by a canonical sequence $\tilde{\boldsymbol{p}}_i \in \mathbb{R}^{N\times 3}$. We obtain this by translating the element so that $\boldsymbol{q}_i$ sits at the origin, and then rotating it by $-\theta_i$ in the ground plane (the vertical axis remains unchanged). The composition function $\mathcal{T}$ simply applies the inverse transformation:

$$\mathcal{T}(\boldsymbol{u}_i, \boldsymbol{r}_i, \tilde{\boldsymbol{p}}_i)_k = \boldsymbol{q}_i + \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{\boldsymbol{p}}_{i,k}, \qquad k = 1, \ldots, N, \tag{6}$$

where $\boldsymbol{q}_i$ is the first row of the endpoint pair $\boldsymbol{u}_i \in \mathbb{R}^{2\times 3}$, and k indexes the individual points in the sequence. The decoder predicts all three factors and applies L_2 normalization to the heading so that $\boldsymbol{r}_i$ remains on the unit circle. It also predicts and supervises the second endpoint; this helps constrain the element's overall length without explicitly feeding it into $\mathcal{T}$. Because the shape branch always views elements in this normalized frame, it only has to learn curvature shapes once, rather than re-learning them for every possible road orientation.

A.2 VECTOR AUTOENCODER

Vector encoder. Pose, shape, and semantic features are then fused into a single token $\boldsymbol{e}_i$:

$$\boldsymbol{e}_i = \text{MLP}_{fuse}([\,\text{MLP}_{pts}(\tilde{\boldsymbol{p}}_i) \parallel \text{MLP}_{sem}(\boldsymbol{c}_i) \parallel \text{MLP}_{pe}(\boldsymbol{u}_i, \boldsymbol{r}_i)\,]), \tag{7}$$

where MLP_{pts}, MLP_{sem}, and MLP_{pe} are multilayer perceptrons designed to extract features from the canonical points, semantics, and pose, respectively. These tokens pass through Transformer encoder layers to capture context between different elements.

Vector decoder. The vector decoder is an element-wise MLP that maps each latent token $\boldsymbol{h}_i$ back to the original geometric and semantic representations independently. It shares a common MLP backbone followed by dedicated lightweight prediction heads: (1) an endpoint head regressing spatial endpoints $\boldsymbol{u}_i \in \mathbb{R}^{2\times 3}$; (2) an orientation head predicting the 2D direction vector $\boldsymbol{r}_i \in \mathbb{R}^2$; (3) a geometric head recovering the canonical waypoint sequence $\tilde{\boldsymbol{p}}_i \in \mathbb{R}^{N\times 3}$; (4) a fine-grained semantic classifier outputting category distribution $\boldsymbol{c}_i$; and (5) a structural discriminator identifying the element type (open polyline vs. closed polygon). Because all prediction heads operate strictly per-token without cross-instance operations, the decoding process is highly lightweight and fully parallelizable.

Hyperspherical posterior. Let $\boldsymbol{g}_i$ be the Transformer encoder's output for element $\boldsymbol{m}_i$. We pass it through a residual MLP to reach the latent dimension d, then normalize the result to the posterior:

$$\boldsymbol{h}_i \sim \mathcal{S}(\bar{\boldsymbol{h}}_i, \kappa), \qquad \bar{\boldsymbol{h}}_i = \Pi\big(\text{MLP}_{lat}(\boldsymbol{g}_i)\big), \qquad \Pi(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}, \tag{8}$$

where $\mathcal{S}$ denotes the Power Spherical distribution. During training, we draw a reparameterized sample $\boldsymbol{h}_i \sim \mathcal{S}(\bar{\boldsymbol{h}}_i, \kappa)$, and at inference, we just use $\bar{\boldsymbol{h}}_i$. We fix the concentration κ as a hyperparameter, and we set it to $\kappa = 2000$ empirically.

Semantic supervision and decoding. Each of the B attribute blocks in $\boldsymbol{c}_i$ is supervised by its own softmax cross-entropy loss ($\mathcal{L}_{ce}$), ensuring classes only compete within their specific block. We also apply a binary cross-entropy loss ($\mathcal{L}_{bce}$) to the lane-versus-road-object indicator t_i:

$$\mathcal{L}^v_{sem} = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{b=1}^{B}\mathcal{L}_{ce}(\boldsymbol{c}_i^b, \hat{\boldsymbol{c}}_i^b) + \mathcal{L}_{bce}(t_i, \hat{t}_i)\right). \tag{9}$$

Decoding naturally follows this structure: we take the argmax within each block to output exactly one class per attribute.

Smoothness. The smoothness loss forces the model to match the first and second derivatives of the ground-truth point sequences. If Δ and Δ^2 are the difference operators and $c_{i,k} = \|\Delta^2 \boldsymbol{p}_{i,k}\|_2$ is a curvature proxy at vertex k, the term is written as:

$$\begin{aligned}\mathcal{L}^v_{smo} = &\frac{1}{n(N-1)} \sum_{i=1}^{n} \sum_{k=1}^{N-1} \big(1 + \alpha_c \, c_{i,\max(k-1,\,1)}\big) \big\|\Delta \hat{\boldsymbol{p}}_{i,k} - \Delta \boldsymbol{p}_{i,k}\big\|_1 \\ &+ \frac{1}{n(N-2)} \sum_{i=1}^{n} \sum_{k=1}^{N-2} \big(1 + \alpha_c \, c_{i,k}\big) \big\|\Delta^2 \hat{\boldsymbol{p}}_{i,k} - \Delta^2 \boldsymbol{p}_{i,k}\big\|_1,\end{aligned} \tag{10}$$

where $c_{i,k}$ is defined for $k = 1, \ldots, N-2$ and reused for the leading tangent. Because the weight $1 + \alpha_c c_{i,k}$ (where α_c is a scaling hyperparameter) scales with curvature, the model focuses its learning capacity on complex, curved segments rather than long, straight lines (which are far more common but easier to model).

Connectivity. Whenever two ground-truth endpoints are connected, the connectivity loss $\mathcal{L}^v_{conn}$ pulls the predicted endpoints toward the midpoint of the ground truth. This ensures that adjacent lanes physically connect after reconstruction, rendering their shared node as a single point.

Polygon invariance. Road objects are closed polygons, meaning they don't have a strict starting vertex or winding direction. We account for this by building a set $\mathcal{P}_i$ of $2N$ possible relabelings for $\boldsymbol{p}_i$ (its N cyclic shifts, plus the reversals of those shifts). We then supervise the model using the relabeling that best matches the current prediction:

$$\pi_i^\star = \arg\min_{\pi \in \mathcal{P}_i} \big\|\hat{\boldsymbol{p}}_i - \pi(\boldsymbol{p}_i)\big\|_1. \tag{11}$$

This chosen relabeling $\pi_i^\star$ dictates the target endpoints, heading, and canonical points, keeping all geometric losses perfectly aligned. During evaluation, we use this same invariant distance metric to match predicted and ground-truth polygons.

Triplet mining. To create a positive sample $\boldsymbol{h}_i^+$, we encode a copy of $\boldsymbol{m}_i$ after slightly jittering its coordinates with Gaussian noise. We keep this noise small so the model learns that minor spatial variations shouldn't change the underlying code. The negative samples are simply the other elements in the same scene. We penalize any pair that violates the margin and normalize the loss by the number of anchors. This ensures that densely packed scenes apply proportionally stronger repulsion, and the loss drops to zero once all elements are sufficiently separated. (We skip scenes containing fewer than two elements).

Objective aggregation. We compute the losses for lanes and road objects separately and weight them by their respective element counts. This prevents rare categories from being overwhelmed by more frequent ones within a batch.

A.3 Map Autoencoder

Map encoder. We equip the K global tokens $\boldsymbol{Q}$ with learnable positional embeddings, endowing each slot with a consistent functional identity across scenes. These tokens aggregate context from element latents $\mathcal{H}$ via cross-attention and interleave with self-attention. Three dedicated MLP heads predict the posterior parameters, allowing us to sample latents via the reparameterization trick:

$$r = \mathrm{MLP}_r(\boldsymbol{Q}), \quad \bar{\boldsymbol{\mu}} = \mathrm{MLP}_{\bar{\mu}}(\boldsymbol{Q}), \quad \log \boldsymbol{\sigma}^2 = \mathrm{MLP}_\sigma(\boldsymbol{Q}), \tag{12}$$

$$\boldsymbol{\mu} = r \cdot \frac{\bar{\boldsymbol{\mu}}}{\|\bar{\boldsymbol{\mu}}\|_2}, \quad \boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{13}$$

where $\odot$ denotes element-wise multiplication. Unlike standard VAE formulations that predict an unconstrained mean vector, we explicitly decompose $\boldsymbol{\mu}$ into a radial magnitude $r \geq 0$ and a normalized direction. Under the standard Gaussian prior $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, this polar decoupling reduces the mean penalty in the KL divergence directly to r^2. Consequently, the KL term acts as a structural sparsity regularizer: it allows redundant query slots to be smoothly deactivated by shrinking their scalar magnitude $r \to 0$, while preserving the geometric orientation of informative tokens without distorting their directional semantics.

Latent regularization. We regularize the posterior toward a standard normal prior using a Gaussian KL divergence. We sum this over the latent dimension and average it across tokens and scenes:

$$\mathcal{L}_{KL} = \mathbb{E}_{\mathcal{M}} \left[\frac{1}{K} \sum_{k=1}^{K} \left(-\tfrac{1}{2} \sum_{j=1}^{D} \left(1 + \log \boldsymbol{\sigma}_{kj}^2 - \boldsymbol{\mu}_{kj}^2 - \boldsymbol{\sigma}_{kj}^2\right) \right) \right], \tag{14}$$

where $\boldsymbol{\mu}_{kj}$ and $\boldsymbol{\sigma}_{kj}$ represent the components of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for the k-th token at dimension j.

Decoder and anchor initialization. The q anchors $\mathcal{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_q\}$ exist on the same unit hypersphere as the target features. We initialize them by running q-cluster K-means on the element latents $\mathcal{H}$ from the training set (using the frozen Stage one encoder). This creates a data-driven baseline covering the populated areas of the sphere. The queries are defined as $\boldsymbol{Q}_d = \text{MLP}_q(\boldsymbol{a}_i)$. The decoder reads the scene latent $\boldsymbol{z}$ as memory in cross-attention, using a per-token MLP to generate keys and values. The predicted features are then projected back onto the sphere, while a parallel head outputs a foreground probability logit s_q for each query. We ablate the effect of anchor initialization in Table 13.

Matching cost. The bipartite matching cost balances feature alignment and foreground classification confidence:

$$\mathcal{C}(q, g) = \frac{\lambda_{\text{feat}}}{d}\, \delta(\hat{\boldsymbol{h}}_q, \boldsymbol{h}_g) + \lambda_{\text{cls}}\, \Delta\mathcal{L}_{\text{FL}}(s_q), \tag{15}$$

where $\delta(\cdot)$ denotes the chordal distance, d is the latent dimension, and $\Delta\mathcal{L}_{\text{FL}}(s_q) = \mathcal{L}_{\text{FL}}(s_q, 1) - \mathcal{L}_{\text{FL}}(s_q, 0)$ represents the marginal cost under focal loss Lin et al. (2017). We solve this assignment via the Hungarian algorithm. Aligning the cost coefficients ($\lambda_{\text{feat}}, \lambda_{\text{cls}}$) with the downstream training objectives ensures that optimal assignments directly correspond to minimization targets.

Denoising branch. For each target $\boldsymbol{h}_i$, we uniformly select an anchor $\boldsymbol{a}_i$ from its top-k most similar anchors. This randomness prevents the model from lazily memorizing a fixed anchor-to-target mapping. We define $\boldsymbol{d}_i = \boldsymbol{h}_i - (\boldsymbol{h}_i^\top \boldsymbol{a}_i)\, \boldsymbol{a}_i$ as the tangent direction pointing to the target. By adding Gaussian noise $\boldsymbol{\epsilon}_i^\perp$ to the tangent plane of $\boldsymbol{a}_i$, we generate the positive sample: $\tilde{\boldsymbol{h}}_i = \Pi(\boldsymbol{a}_i + \lambda_i \boldsymbol{d}_i + \boldsymbol{\epsilon}_i^\perp)$. By drawing λ_i between 0 and $1/(\boldsymbol{h}_i^\top \boldsymbol{a}_i)$, we sweep along the geodesic arc, while the tangent noise thickens this path into a continuous tube.

To teach the branch how to handle empty space, we create negative samples. We reassign a fraction ρ of the slots to completely random anchors, perturb them with stronger tangent noise, do not interpolate them toward any target, and label them as background. Both the positive and negative denoising queries share the same projector network as the matching queries, generating both feature and foreground predictions.

We keep the matching and denoising branches completely separate by design. We replicate the latent memory and concatenate the denoising group with the matching group along the batch axis. As a result, the two sets never share an attention map, meaning we don't need complex masking to prevent data leakage.

Objective terms. The matching branch is trained using a feature loss on the assigned pairs and a focal loss on the foreground classification across all q queries:

$$\mathcal{L}_{feat}^{s} = \frac{1}{nd} \sum_{i=1}^{n} \left\| \boldsymbol{h}_i - \hat{\boldsymbol{h}}_{\mathcal{C}_i^\star} \right\|_2^2, \qquad \mathcal{L}_{cls}^{s} = \frac{1}{q} \sum_{i=1}^{q} \mathcal{F}\big(s_i, y_i\big), \tag{16}$$

where $\mathcal{C}_i^\star$ is the optimal assignment based on the cost matirx, y_i indicates if a query was matched, and $\mathcal{F}$ is the sigmoid focal loss. The denoising branch uses exactly the same setup: the feature loss applies to its positive slots, and the classification loss applies to all its valid slots:

$$\mathcal{L}_{dn} = \lambda_{feat}\, \frac{1}{|\mathcal{P}|d} \sum_{i \in \mathcal{P}} \left\| \boldsymbol{h}_i - \hat{\boldsymbol{h}}_i^{dn} \right\|_2^2 \;+\; \lambda_{cls}\, \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \mathcal{F}\big(s_i^{dn}, y_i^{dn}\big), \tag{17}$$

where $\mathcal{D}$ contains the valid denoising slots, $\mathcal{P} \subseteq \mathcal{D}$ represents the positive subset, and $\hat{\boldsymbol{h}}_i^{dn}$ is the decoder output for the denoising query built from $\tilde{\boldsymbol{h}}_i$. Since $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}$ reside on a unit hypersphere, minimizing the squared error naturally maximizes their cosine similarity.

A.4 LATENT DIFFUSION MODEL

Latent standardization. Prior to diffusion training, the latent bottleneck representation $\boldsymbol{z} \in \mathbb{R}^{K \times D}$ is standardized element-wise via:

$$\tilde{\boldsymbol{z}} = \frac{\boldsymbol{z} - m_z}{s_z}, \tag{18}$$

where m_z and s_z are global mean and standard deviation scalars pre-computed over the training split. This shift-and-scale normalization centers the latent distribution at zero with unit variance, aligning it with standard Gaussian initialization. During inference, we invert this mapping ($\boldsymbol{z} = \tilde{\boldsymbol{z}}_0 \cdot s_z + m_z$) before passing the latents to the decoders.

Diffusion target and formulation. During training, the clean diffusion state $\tilde{\boldsymbol{z}}_0$ is standardized from a posterior sample $\boldsymbol{z} = \boldsymbol{\mu}$. Owing to our polar decomposition $\boldsymbol{\mu} = r\frac{\bar{\boldsymbol{\mu}}}{\|\bar{\boldsymbol{\mu}}\|_2}$, inactive query slots undergo radial collapse ($r \to 0$), causing uninformative tokens to naturally cluster around zero with isotropic variance. This prevents latent drift and relieves the diffusion model from memorizing erratic, high-norm empty representations.

We adopt the velocity formulation (v-prediction) Salimans & Ho (2022). For a forward Gaussian perturbation $\tilde{\boldsymbol{z}}_t = \alpha_t \tilde{\boldsymbol{z}}_0 + \sigma_t \boldsymbol{\epsilon}$ with $\alpha_t^2 + \sigma_t^2 = 1$ under a cosine schedule, the target velocity vector is analytically defined as $\boldsymbol{v}_t \equiv \alpha_t \boldsymbol{\epsilon} - \sigma_t \tilde{\boldsymbol{z}}_0$. The Diffusion Transformer denoiser $\boldsymbol{v}_\theta$ is optimized via the mean squared error:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t, \tilde{\boldsymbol{z}}_0, \boldsymbol{\epsilon}} \left[\|\boldsymbol{v}_\theta(\tilde{\boldsymbol{z}}_t, t, \boldsymbol{c}) - \boldsymbol{v}_t\|_2^2 \right], \tag{19}$$

where $\boldsymbol{c}$ denotes the conditioning descriptor. Compared to standard noise prediction (ϵ-prediction), v-parameterization avoids sample instability in high-noise regimes and prevents numerical collapse.

Architecture and complexity conditioning. Our denoiser is built upon standard Diffusion Transformer blocks Peebles & Xie (2023). Since the K slots in Stage Two are already equipped with consistent functional identities through the slot queries, we omit 1D token-axis positional embeddings.

To explicitly steer layout density, we construct the condition tuple as $\boldsymbol{c} = [N_{\text{lane}}, N_{\text{obj}}]$, representing the explicit counts of lane polylines and road objects, respectively. Both scalars are mapped through sinusoidal position embeddings followed by a two-layer MLP. The resulting complexity representation is concatenated with the diffusion timestep embedding and injected into every DiT layer via adaptive layer normalization (adaLN).

To enable Classifier-Free Guidance (CFG) Ho & Salimans (2021), we independently drop the condition vector $\boldsymbol{c}$ with an unconditional dropout probability $p_{\text{uncond}} = 0.1$ during training, replacing it with a learnable null embedding $\varnothing$. During sampling, the extrapolated velocity prediction is given by:

$$\hat{\boldsymbol{v}}_\theta(\tilde{\boldsymbol{z}}_t, t, \boldsymbol{c}) = \boldsymbol{v}_\theta(\tilde{\boldsymbol{z}}_t, t, \varnothing) + w \left(\boldsymbol{v}_\theta(\tilde{\boldsymbol{z}}_t, t, \boldsymbol{c}) - \boldsymbol{v}_\theta(\tilde{\boldsymbol{z}}_t, t, \varnothing) \right), \tag{20}$$

where $w \geq 1.0$ controls the guidance intensity.

Sampling and hierarchical decoding. At test time, generation starts from standard Gaussian noise $\tilde{\boldsymbol{z}}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We integrate the reverse diffusion trajectory using a DDIM sampler Song et al. (2021) under velocity parameterization to synthesize the clean latent $\tilde{\boldsymbol{z}}_0$, which is then unstandardized to $\boldsymbol{z}_0$.

The reconstruction unfolds hierarchically:

1. **Slot decoding (Stage Two).** The global latent $\boldsymbol{z}_0$ is fed into the Stage Two decoder to predict q candidate element latents paired with foreground probabilities, $\{(\hat{\boldsymbol{h}}_i, s_i)\}_{i=1}^{q}$. We filter valid elements by thresholding the classification logits, retaining slots where $\text{sigmoid}(s_q) > \tau_{\text{fg}}$. This step dynamically materializes the variable-length structure from the fixed-size bottleneck.
2. **Vector reconstruction (Stage One).** The surviving feature tokens $\hat{\boldsymbol{h}}_i \in \mathbb{S}^{d-1}$ are passed into the frozen Stage One vector decoder. The dedicated lightweight heads independently recover spatial endpoints $\boldsymbol{u}_i$, direction vectors $\boldsymbol{r}_i$, canonical waypoints $\tilde{\boldsymbol{p}}_i$, and semantic categories $\boldsymbol{c}_i$.

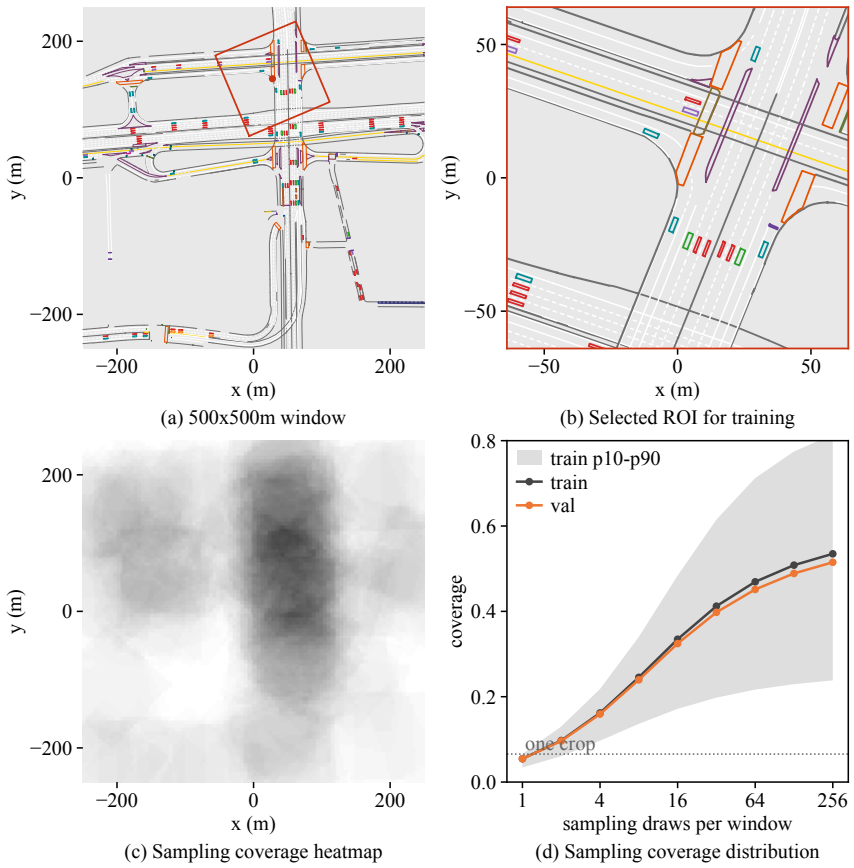

Figure 4: Training-sample construction on our private dataset. (a) A raw $500 \times 500\,\mathrm{m}$ map window; the red box marks a selected crop. (b) The corresponding ROI actually fed to the model, randomly translated and rotated ($p_{\mathrm{rot}} = 0.9$) from the window. (c) Heatmap of sampling coverage accumulated over many random draws from the window. Denser regions are sampled far more often than the sparse periphery. (d) Fraction of the window covered as a function of the number of sampling draws per window ($\log_2$ axis) for train and validation; a single crop covers only a small fraction, and coverage grows steadily but sub-linearly as draws increase.

3. **Global realization.** Canonical geometries are transformed into absolute world coordinates via their predicted SE(2) reference frames $\mathcal{T}_i(\boldsymbol{u}_i, \boldsymbol{r}_i)$, yielding complete, structured vector HD maps.

B Experiments

B.1 Evaluation of reconstruction

Evaluation metrics. We evaluate vector map reconstruction across geometric fidelity, semantic correctness, and scene-level retrieval accuracy. For reconstructed elements, we report permutation-aligned RMSE (in meters) on the 2D ground plane along with attribute classification accuracy (ACC). Because standard RMSE is heavily dominated by prevalent straight segments, we introduce Curvature Fidelity (CF) to explicitly evaluate how well non-linear geometry is recovered relative to a linear chord baseline (CF $= 1$ indicates perfect recovery). Furthermore, to evaluate scene-level reconstruction where the Map Autoencoder decodes variable numbers of queries without fixed one-to-one ground-truth correspondence, we follow standard HD map detection protocols Liao et al. (2023) to report Precision, Recall, and Average Precision (AP) averaged over distance thresholds $\tau \in \{0.5, 1.0, 1.5\}$ m. Complete mathematical formulations are provided in App. B.2.

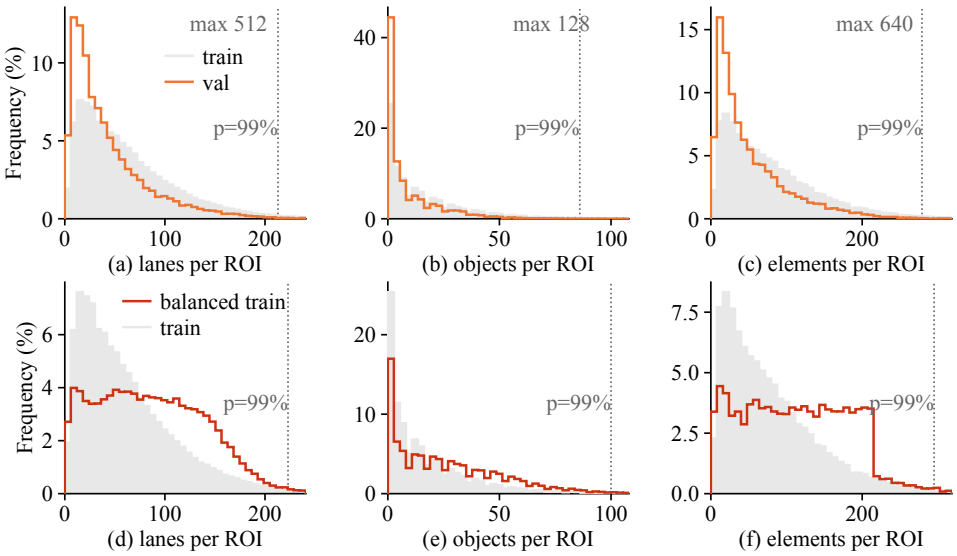

Figure 5: Element counts of a private training sample. Every sample is a 128 m region of interest (ROI) cropped from a 500×500 m source window, one crop per window. Columns report lanes, road objects and their sum, respectively. (a)–(c) show plain random cropping over 60,000 randomly sampled training windows (blue) and all 29,263 validation windows (orange). (d)–(f) show count-balanced resampling of the same 60,000 training source windows: 24 candidate crops are drawn per window and one is selected with a weight favouring globally rare total element counts (dark red); the original train distribution is repeated with the same blue outline and fill used in the upper row for direct comparison. Balanced cropping is train-only, so no validation curve is shown in the lower row. Each dotted line marks the primary distribution's own 99th percentile.

Quantitative results. Tab. 3 summarizes the quantitative reconstruction performance against the baseline. Our framework consistently achieves substantially lower geometric RMSE on lane polylines, demonstrating superior point-wise reconstruction accuracy. More importantly, while Scenario Dreamer often suffers from geometric flattening along sharp turns, our method yields significantly higher Curvature Fidelity, indicating that our vector latent effectively captures intricate non-linear profiles without over-straightening artifacts. Furthermore, our unified architecture accomplishes this high precision while simultaneously reconstructing multi-class road objects, a capability fundamentally absent in Scenario Dreamer.

Qualitative results. Figure 7 and Figure 8 illustrate the qualitative reconstruction comparisons between the baselines and our method on the internal dataset. Specifically, Figure 7 presents scenes across varying levels of complexity, while Figure 8 highlights challenging scenarios featuring complex road objects. As observed, both our Stage 1 and Stage 2 models achieve high reconstruction fidelity, faithfully recovering lane polylines as well as road objects, whereas Scenario Dreamer is restricted to reconstructing lanes alone. In comparison, our approach demonstrates noticeable advantages in curvature smoothness and lane graph topological connectivity. Furthermore, because Stage 1 performs one-to-one autoencoding without relying on Hungarian bipartite matching, it reconstructs map elements more exhaustively without omissions, whereas Stage 2 exhibits occasional minor dropouts under highly complex traffic layouts.

B.2 Detailed Metrics

We adhere to the unified mathematical notation established in Sec. 3. A vector scene comprises ground-truth elements $\mathcal{M} = \{\boldsymbol{m}_i\}_{i=1}^{n}$, where each primitive $\boldsymbol{m}_i$ is parameterized by a sampled polyline $\boldsymbol{p}_i \in \mathbb{R}^{N\times 3}$ and categorical attribute vectors $\boldsymbol{c}_i = [\boldsymbol{c}_i^1, \ldots, \boldsymbol{c}_i^B]$ across B semantic fields (e.g., line style, paint color, and object category). We denote predicted quantities with $\hat{\cdot}$. Although our representations and autoencoders preserve full 3D coordinates, all metric evaluations are performed on the horizontal ground plane by projecting points onto their 2D Cartesian coordinates

Table 2: Semantic classes of the private dataset and their instance counts in the training and validation splits. Lanes carry a line-type and a color attribute, objects a single type; each is a separate one-hot block of the element descriptor. The virtual line is removed from the data but keeps its one-hot slot, so it is listed with a zero count. Counts are over every window ($500 \times 500\,\mathrm{m}$).

Attribute	ID	Class	Train	Validation
Lane line type	0	single solid line	15,349,771	2,645,351
	1	single dashed line	11,008,809	2,048,844
	2	broad dashed (entry/exit) line	419,739	50,873
	3	double solid line	1,056,216	221,305
	4	double dashed line	182,406	32,755
	5	left solid right dashed line	132,520	23,386
	6	left dashed right solid line	270,960	54,683
	7	chevron (gore area)	1,385,090	231,178
	8	virtual line†	0	0
	9	guardrail / barrier	5,807,630	790,944
	10	wall / curb boundary	13,357,547	2,607,281
	11	normal line (other)	1,131	165
		Total	48,971,819	8,706,765
Lane color	0	white	25,857,256	4,478,365
	1	yellow	4,249,623	879,153
	2	left white right yellow	59,261	5,300
	3	left yellow right white	201,893	38,402
	4	unspecified / other	18,603,786	3,305,545
		Total	48,971,819	8,706,765
Object type	0	straight arrow	3,040,583	441,149
	1	straight-left arrow	333,968	70,967
	2	straight-right arrow	565,389	99,089
	3	straight-U-turn arrow	26,965	5,196
	4	left-turn arrow	653,646	120,311
	5	left-U-turn arrow	46,504	4,916
	6	merge-left arrow	109,604	30,620
	7	right-turn arrow	652,856	123,349
	8	U-turn arrow	68,109	10,081
	9	parking slot	1,177,390	268,418
	10	stop line	948,531	169,357
	11	refuge island	85,988	8,711
	12	box junction	1,236	225
	13	gore area	590,320	98,119
	14	green belt	52,598	7,819
	15	median strip	25,571	5,881
	16	arrow (other)	329,177	31,263
	17	marking (other)	365,685	68,064
	18	text marking	2,241,813	338,800
	19	symbol	2,720	975
	20	pedestrian signal	3,465	1,535
	21	vehicle signal	1,254,886	226,622
	22	crosswalk	1,059,604	194,803
		Total	13,636,608	2,326,270

$\boldsymbol{p}_{i,j}^{\mathrm{2D}} \in \mathbb{R}^2$ ($j \in \{1, \ldots, N_p\}$) in meters, strictly following established 2D benchmarks. Lanes and road objects are evaluated independently.

Table 3: Comparison of reconstruction performance.

Model	Lane			Road Object	
	RMSE/m (↓)	CF (↑)	AP(%, stage 2)	RMSE/m (↓)	AP(%, stage 2)
Scenario Dreamer	0.462	0.593	-	-	-
Ours	**0.180**	**0.885**	**94.04**	**0.366**	**83.44**

Geometric RMSE. For each reconstructed element, geometric fidelity is quantified by the root-mean-squared distance under canonical geometric alignment:

$$e_i = \sqrt{\frac{1}{N_p}\sum_{j=1}^{N_p}\left\|\hat{\boldsymbol{p}}_{i,j}^{\mathrm{2D}} - \boldsymbol{p}_{i,\pi_i^\star(j)}^{\mathrm{2D}}\right\|^2}, \qquad \mathrm{RMSE} = \frac{1}{n}\sum_{i=1}^{n} e_i, \tag{21}$$

where $\pi_i^\star \in \Pi_{\mathrm{can}}$ denotes the optimal index permutation determined during training (Sec. A.2), ensuring invariance to traversal direction for open polylines and cyclic shifts/reflections for closed bounding polygons. We report *Lane RMSE* and *Object RMSE* by averaging e_i within their respective cohorts.

Attribute Accuracy. For each discrete attribute head $b \in \{1, \ldots, B\}$, classification performance is computed via exact top-1 match:

$$\mathrm{ACC}^b = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[\arg\max \hat{\boldsymbol{c}}_i^{\,b} = \arg\max \boldsymbol{c}_i^{\,b}\right], \tag{22}$$

where $\mathbb{I}[\cdot]$ is the indicator function. In Tab. 8, *Type Accuracy* denotes the macro-average across primary lane markings and road-object classes.

Curvature Fidelity (CF). Standard RMSE is dominated by straight stretches and remains largely insensitive to over-straightening artifacts along sharp turns. To explicitly measure the preservation of non-linear geometry, we define a chord baseline connecting the endpoints of the canonicalized ground-truth polyline:

$$\bar{\boldsymbol{p}}_{i,j} = \boldsymbol{p}_{i,\pi_i^\star(1)}^{\mathrm{2D}} + \frac{j-1}{N_p - 1}\left(\boldsymbol{p}_{i,\pi_i^\star(N_p)}^{\mathrm{2D}} - \boldsymbol{p}_{i,\pi_i^\star(1)}^{\mathrm{2D}}\right), \quad j = 1, \ldots, N_p. \tag{23}$$

The residual vector $\bar{\boldsymbol{p}}_{i,j} - \boldsymbol{p}_{i,\pi_i^\star(j)}^{\mathrm{2D}}$ captures the ground-truth geometric sagitta (departure from linearity). Curvature Fidelity quantifies the proportion of this true curvature profile successfully reconstructed by the model:

$$\mathrm{CF} = 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{N_p}\|\hat{\boldsymbol{p}}_{i,j}^{\mathrm{2D}} - \boldsymbol{p}_{i,\pi_i^\star(j)}^{\mathrm{2D}}\|^2}{\sum_{i=1}^{n}\sum_{j=1}^{N_p}\|\bar{\boldsymbol{p}}_{i,j} - \boldsymbol{p}_{i,\pi_i^\star(j)}^{\mathrm{2D}}\|^2}. \tag{24}$$

By computing this ratio globally across all instances, near-straight segments do not dilute the contribution of genuinely curved structures. A score of $\mathrm{CF} = 1$ indicates perfect geometric recovery, $\mathrm{CF} = 0$ corresponds to trivial linear interpolation, and $\mathrm{CF} < 0$ implies the prediction degrades geometric fidelity beyond a naive straight chord.

Detection Metrics: Precision, Recall, and AP. The Map AE reconstructs complete scenes using Q set queries filtered by predicted existential scores $s_q \in [0, 1]$. Because the cardinality of predictions and ground-truth elements varies dynamically across frames, evaluating solely on matched pairs introduces selection bias (rewarding degenerate models that predict only a few confident elements). We therefore evaluate the scene-level decoder following standard HD map detection protocols Liao et al. (2023).

In each scene, predictions are sorted descending by confidence s_q and greedily matched to the nearest unassigned ground truth using the permutation-aligned distance e_i. A prediction is deemed a True Positive (TP) at threshold τ if $e_i \leq \tau$ meters and all predicted attribute classes strictly match the ground truth; otherwise, it is classified as a False Positive (FP). Missed ground-truth elements are counted as False Negatives (FN).

We report *Precision* and *Recall* at the fixed inference cutoff $s_q \geq 0.6$. To evaluate holistic trade-offs across confidence profiles, we compute Average Precision (AP_τ) as the area under the precision-recall curve:

$$\mathrm{AP}_\tau = \int_0^1 \max_{R' \geq R} P_\tau(R') \, \mathrm{d}R. \tag{25}$$

Following MapTR Liao et al. (2023), all reported metrics represent the mean across distance thresholds $\tau \in \{0.5, 1.0, 1.5\}$ m.

B.3 Ablative Studies

B.3.1 Vector Autoencoder

To understand how each design choice influences stage 1, we ablate key components of our Vector Autoencoder, focusing on geometric precision (lane point-wise RMSE and road object positional RMSE), high-frequency detail preservation (curvature fidelity), and Stage 2 reconstruction Average Precision (AP). Detailed definitions of these evaluation metrics are provided in Appendix B.2. Since attribute classification accuracies consistently remain high their full breakdowns are deferred to the Table 8 to keep our geometric discussion concise.

Coordinate formulation: absolute vs. relative. Comparing our decoupled relative representation against standard absolute coordinates (Table 4) demonstrates distinct advantages in both point-wise accuracy and curvature fidelity. This confirms the benefit of local canonicalization: mapping elements into a standardized local frame relieves the network from fitting unbounded global coordinates, allowing it to focus capacity on recovering intricate geometric curves.

Table 4: Ablation on coordinate input formulation.

Input Mode	Lane		Road Object
	RMSE/m (↓)	CF (↑)	RMSE/m (↓)
Absolute	0.295	0.688	0.535
Relative	**0.180**	**0.885**	**0.366**

Vector latent dimension. We examine the capacity bottleneck across latent dimensions $d \in \{16, 20, 24\}$ (default $d = 24$). As reported in Table 5, shrinking the latent space steadily harms point-wise reconstruction, with a sharp decline in curvature fidelity at $d = 16$. This trade-off reflects the natural behavior of information bottlenecks: when capacity is overly tight, the encoder preserves coarse low-frequency positions first, inevitably sacrificing subtle high-frequency curvature cues.

Table 5: Ablation on vector latent dimension d.

Hidden Dim	Lane		Road Object
	RMSE/m (↓)	CF (↑)	RMSE/m (↓)
16	0.320	0.353	0.415
20	0.233	0.834	0.426
24	**0.180**	**0.885**	**0.366**

vMF sampling. While disabling von Mises-Fisher (vMF) sampling improves Stage 1 reconstruction thanks to undisturbed deterministic decoding, it leads to a visible performance drop in Stage 2 reconstruction AP (Table 6). Injecting vMF noise during autoencoder training acts as an essential regularizer, equipping the vector decoder with the robustness needed to tolerate imperfect latent predictions from Stage 2.

Margin in triplet loss. The margin controls the minimum feature separation between distinct elements. In Table 7, reducing the margin to 0.2 or 0.4 degrades Stage 2 AP. Without an adequate margin, ground-truth latent tokens tend to cluster too closely in the embedding space, diminishing token discriminability and introducing ambiguity into Hungarian bipartite matching during Stage 2 reconstruction.

Table 6: Effect of vMF sampling on Stage 1 and Stage 2 reconstruction.

vMF Sampling	Lane			Road Object	
	Stage 1		Stage 2	Stage 1	Stage 2
	RMSE/m (↓)	CF (↑)	AP/% (↑)	RMSE/m (↓)	AP/% (↑)
w/o	**0.096**	**0.973**	66.62	**0.222**	56.73
w/	0.180	0.885	**94.04**	0.366	**83.44**

Table 7: Ablation on triplet loss margin parameter.

Margin	Lane			Road Object	
	Stage 1		Stage 2	Stage 1	Stage 2
	RMSE/m (↓)	CF (↑)	AP/% (↑)	RMSE/m (↓)	AP/% (↑)
0.2	0.203	0.873	<u>92.19</u>	**0.360**	81.17
0.4	0.201	0.868	91.56	0.375	<u>81.74</u>
0.6	**0.180**	**0.885**	**94.04**	<u>0.366</u>	**83.44**

Table 8: Ablation studies of vector autoencoder: attribute classification accuracy. †For our model, hidden dim=24, input mode=relative, margin α_m=0.6.

Parameter	Value	Lane		Road Object
		Color/% (↑)	Type/% (↑)	Type/% (↑)
Input mode	absolute	99.999	99.998	99.998
Hidden dim	16	99.992	99.991	99.968
	20	99.996	99.995	99.993
vMF sampling	w/o	99.999	99.994	99.951
Margin	0.2	99.993	99.999	99.988
	0.4	99.998	99.995	99.977
Ours†		99.998	99.998	99.988

B.3.2 MAP AUTOENCODER

We further investigate the key architectural designs in the second stage (Map Autoencoder). Quantitative evaluations are conducted using Precision, Recall, and AP, with detailed formulations given in Appendix B.2.

Effect of spherical denoising. As an essential design in Stage 2, spherical denoising constructs an auxiliary objective on the unit hypersphere. This design establishes direct one-to-one gradient pathways that bypass bipartite matching ambiguities, guiding query representations toward target ground-truth latents. As reported in Table 9, incorporating spherical denoising yields a substantial performance gain, elevating lane AP from 47.35% to 94.04% and road object AP from 35.70% to 83.44%. This pronounced improvement validates its necessity for stable decoder convergence.

Dimension of global tokens. Table 10 demonstrates the impact of the token embedding dimension d. Mirroring our findings in Stage 1, shrinking d from 24 to 16 incurs an evident performance drop across both categories (e.g., road object AP drops by 4.18%). This degradation confirms that an overly constrained latent bottleneck causes irreversible loss of scene-level spatial and semantic information.

Number of global tokens. As presented in Table 11, the number of global tokens strongly influences both precision and recall. Halving the token count from 256 to 128 leads to a severe decline in lane AP (from 94.04% to 84.06%). Corroborated by the qualitative visualizations in Figure 3, the token count directly governs the informational bandwidth of the scene representation, where fewer tokens restrict the capacity to encode dense, complex road layouts.

Table 9: Effect of spherical denoising.

Spherical Denoising	Lane			Road Object		
	Precision/% (↑)	Recall/% (↑)	AP (↑)	Precision/% (↑)	Recall/% (↑)	AP (↑)
w/o	70.73	50.13	47.35	62.90	40.73	35.70
w/	**95.54**	**91.25**	**94.04**	**89.96**	**85.25**	**83.44**

Table 10: Ablation on hidden dimension of global tokens.

Hidden Dim	Lane			Road Object		
	Precision/% (↑)	Recall/% (↑)	AP (↑)	Precision/% (↑)	Recall/% (↑)	AP (↑)
16	94.38	88.99	91.03	87.79	82.74	79.26
20	**96.34**	88.53	93.46	**89.98**	82.49	82.21
24	95.54	**91.25**	**94.04**	89.96	**85.25**	**83.44**

Number of decoder queries. In the Stage 2 decoder, queries extract vectorized elements from global tokens through bipartite matching. Table 12 reveals a clear trend where increasing the query count consistently improves recall for both lanes and road objects, indicating that a larger query set facilitates more comprehensive element retrieval through finer functional specialization. Since expanding the query pool beyond 800 (to 1200) yields diminishing returns in overall AP while increasing computation, we select 800 as the default operating point.

Decoder query initialization. To assist optimization, we initialize decoder queries using 800 cluster centroids obtained via K-means clustering on the pre-trained vector latents from Stage 1. As shown in Table 13, this data-driven prior provides consistent benefits, notably boosting road object AP by +1.60% by providing well-distributed semantic anchors from the start of training.

Effect of polar coordinate parameterization for $\boldsymbol{\mu}$. We investigate the impact of latent coordinate parameterization on the organization of the scene-level bottleneck. As shown in Fig. 6, even when $\boldsymbol{\mu}$ is directly regressed in standard Cartesian coordinates, the resulting latent space already exhibits a remarkably structured topology. Across scenes containing 10, 40, and 120 map elements, the $K = 256$ global tokens naturally coalesce into 12, 45, and 121 tight clusters, respectively, demonstrating that the autoencoder adaptively allocates its representational capacity in close correspondence to the underlying primitive cardinality.

However, this Cartesian parameterization inevitably retains substantial token redundancy. Because there is no explicit inductive bias to deactivate unneeded capacity, surplus tokens are forced to replicate existing element representations and disperse across finite non-zero coordinates. In contrast, decomposing the latent mean into polar coordinates, separating a radial magnitude $r = \|\boldsymbol{\mu}\|_2$ from a directional unit vector on the hypersphere, induces explicit token sparsity, as illustrated in Fig. 3(a). Under an isotropic Gaussian prior, the KL divergence penalty directly optimizes the shrinkage of the radial component. Driving $r \to 0$ provides a mathematically straightforward path to mute task-irrelevant tokens, collapsing inactive queries directly to the origin. This mechanism purges redundant representations while preserving the structured geometry, producing a clean, sparse latent substrate that facilitates more stable diffusion training.

Table 11: Ablation on number of global tokens.

Global Tokens	Lane			Road Object		
	Precision/% (↑)	Recall/% (↑)	AP (↑)	Precision/% (↑)	Recall/% (↑)	AP (↑)
128	93.28	80.72	84.06	88.49	75.58	74.85
192	95.74	88.22	91.57	89.74	82.56	79.79
256	95.54	91.25	94.04	89.96	85.25	83.44
400	**96.27**	**93.75**	**95.91**	**90.46**	**87.41**	**84.66**

Table 12: Ablation on decoder query number.

Decoder Queries	Lane			Road Object		
	Precision/% (↑)	Recall/% (↑)	AP (↑)	Precision/% (↑)	Recall/% (↑)	AP (↑)
400	**95.71**	88.98	92.79	<u>89.72</u>	83.01	81.90
800	<u>95.54</u>	<u>91.25</u>	**94.04**	**89.96**	<u>85.25</u>	<u>83.44</u>
1200	93.82	**93.47**	<u>93.74</u>	88.68	**87.76**	**83.71**

Table 13: Effect of decoder query initiation.

Query Initiation	Lane			Road Object		
	Precision/% (↑)	Recall/% (↑)	AP (↑)	Precision/% (↑)	Recall/% (↑)	AP (↑)
w/o	95.26	90.16	93.39	89.28	84.36	81.84
w/	**95.54**	**91.25**	**94.04**	**89.96**	**85.25**	**83.44**

B.4 Evaluation of Generation

Evaluation metrics. In addition to the three Fréchet distances, we report the absolute value of every individual metric as mean and standard deviation. These values provide a more direct interpretation of model behaviour and help reveal cases where a good combined score may hide an error in a specific feature. For these individual metrics, the desired result is closeness to the ground truth rather than simply a higher or lower value. The quantitative comparisons are conducted on validation split of the internal dataset, with overly sparse scenarios containing fewer than 10 lane elements excluded to ensure meaningful evaluation.

Lane geometry. We measure the shape of each lane using its arc length and maximum turning angle. Arc length is the sum of the distances between consecutive points and is reported in metres. Maximum turning angle is the largest change in direction between two consecutive lane segments and is reported in radians. We additionally divide lanes into three geometric categories according to their accumulated turning angle: straight lanes have less than 0.1 radians of total turning, moderately curved lanes have between 0.1 and 0.5 radians, and sharply curved lanes have at least 0.5 radians. These categories describe geometric curvature and do not represent semantic lane-marking types. The geometry Fréchet distance is computed from the joint distribution of lane length, maximum turning angle, and the three curvature categories. Results are shown in Table 14.

Table 14: Lane geometry comparison on validation split of the internal dataset. F_{geo} denotes the geometry Fréchet distance, where lower is better. All other values are reported as mean $\pm$ standard deviation and are better when closer to the ground truth. Straight, mild, and sharp denote geometric curvature categories.

Method	$F_{\text{geo}}\downarrow$	Length (m)	Max turn (rad)	Straight	Mild	Sharp
Ground truth	–	24.53 ± 27.23	0.31 ± 0.88	0.78 ± 0.42	0.10 ± 0.30	0.13 ± 0.34
Scenario Dreamer (B)	3.65	14.33 ± 13.29	0.19 ± 0.23	0.00 ± 0.00	0.13 ± 0.34	0.87 ± 0.34
DiT without Stage-2	**0.63**	29.62 ± 27.82	0.35 ± 0.95	0.69 ± 0.46	0.17 ± 0.37	0.14 ± 0.35
MapLDM (B)	1.23	20.82 ± 20.71	0.31 ± 0.87	0.53 ± 0.50	0.31 ± 0.46	0.16 ± 0.37
MapLDM (L)	1.21	21.23 ± 21.47	0.31 ± 0.88	0.53 ± 0.50	0.30 ± 0.46	0.16 ± 0.37
MapLDM (L), $K = 400$	<u>1.18</u>	21.84 ± 22.65	0.32 ± 0.89	0.54 ± 0.50	0.30 ± 0.46	0.17 ± 0.37

Directional diversity. Each lane direction is defined by the vector from its first point to its last point. Bearing range measures the angular span covered by all lane directions in a scene. Bearing entropy is calculated from an eight-bin direction histogram and measures how evenly the lanes are distributed across different directions. Mean nearest-neighbour angle measures the average undirected angle between each lane and its geometrically closest neighbouring lane. Small values indicate mostly parallel lane structures, while larger values indicate more varied local arrangements. The diversity Fréchet distance combines bearing range, bearing entropy, and nearest-neighbour angle. Results are shown in Table 15.

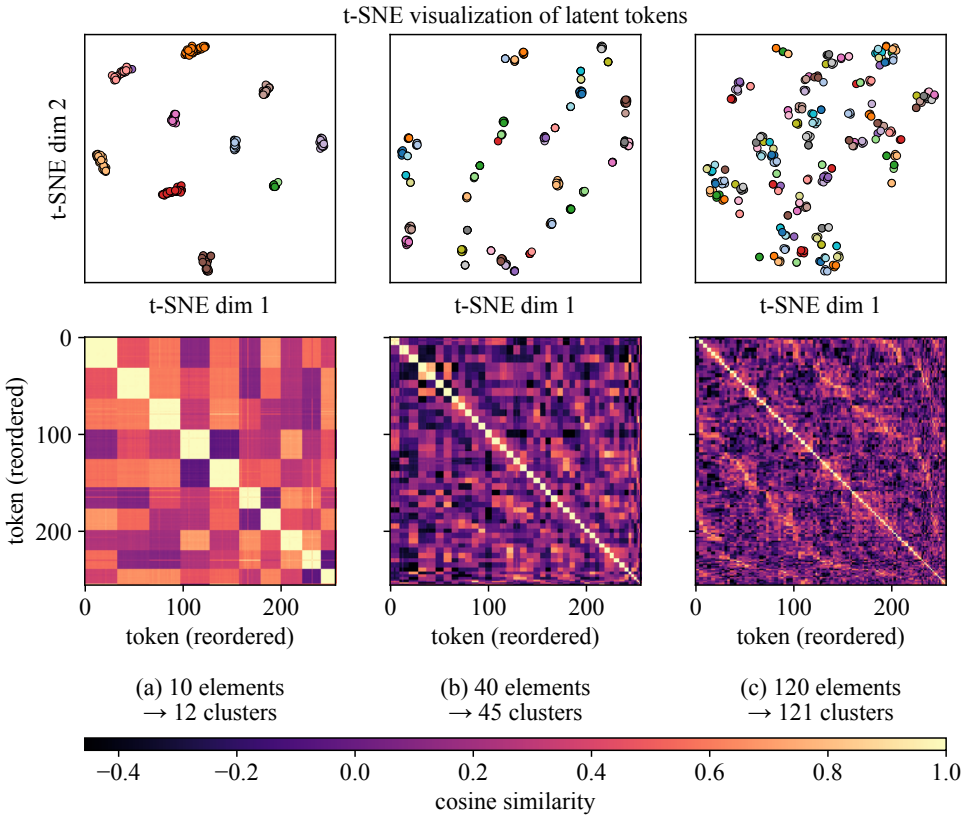

Figure 6: **The 256 global tokens collapse onto as many centres as the scene has map elements.** *Top:* t-SNE of the K=256 latent tokens of one validation scene, coloured by agglomerative cluster (cosine distance, average linkage, cut at $\tau^{\star}$=0.1). *Bottom:* the cosine matrix of the same tokens reordered by cluster, on a shared colour scale. Scenes with (a) 10, (b) 40 and (c) 120 map elements give 12, 45 and 121 clusters.

Table 15: Directional diversity comparison on validation split of the internal dataset. F_{div} denotes the diversity Fréchet distance, where lower is better. All other values are reported as mean $\pm$ standard deviation and are better when closer to the ground truth. NN angle denotes the mean nearest-neighbour lane angle.

Method	$F_{\text{div}}\downarrow$	Range (rad)	Entropy	NN angle (rad)
Ground truth	–	3.54 ± 0.87	1.21 ± 0.33	0.07 ± 0.11
Scenario Dreamer (B)	1.52	3.22 ± 0.69	0.81 ± 0.30	0.05 ± 0.03
DiT without Stage-2	1.09	3.39 ± 0.53	0.97 ± 0.29	0.03 ± 0.05
MapLDM (B)	<u>0.72</u>	3.95 ± 0.83	1.13 ± 0.42	0.11 ± 0.11
MapLDM (L)	<u>0.72</u>	3.96 ± 0.86	1.12 ± 0.42	0.11 ± 0.11
MapLDM (L), $K = 400$	**0.71**	3.93 ± 0.87	1.11 ± 0.42	0.11 ± 0.11

Scene layout. We measure scene-level structure using the number of lanes and lane coverage. Lane coverage is the fraction of occupied cells in a 32×32m grid covering a 128×128m region. It reflects how widely the generated lanes are distributed across the scene. The layout Fréchet distance is calculated from lane count and coverage. The number of road objects is reported separately because Scenario Dreamer does not generate road objects; including it in the combined distance would therefore make the comparison unfair. Results are shown in Table 1.

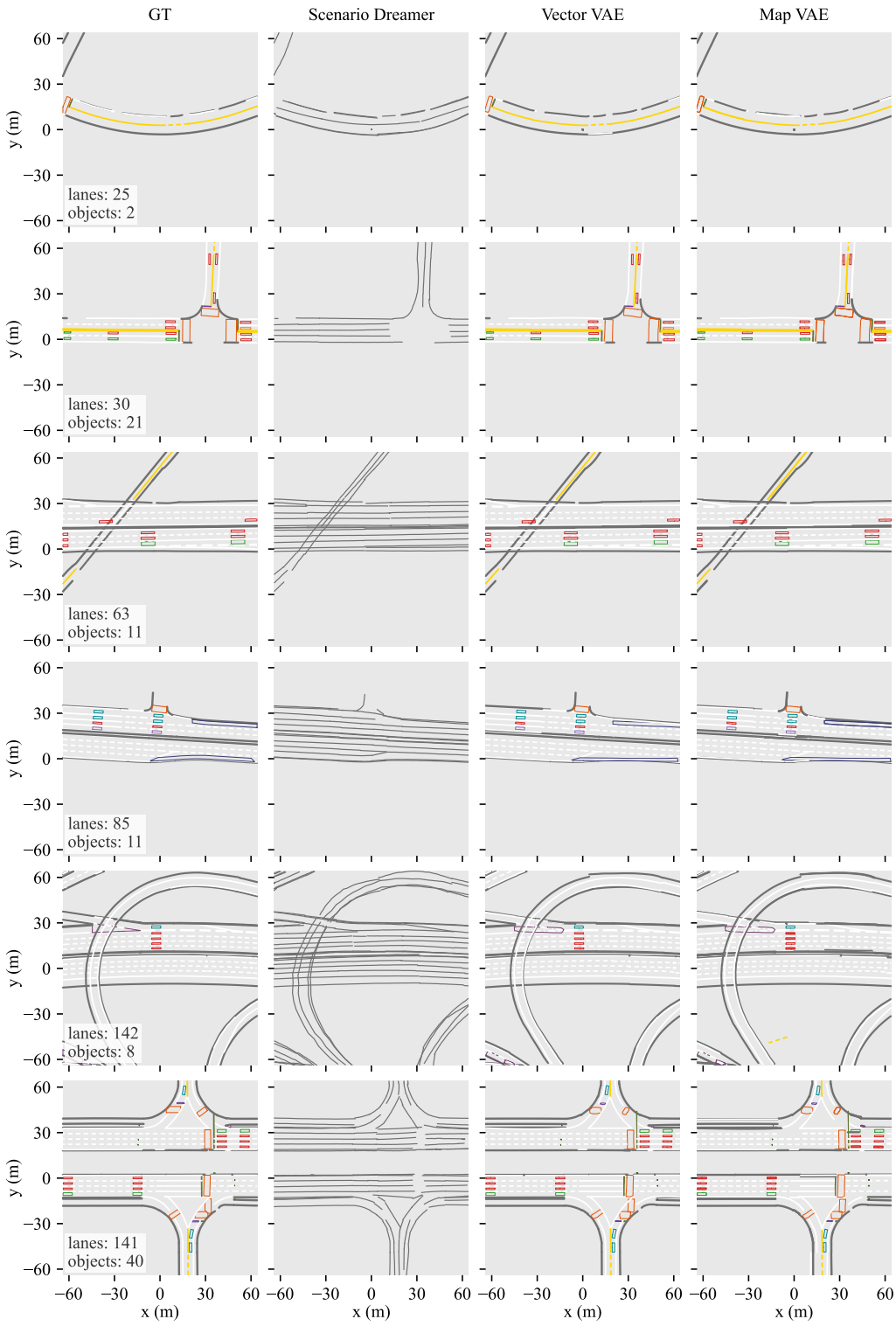

Figure 7: **Reconstruction results across six scenarios.** Map elements are visualized using their ground-truth/predicted semantic classes, paint colors, and line styles (e.g., solid vs. dashed). Our framework reconstructs both patterned lane polylines and multi-class road objects within a single unified representation.

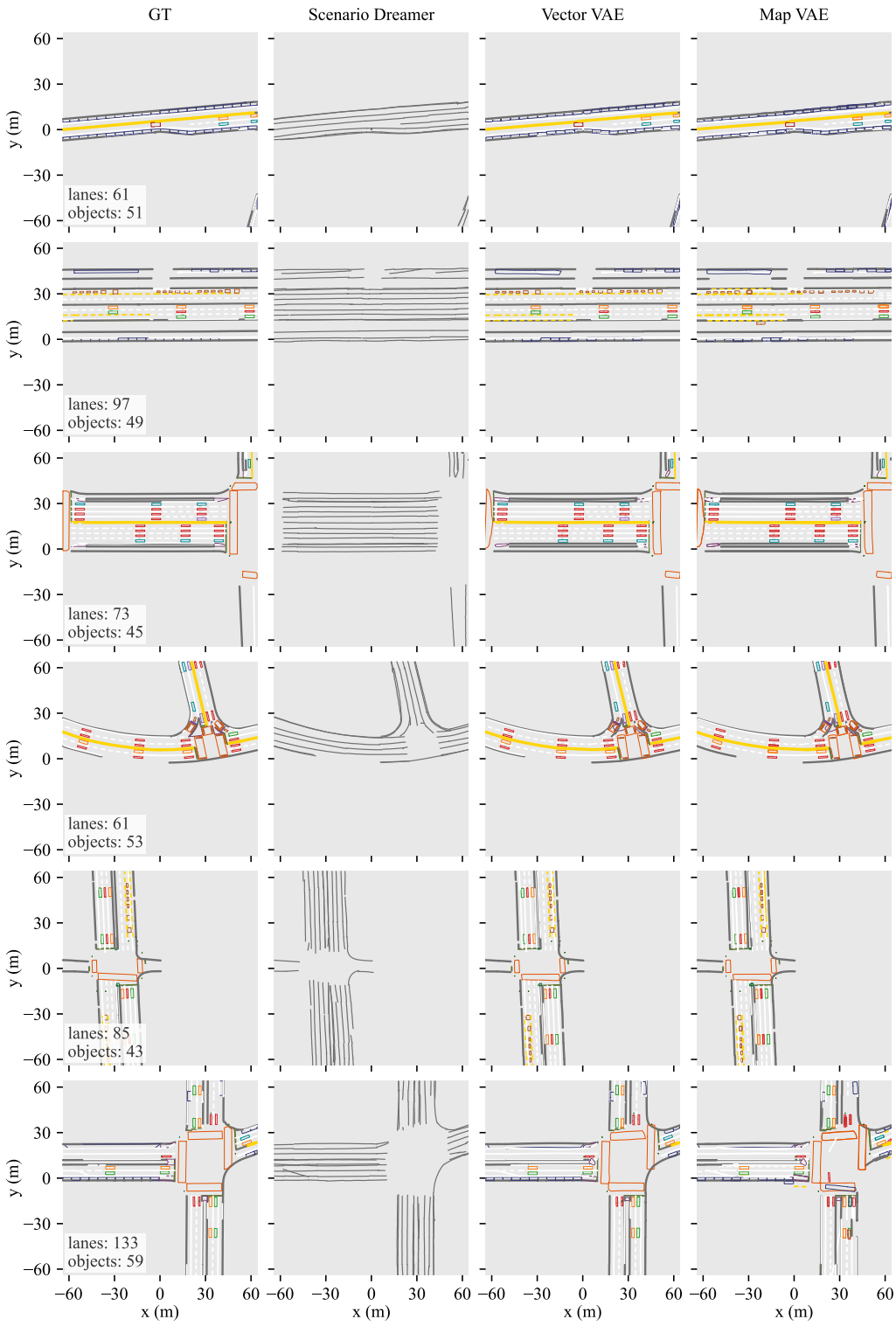

Figure 8: **Reconstruction results across six object-rich scenarios.** Map elements are visualized using their ground-truth/predicted semantic classes, paint colors, and line styles (e.g., solid vs. dashed). Our framework reconstructs both patterned lane polylines and multi-class road objects within a single unified representation.

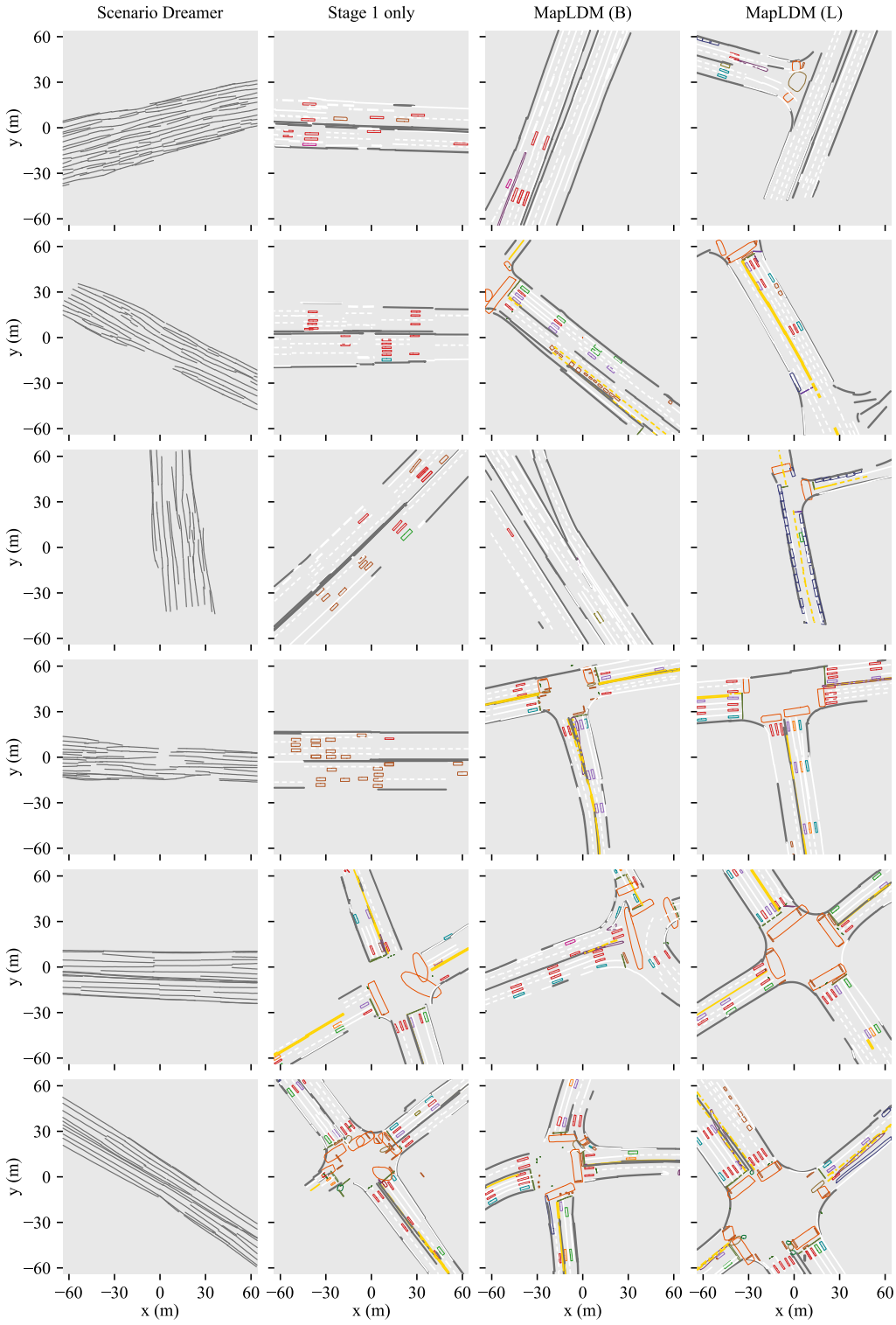

Figure 9: **Visual comparison of generated HD maps.** (a) Scenario Dreamer Rowe et al. (2025) suffers from mode collapse (e.g., trivial straight roads) due to semantic drift from heuristic token sorting. (b) W/o Stage-2 bottleneck: Direct diffusion on Stage-1 latents produces fragmented layouts. (c, d) MapLDM (Base vs. Large) using 256 latent tokens: Synthesizes diverse, realistic topologies, such as highways, ramps, T-junctions, and urban intersections, with coherent road structures and object layouts.