MAPDLM: LANE-ALIGNED BLOCK DIFFUSION FOR LANE GRAPH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Constructing HD maps from multi-frame LiDAR BEV representations requires predicting lane-centerline geometry and directed lane connectivity. Generic vision–language model tokenizers are not designed to represent coordinates, while long token sequences weaken the association between coordinate points and the lane geometry and topology they define. We present MAPDLM, a lane-supervised multi-token generation model based on *lane-aligned block diffusion*. MAPDLM first predicts a lane-ID list and directed lane connectivity, then reconstructs each lane as fixed-length token blocks conditioned on this structure. In addition to the standard cross-entropy loss, training combines point-level coordinate objectives, lane-level polyline constraints, and topology-chain supervision. Anchor-Point Gaussian Loss (APG Loss) tolerates errors between nearby coordinate bins in lane interiors while retaining exact targets at lane endpoints. Semantic-Adaptive Noise Scheduling and structure-aware commitment expose the model to the highly masked states encountered during iterative denoising. By jointly predicting multiple lanes within each diffusion block, MAPDLM reduces serial decision depth and error propagation. We evaluate MAPDLM on Argoverse 2 and nuScenes under the OpenLane-V2 protocol, compare it with corresponding autoregressive baselines, and conduct controlled ablations of lane alignment, multi-level supervision, and state-aligned denoising. The results show effective geometry and topology reconstruction on both benchmarks and support lane-aligned diffusion as a promising approach to structured map generation.

1 INTRODUCTION

HD map construction from bird's-eye-view (BEV) observations requires predicting a lane graph: the geometry of individual lanes and their directed connections. We study local map reconstruction from pose-aligned accumulated LiDAR within a fixed spatial region, with the graph expressed in the reference ego frame. This graph supports HD mapping Lu et al. (2023), traffic forecasting Fang et al. (2021), and motion planning Jiang et al. (2023). Many systems represent these outputs through learned queries and specialized detection and graph heads Li et al. (2023b; 2025). Sequence-based methods Peng et al. (2024); Xie et al. (2025); Lu et al. (2025) and MLLM-based map generation Yuan et al. (2026) offer an alternative in which geometry and connectivity are expressed through a token interface. This raises a modeling question: how should a generative model organize its predictions and supervision around the lane objects represented by those tokens?

The lane is a semantic prediction target rather than an arbitrary span of text. Its ordered coordinate pairs jointly define a centerline, its endpoints participate in directed connections, and its identity associates that geometry with a graph node. A coordinate loss can teach numerical proximity without explicitly organizing jointly recovered tokens into these objects. In particular, our autoregressive (AR) controls already use a coordinate-space L1 objective to learn geometric values. The design question is how a token interface can preserve the association between lane identity, geometric supervision, and iterative recovery when a complete lane spans several decoding blocks.

Block diffusion makes this question explicit because several positions are predicted together. An arbitrary partition of a serialized graph can separate structural tokens from the coordinates they identify, or divide a lane without preserving its identity. In our experiments, applying block diffusion to the original token sequence produces nearly input-independent outputs during free-running generation despite reasonable masked-prediction diagnostics. Replacing ordinary coordinate strings with atomic tokens gives a modest improvement in our AR control. These observations motivate a

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.

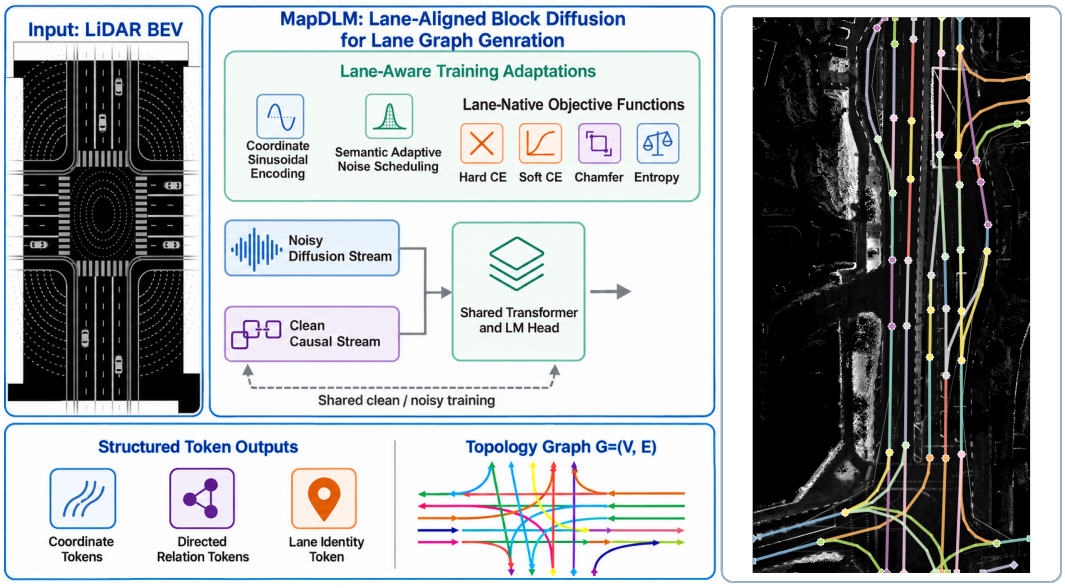

Figure 1: **MapDLM overview.** Accumulated LiDAR BEV input is mapped to lane geometry and directed connectivity through lane-aware block diffusion and geometric supervision. The clean causal and noisy branches share the inherited transformer and language-model head; the overview introduces no separate retrieval or verification network. The road and graph drawings are schematic; the right panel illustrates the lane representation over LiDAR BEV.

representation in which the tokens recovered together have explicit lane-level meaning; they do not imply that autoregression is incapable of learning lane geometry.

Figure 1 gives an overview of MAPDLM, which adapts Fast-dVLM-3B Wu et al. (2026) by making the lane the common semantic target of representation, supervision, and recovery. Directed topology and a lane-ID inventory precede geometry. Geometry is organized into fixed 32-token diffusion blocks whose internal lane records preserve identity and point order. Longer lanes retain the same ID across records and are reassembled for complete-polyline supervision. Coordinate-Value Sinusoidal Encoding (CVSE) supplies value-indexed coordinate features; Anchor-Point Gaussian Loss (APG) retains hard endpoint targets while allowing local tolerance for interior coordinates; and geometric objectives supervise reconstructed polylines. Semantic-Adaptive Noise Scheduling (SANS) and section-weighted CE coordinate masked-state coverage with the allocation of supervision.

This design keeps the semantic target fixed as a lane while allowing variable-length geometry and joint recovery within each block, reducing serial decisions inside the active block. Topology and completed blocks remain causal context, so cross-block error propagation remains possible. Our AR comparison evaluates complete systems with geometric supervision; internal ablations examine MapDLM's representation and training configurations.

Our contributions are threefold:

- We formulate lane-graph generation around persistent lane identities, with topology-first serialization and lane-aware 32-token recovery blocks. This organization binds coordinate predictions and directed graph references to the same semantic objects.
- We adapt a pretrained block-diffusion VLM by coordinating coordinate- and polyline-level supervision with section-dependent corruption and structure-aware token commitment. The serialization layout remains fixed while the effective training partition grows to the inference block size.
- We evaluate the complete system on Argoverse 2 and nuScenes with a shared LiDAR-BEV pipeline and BEV-projected lane metrics. The ablations document a configuration-dependent interaction between noise scheduling and section-weighted supervision in the reported runs, while the resulting tokenization reduces mean output length by 42.2%.

2 Related Work

2.1 Lane Graphs and Generative Map Construction

Vectorized mapping and lane-topology prediction use task-specific detection and graph architectures. MapTR Liao et al. (2022) groups points into structured queries; VectorMapNet Liu et al. (2023) autoregressively decodes map elements; TopoNet Li et al. (2023a) couples centerline detection and topology reasoning; and LaneSegNet Li et al. (2023b) represents lane segments. TopoMLP Wu et al. (2023), TopoLogic Fu et al. (2025), and RATopo Li et al. (2025) improve coordinate representations, endpoint reasoning, and assignment; MachMap Qiao et al. (2023) and CenterLineDet Xu et al. (2023) address compact vector maps and centerline graphs.

Sequence-based methods organize map structure through a generation order: LaneGraph2Seq Peng et al. (2024) serializes vertices and edges, SeqGrowGraph Xie et al. (2025) grows graphs, RoadNet Lu et al. (2025) models landmark and lane entities, and Topo2Seq Yang et al. (2025a) learns topology sequences. UniMapGen Yuan et al. (2026) generates vectorized maps with specialized tokens, multimodal inputs, and state updates for large-scale continuity, and evaluates nuScenes lane topology with landmark and reachability metrics. MapDLM instead studies iterative block denoising around lane identity for centerlines and explicit directed relations within a pretrained block-diffusion VLM. TopoGPT Fu et al. (2026) learns a lane-graph geometry prior by AR pretraining and adapts multi-view BEV features to it; our in-house AR controls do not reproduce that pipeline.

2.2 Coordinate Representations and Geometric Supervision

VLMs Bai et al. (2025); Chen et al. (2024) have been adapted to spatial driving through structured interfaces Zhou et al. (2025); Yang et al. (2025b); Rex-Omni Jiang et al. (2025) uses point tokens and UniMapGen Yuan et al. (2026) uses specialized map-vector tokens. Atomic tokens prevent subword fragmentation but do not encode numerical distance. Our AR controls therefore combine token CE with L1-based `coord_soft`; MapDLM combines value-indexed features, endpoint-sensitive targets, and complete-polyline objectives. Current ablations do not isolate their individual contributions.

2.3 Diffusion and Structured Parallel Generation

Masked diffusion models discrete sequences Austin et al. (2021); Lou et al. (2024), including LLaDA Nie et al. (2025); blockwise Wan et al. (2026) and hybrid AR–diffusion He (2025); Chandrasegaran et al. (2025); Huang (2025) methods study related trade-offs. Fast-dVLM Wu et al. (2026) converts an AR VLM through joint clean/noisy training and a block-size curriculum. MapDLM retains this backbone while adapting response organization, objectives, corruption, and commitment to lane graphs. Fast-dDrive Zhang et al. (2026) already combines semantic-section diffusion, section-weighted losses, and section-adaptive Beta noise with a frozen structural scaffold; we therefore treat corruption and weighting as related strategies, not standalone novelty. LaneDiffusion Wang et al. (2025) uses diffusion BEV feature priors, and MapDreamer Brandes et al. (2026) generates lane geometry and topology by aerial-image-conditioned latent diffusion. MapDLM instead denoises discrete tokens from accumulated LiDAR BEV, with IDs linking graph references and coordinate records.

Structured generation also uses schema scaffolds Xiong et al. (2025) and object-aligned parallel outputs: LocateAnything Wang et al. (2026) predicts detection boxes in parallel, and TrajDLM Wongso et al. (2026) generates trajectories over road-segment IDs. In MapDLM, lane identity organizes geometric supervision and within-block records, while token roles guide commitment and corruption controls training states.

3 Method

3.1 Lanes as Semantic Prediction Targets

MapDLM builds on Fast-dVLM-3B Wu et al. (2026), a block-diffusion VLM converted from Qwen2.5-VL-3B Bai et al. (2025). Given a BEV image $\mathbf{I}_{\mathrm{BEV}}$ rendered from accumulated LiDAR, it predicts a directed lane graph $G = (\mathcal{V}, \mathcal{E})$. Each node $i \in \mathcal{V}$ has a persistent lane ID and an ordered centerline $\mathcal{P}_i = (p_{i,1}, \ldots, p_{i,n_i})$; an edge $(i, j) \in \mathcal{E}$ denotes a directed lane connection. Coordinates describe the BEV plane. The transformer backbone and language-model head are inherited from Fast-dVLM.

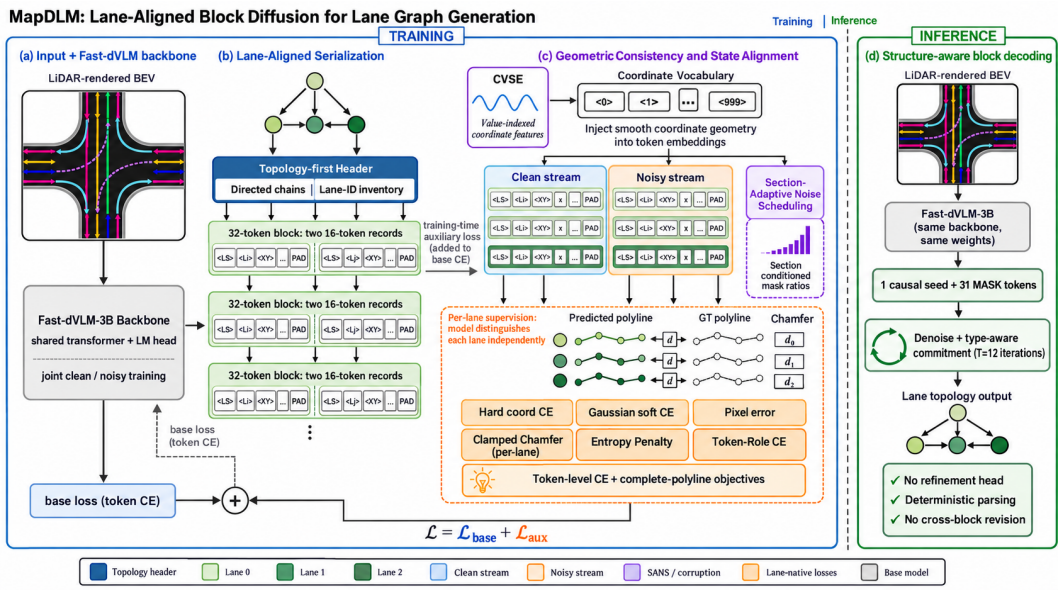

Figure 2: **MapDLM training and inference.** Lane-aware serialization connects token-level and complete-polyline supervision to iterative recovery. Each 32-token geometry block packs two 16-token records; a complete lane may occupy several records. Inference starts from one causal seed and 31 MASK positions and uses structure-aware commitment. The geometry layout and training curriculum are specified in Sections 3.2 and 3.3.

Figure 2 summarizes the pipeline. The lane is the semantic unit of prediction and geometric supervision; the $B = 32$ token block is the iterative decoding unit. Each fixed-capacity lane record retains an ID and point order, carries up to K coordinate pairs, and connects these levels: long lanes span records or blocks and are reassembled for polyline supervision. Thus, identity persists across boundaries while joint prediction is limited to the active block, without redefining lane endpoints. Here, lane-aligned means record boundaries respect lane identity; it does not require a one-to-one correspondence between lanes and blocks.

3.2 Lane-Aligned Serialization

Topology before geometry. The response has the form

$$\underbrace{\langle \text{MAP_S} \rangle\, \mathcal{H}\, \mathcal{G}}_{\text{topology and lane inventory}} \quad \mathcal{A}\, \mathcal{D}_1 \cdots \mathcal{D}_M\, \langle \text{MAP_E} \rangle, \tag{1}$$

where $\mathcal{H}$ contains directed lane-ID chains, $\mathcal{G}$ lists lanes to be grounded by coordinates, $\mathcal{A}$ aligns geometry to a block boundary, and $\mathcal{D}_j$ is a 32-token geometry block. A topology chain is

$$\mathcal{T}_k = \langle \text{CH_S} \rangle \langle \text{C}_\text{k} \rangle \langle \text{L}_{\text{k},1} \rangle \cdots \langle \text{L}_{\text{k},\text{m}_\text{k}} \rangle \langle \text{CH_E} \rangle. \tag{2}$$

Consecutive IDs encode $L_{k,j} \to L_{k,j+1}$, and a lane may appear in multiple chains for branching or merging. Both prefix and geometry are predicted from the input; the prefix is not ground truth at inference. The layout defines the target grammar rather than supplying lane lengths or a geometry scaffold. IDs, coordinates, padding, and the map-end token are generated; repeated IDs continue a lane and changed IDs start another. Each later block conditions on the generated prefix and completed blocks, so early topology or inventory errors can propagate without cross-block revision.

The 32-token recovery block. Each geometry block contains two fixed-layout lane records:

$$\mathcal{D}_j = [\mathcal{Q}_{2j-1} \mid \mathcal{Q}_{2j}], \qquad |\mathcal{D}_j| = B = 32. \tag{3}$$

A record $\mathcal{Q}_a$ associates lane ID $L_{\ell(a)}$ with at most $K = 6$ coordinate pairs:

$$\mathcal{Q}_a = \langle \text{LS} \rangle \langle \text{L}_{\ell(\text{a})} \rangle \langle \text{XY} \rangle\ \langle \text{x}_1 \rangle \langle \text{y}_1 \rangle \cdots \langle \text{x}_{\text{k}_\text{a}} \rangle \langle \text{y}_{\text{k}_\text{a}} \rangle\ [\langle \text{PAD} \rangle]^*, \tag{4}$$

where $1 \le k_a \le K$, each nonempty record has 16 positions, and $16 - (3 + 2k_a)$ PAD tokens fill the remainder; even six coordinate pairs leave one trailing PAD. Longer lanes use consecutive same-ID records, and an all-PAD record completes the last block when required. The records may

describe distinct lanes or successive portions of one lane, preserving layout and identity without requiring a lane to fit in one block. Coordinates use $V = 1{,}000$ BEV value bins. In Eq. 4, (x_j, y_j) are image-plane (u_j, v_j) coordinates, not ego-frame axes, with $0 \le u < 500$ and $0 \le v < 1000$. Each coordinate prediction has an explicit lane target, and active records are recovered jointly from the generated prefix and completed causal blocks.

3.3 Joint Training and Block Alignment

We retain Fast-dVLM's joint clean/noisy training. The noisy branch masks a sampled subset $\mathcal{M}$ of response positions and predicts their clean identities:

$$\mathcal{L}_{\text{noisy}} = -\sum_{i \in \mathcal{M}} \log P_\theta(y_i \mid \widetilde{\mathbf{y}}, \mathbf{I}_{\text{BEV}}), \tag{5}$$

where $\mathcal{M}$ denotes masked non-PAD response positions and $\widetilde{\mathbf{y}}$ is the corrupted response and access to its positions is restricted by the effective block-causal attention mask. The clean branch provides causal next-token supervision supporting the seed predictions used during decoding. Both branches share transformer weights:

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{clean}} + \lambda_{\text{noisy}} \mathcal{L}_{\text{noisy}}. \tag{6}$$

Fixed layout, growing training partition. A *block* is always the fixed 32-token serialization and inference unit. Training grows its effective attention and corruption partition to this target (Appendix A.8): early subwindows can split a record, whereas the final 32-token stage lets records share a sampled noise level and partially masked states, matching inference. This stage occupies about 44% of training. PAD positions may be masked but are excluded from base CE and receive a separate objective.

3.4 Learning Coordinates as Lane Geometry

Coordinate-Value Sinusoidal Encoding (CVSE). We add a fixed value-indexed feature to the learned embedding of coordinate bin q:

$$\mathbf{e}_{\text{coord}}(q) = \mathbf{e}_{\text{learned}}(q) + \alpha_{\text{cvse}} \boldsymbol{\phi}_{\sin}(q), \tag{7}$$

$$\begin{aligned} \boldsymbol{\phi}_{\sin}(q, 2j) &= \sin(q/10000^{2j/D}), \\ \boldsymbol{\phi}_{\sin}(q, 2j+1) &= \cos(q/10000^{2j/D}). \end{aligned} \tag{8}$$

Its inner product depends on coordinate separation Vaswani et al. (2017), giving visible coordinate tokens a value-indexed basis. It does not impose a globally monotonic distance metric on learned embeddings. CVSE adds no trainable parameters and leaves MASK embeddings unchanged.

Anchor-Point Gaussian Loss (APG Loss). Coordinate supervision distinguishes the first and last point of a complete lane from its interior points:

$$\ell_i = (1 - \alpha_i)\, \text{CE}_{\text{hard}}(z_i, y_i) + \alpha_i\, \text{CE}_{\text{soft}}(z_i, \widetilde{y}_i). \tag{9}$$

The soft target $\widetilde{y}_i$ is a truncated Gaussian on the one-dimensional coordinate-bin axis. Interior points use $\alpha_i = \alpha_{\text{int}} > 0$; lane endpoints use $\alpha_i = 0$. Endpoints therefore retain exact target labels, while nearby-bin deviations receive partial credit in lane interiors. Record boundaries do not redefine the endpoints of a reassembled lane. This retains exact endpoint labels while allowing local tolerance in lane interiors. It does not impose equality between the endpoints of two connected lanes.

Polyline geometry and supervision allocation. Expected-coordinate and point-wise L2 losses penalize numerical displacement after normalization by s_{px}. Symmetric clipped Chamfer distance and tangential-direction loss operate on reconstructed polylines, while an entropy hinge discourages overly diffuse coordinate distributions. The lane objective is

$$\begin{aligned} \mathcal{L}_{\text{lane}} = \mathcal{L}_{\text{APG}} &+ \gamma_{\text{pos}} \mathcal{L}_{\text{pos}} + \gamma_{\text{point}} \mathcal{L}_{\text{point}} \\ &+ \gamma_{\text{ch}} \mathcal{L}_{\text{ch}} + \gamma_{\text{dir}} \mathcal{L}_{\text{dir}} + \gamma_{\text{ent}} \mathcal{L}_{\text{ent}}. \end{aligned} \tag{10}$$

Repeated-ID records are reassembled before polyline supervision, so boundaries introduce no endpoints. Coordinate distributions yield expected scalars before point pairing, adding numerical supervision to categorical CE (Appendix A.6). Skeleton and topology-chain tokens receive role-weighted CE, with IDs referring to the same objects as the geometric objectives; section weighting allocates

supervision by role, and a separately normalized PAD loss recovers masked padding. The complete objective is

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda_{\text{sec}}\mathcal{L}_{\text{sec}} \\ + \lambda_{\text{lane}}(\mathcal{L}_{\text{lane}}^{\text{clean}} + \lambda_{\text{noisy}}\mathcal{L}_{\text{lane}}^{\text{noisy}}) + \lambda_{\text{pad}}\mathcal{L}_{\text{pad}}.\end{aligned} \tag{11}$$

These terms supplement base CE. APG differs from the L1-based `coord_soft` used by AR controls: both provide a numerical geometry signal, but MapDLM additionally organizes supervision and joint recovery around lane identity. Hyperparameters are in Section 4.1.

3.5 Corruption for Iterative Lane Recovery

Semantic-Adaptive Noise Scheduling (SANS), shown as section-adaptive masking in Fig. 2, conditions the training mask-ratio distribution on semantic token roles. Inference starts each block from a seed and $B-1$ MASK tokens, motivating explicit coverage of highly masked geometry. SANS assigns the effective training partition to its dominant semantic section by majority vote over token roles and samples its mask ratio from

$$P_s(p) = \sum_{k=1}^{K_s} w_{s,k} D_{s,k}(p), \qquad w_{s,k} \geq 0, \quad \sum_k w_{s,k} = 1, \tag{12}$$

where each $D_{s,k}$ is Beta or Uniform. Geometry includes a high-mask $\mathcal{U}(p_{\min}, 1)$ component and other sections use lighter corruption. These fixed per-section distributions, rather than current model uncertainty, control recovery difficulty; section-weighted CE controls relative emphasis. SANS changes corruption, section weighting changes supervision allocation, and structure-aware commitment is a separate inference rule (Section 4.1). Matching mask density does not correct errors in generated causal context.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on Argoverse 2 Wilson et al. (2023) and nuScenes Caesar et al. (2020) using OpenLane-V2 lane annotations Wang et al. (2023) with a BEV evaluation setting. For Argoverse 2 we use the official sensor split and subset A annotations; for nuScenes we use the default split and subset B annotations.

Data processing. We register consecutive LiDAR frames in the reference ego frame, accumulate them, and rasterize the result at 0.1 m/px over a 100×50 m BEV region. Lane centerlines and directed links covering the same region are transformed, clipped, and aligned to the resulting $H \times W = 1000 \times 500$ grid before lane-aware serialization (Section 3.2). The official AV2 training split is further divided for optimization and validation; its official validation split is reserved for testing.

Optimization and training configuration. We fine-tune Fast-dVLM-3B Wu et al. (2026) on 8 GPUs under DeepSpeed ZeRO-2 with bf16 and gradient checkpointing. The effective batch size is 8, and the vision encoder, projector, and language backbone are trainable. We use AdamW with a cosine schedule, peak learning rate 2×10^{-5}, warmup ratio 0.1, and 20 epochs for both main and ablation experiments. The coordinate supervision follows a separate staged schedule: a conservative warmup first learns the structural distribution, after which the full lane objective in Section 3.4 is enabled. The effective attention and corruption partition also follows the inherited curriculum, reaching the fixed 32-token inference block in the final training stage (Appendix A.8).

Representation and loss parameters. We use a fixed $B = 32$ diffusion block containing two lane records. Each record holds at most $K = 6$ coordinate pairs; this capacity covers about 94% of lanes, while longer lanes use repeated-ID records. The semantic prediction target remains the complete lane; the block defines which records are recovered jointly. The coordinate vocabulary has $V = 1{,}000$ bins, $\langle 0 \rangle$–$\langle 999 \rangle$. We use $\lambda_{\text{noisy}} = 0.5$, $\lambda_{\text{lane}} = 0.2$, $\lambda_{\text{sec}} = 0.1$, and $\lambda_{\text{pad}} = 0.02$. APG uses $\sigma = 1$, truncation radius $r = 2$, and $\alpha_{\text{int}} = 0.2$; endpoints use $\alpha_i = 0$. Expected-coordinate and point losses use $s_{\text{px}} = 50$ px and weights $\gamma_{\text{pos}} = \gamma_{\text{point}} = 1.0$. We set $\gamma_{\text{ch}} = 0.1$ with a 200 px clipping radius and 2.5 normalization, $\gamma_{\text{dir}} = 0.2$, and $\gamma_{\text{ent}} = 0.01$ with hinge target $\log 3$. Structural closing, lane-ID, and topology-chain CE weights are 0.5, 1.0, and 3.0. CVSE uses $\alpha_{\text{cvse}} = 0.3$.

Noise parameters. The single-Beta schedules are geometry $\text{Beta}(2, 1)$, summary $\text{Beta}(1, 1.5)$, topology $\text{Beta}(1, 2)$, and shape $\text{Beta}(0.5, 3)$. Their section CE weights ω_s are $6.0/1.0/3.0/1.5$,

Table 1: **Argoverse 2, OpenLane-V2 subset A.** L-OLS, DET_l, and TOP_{ll} are reported on the percentage scale. Only in-house systems under the common planar LiDAR-BEV protocol are compared. *AR-VLM uses ordinary BPE tokens; †the special-token control uses registered atomic tokens.

Method	L-OLS↑	DET_l↑	TOP_{ll}↑
In-house LiDAR-BEV systems			
AR-VLM*	22.9	9.5	13.2
AR-VLM + special tokens†	24.1	10.7	14.2
MAPDLM	**43.9**	**33.6**	**29.5**

Table 2: **nuScenes, OpenLane-V2 subset B.** L-OLS, DET_l, and TOP_{ll} are reported on the percentage scale. Only in-house systems under the common planar LiDAR-BEV protocol are compared. *AR-VLM uses Qwen2.5-VL-3B with the reported autoregressive setup.

Method	L-OLS↑	DET_l↑	TOP_{ll}↑
In-house LiDAR-BEV systems			
AR-VLM*	21.3	8.2	11.9
MAPDLM	**43.5**	**32.2**	**30.1**

respectively. The full model replaces the geometry distribution with

$$P_{\mathrm{geo}}(p) = 0.3\,\mathrm{Beta}(2,1) + 0.2\,\mathrm{Beta}(5,1) + 0.5\,\mathcal{U}(0.9,1.0), \tag{13}$$

Inference configuration. Decoding uses the fixed $B = 32$ block. The prompt supplies the first seed token; subsequent seeds are generated from causal context. Each active block is initialized as $[\text{seed} \mid \text{MASK} \times 31]$ and denoised for $T = 12$ steps. At each iteration, predictions are ranked by confidence and committed according to an annealed schedule. Token roles inferred from committed left context prioritize structural or coordinate-compatible predictions. This prioritization encourages preservation of the lane-record skeleton. Completed blocks provide causal context through KV-cache reuse, and decoding stops at $\langle \mathrm{MAP_E} \rangle$. No learned verification head or cross-block revision is used.

Evaluation metrics. We report centerline detection (DET_l), directed lane-lane topology (TOP_{ll}), and their lane-only aggregate, L-OLS. With both components normalized to $[0, 1]$, we define

$$\text{L-OLS} = \frac{1}{2}\left[\mathrm{DET}_l + \sqrt{\mathrm{TOP}_{ll}}\right]. \tag{14}$$

All three metrics are reported on a percentage scale. Equivalently, for percentage-scale component scores D and T, the reported L-OLS is $(D + 10\sqrt{T})/2$. L-OLS evaluates the lane subset only; it is not the full OpenLane-V2 OLS, which additionally includes traffic-element detection and lane-traffic topology. MapDLM predicts neither of these additional outputs.

Evaluation and comparison scope. Our in-house evaluation discards height z and matches predicted and ground-truth centerlines using planar (x, y) coordinates; directed topology is scored on the associated lane matches. Thus DET_l, TOP_{ll}, and L-OLS describe the BEV lane-graph task for all in-house systems. Tables 1 and 2 compare only systems evaluated under this common protocol. Published camera-based component scores are retained separately in Appendix A.12, since their inputs and geometric matching protocols differ. These experiments establish an internal generative-model comparison, rather than a benchmark ranking against those methods.

Relation to generative mapping methods. UniMapGen Yuan et al. (2026) uses Mask/Chamfer AP and Landmark/Reachability precision, recall, and F1, which measure different matching and graph properties from DET_l/TOP_{ll}. TopoGPT Fu et al. (2026) is a relevant generative AR alternative with a learned geometry prior; it is discussed as related work and is not a rerun baseline in this study.

AR comparison protocol. Our AR controls use Qwen2.5-VL-3B with token cross-entropy and `coord_soft`, an L1 objective that penalizes coordinate-value errors. They therefore receive explicit geometric supervision. The special-token control retains this AR training setup and replaces coordinate strings and structural units with atomic registered tokens. MapDLM uses the converted Fast-dVLM checkpoint, lane-aligned serialization, and the training objectives in Section 3. The comparison measures complete systems with coordinate supervision and does not isolate decoder factorization. In particular, the L1-based AR objective and MapDLM's Gaussian soft-label APG loss are distinct formulations.

4.2 Main Results

Tables 1 and 2 report lane-graph performance using L-OLS and its detection and topology components for the in-house LiDAR-BEV systems. Under the common in-house protocol, MapDLM

Table 3: **Representation and training configurations on AV2 subset A.** All variants use the 32-token target block and same epoch budget. "Uniform" is the baseline mask schedule; Sec. CE is section-weighted cross-entropy. Variant A has qualitative free-running collapse and no quantitative score.

Variant	Representation	Noise	Sec. CE	$\text{DET}_l \uparrow$	$\text{TOP}_{ll} \uparrow$
A	token sequence	Uniform	–	*collapse*	
B	lane-aligned	Uniform	–	18.1	15.9
C	lane-aligned	Beta	–	18.6	16.4
D	lane-aligned	Uniform	✓	23.0	20.2
E	lane-aligned	Beta	✓	32.9	28.8
F	lane-aligned	Mixture	✓	33.6	29.5

reaches 33.6/29.5 DET_l/TOP_{ll} on AV2 and 32.2/30.1 on nuScenes, improving both components over the AR controls. On AV2, atomic tokens improve the AR control from 9.5/13.2 to 10.7/14.2 despite both variants receiving explicit L1 geometric supervision. These results support the complete lane-aligned system; the AR comparison protocol above does not separately attribute gains to initialization, representation, auxiliary objectives, or decoder factorization. The following ablations examine representation and training choices within the block-diffusion setting.

4.3 Ablation Studies

All ablations use AV2 subset A, a 20-epoch training budget, and a target block size of 32. Variant A uses the original token sequence; B–F use the lane-aligned representation. Within B–E, the table varies the noise schedule and section-weighted CE, while E–F compare geometry-noise distributions. These configurations do not separately isolate CVSE, APG, or each polyline-loss term. Thus, A–B should be read as a representation-and-recovery comparison that includes the implemented layout and padding, rather than as an isolated test of persistent lane IDs. The remaining contrasts are descriptive tests of the reported configurations and do not establish universal effects across seeds or alternative hyperparameter choices.

Interaction between corruption and supervision. The effect of the Beta schedule depends on section-weighted CE. Without it, B→C improves DET_l/TOP_{ll} by only 0.5/0.5; with it, D→E improves the same metrics by 9.9/8.6. Section weighting alone, B→D, gives 4.9/4.3. The difference-of-differences, $(E-D)-(C-B)$, is therefore 9.4/8.1. These descriptive contrasts show a configuration-dependent interaction. Changing the mask distribution also changes the expected number of masked targets; without a matched-supervision-budget control, the contrast cannot distinguish recovery difficulty from effective loss weighting. They do not establish significance across random seeds or the standalone effect of each auxiliary loss.

High-mask coverage. Changing the geometry schedule from Beta to the mixture in Eq. 13 gives a further 0.7/0.7 improvement, from E's 32.9/28.8 to F's 33.6/29.5. The schedule change has a precise interpretation: for $p \sim \text{Beta}(2,1)$, $\Pr(p \geq 0.9) = 0.19$, whereas the mixture gives

$$\Pr_{\text{mixture}}(p \geq 0.9) = 0.3(1-0.9^2) + 0.2(1-0.9^5) + 0.5 \approx 0.639. \tag{15}$$

Thus training samples high mask ratios more frequently. The mean mask ratio also rises from $2/3$ for the Beta schedule to approximately 0.842 for the mixture (versus 0.5 for Uniform). This calculation concerns the sampled ratio, not the realized fraction of masked tokens after token sampling and seed handling, and documents a change in the training distribution rather than a direct estimate of per-token reconstruction difficulty. The observed gain is limited to the reported run. Token-level validation diagnostics are given separately in Appendix A.13.

4.4 Qualitative Results

Figure 3 illustrates recovery of the dominant centerline layout together with residual endpoint displacement, local shape errors, and differing short branches. The examples illustrate lane-level geometric reconstruction, while directed lane relations are evaluated separately through TOP_{ll} in Table 2. These cases also clarify the scope of the quantitative results: the method preserves lane identity across records, but local geometric deviations and missing short branches remain possible under free-running decoding.



Figure 3: **Lane geometry on nuScenes subset B.** Two scenes, each shown as ground truth, prediction, and overlay from left to right. Instance colors and point markers show centerline geometry in the first two panels of each triplet; overlays use green for ground truth and red for predictions.

4.5 Output Compression and Throughput

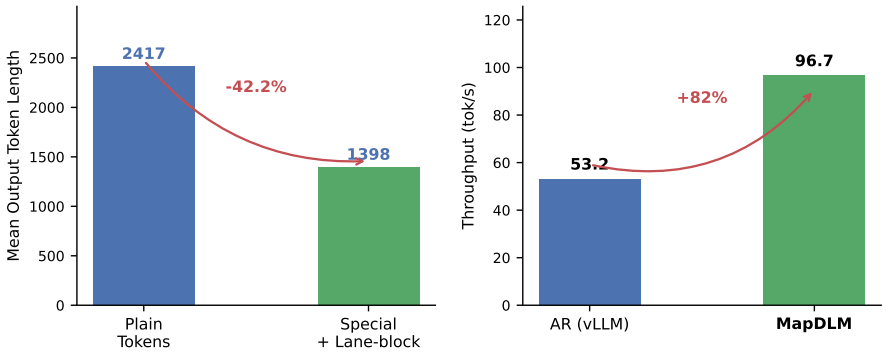

Figure 4: **Output compression and reported decoder throughput.** Left: mean response length. Right: tokens per second under the respective serving implementations; token granularity differs between systems.

Atomic coordinate tokens and the lane-aware layout reduce mean response length from 2,417 to 1,398 tokens (-42.2%). Geometry is packed into the same 32-token block used for iterative recovery. A vLLM-served Qwen2.5-VL-3B AR baseline Kwon et al. (2023) reports 53.2 tok/s, while MapDLM reports 96.7 tok/s ($+82\%$). These measurements document output compactness and reported token throughput, but do not establish a scene-level speedup: token vocabularies differ in semantic granularity, and the measurement excludes LiDAR accumulation, BEV rendering, end-to-end latency, memory, and energy. Response length and token throughput are therefore separate operational measurements under the reported serving setups; they do not isolate a decoder-only speed advantage.

5 Conclusion

MapDLM adapts a pretrained block-diffusion VLM to local lane-graph mapping. A topology-first response and lane-aware 32-token blocks associate coordinates with persistent lane IDs, while geometric objectives supervise complete centerlines across record boundaries. On AV2 and nuScenes, the complete system improves detection and directed topology over the reported AR controls under a common LiDAR-BEV protocol. Ablations document an interaction between supervision allocation and noise scheduling, and the representation shortens mean responses by 42.2%.

MapDLM provides a reference formulation for generative mapping through a shared interface for graph relations and variable-length geometry. Broader benchmark competitiveness and attribution to lane alignment require matched baselines and controlled experiments; token throughput does not establish end-to-end latency. The evaluated model is a map-reconstruction component, not a validated planner. Residual endpoint and branch errors motivate independent geometry and connectivity checks before downstream use.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 2021.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Julian Brandes, Philipp Crocoll, and Wolfram Burgard. Mapdreamer: Aerial imagery conditioned latent diffusion for lane-level map generation. *arXiv preprint arXiv:2607.01370*, 2026. URL https://arxiv.org/abs/2607.01370.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15661–15670, 2021.

Keshigeyan Chandrasegaran, Armin W. Thomas, Jerome Ku, Jae Myung Kim, Federico Berto, Garyk Brixi, Eric Nguyen, Stefano Massaroli, and Michael Poli. Rnd1: Simple, scalable ar-to-diffusion conversion. https://www.radicalnumerics.ai/blog/rnd1, Oct 2025. Radical Numerics blog.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 364–373, 2021.

Jiahui Fu, Zehao Huang, Han Li, Naiyan Wang, and Si Liu. Generative lane topology reasoning via autoregressive model with geometry prior. *arXiv preprint arXiv:2606.31814*, 2026.

Yanping Fu, Wenbin Liao, Xinyuan Liu, Hang Xu, Yike Ma, Yucheng Zhang, and Feng Dai. Topologic: An interpretable pipeline for lane topology reasoning on driving scenes. *Advances in Neural Information Processing Systems*, 37:61658–61676, 2025.

Yifan He. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Chen Huang. Tidar: Think in diffusion, talk in autoregression. *arXiv preprint arXiv:2511.08923*, 2025.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. *arXiv preprint arXiv:2510.12798*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Han Li, Zehao Huang, Zitian Wang, Wenge Rong, Naiyan Wang, and Si Liu. Enhancing 3d lane detection and topology reasoning with 2d lane priors. *arXiv preprint arXiv:2406.03105*, 2024.

Han Li, Shaofei Huang, Longfei Xu, Yulu Gao, Beipeng Mu, and Si Liu. Ratopo: Improving lane topology reasoning via redundancy assignment. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 777–786, 2025.

Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, et al. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023a.

Tianyu Li, Peijin Jia, Bangjun Wang, Li Chen, Kun Jiang, Junchi Yan, and Hongyang Li. Lanesegnet: Map learning with lane segment perception for autonomous driving. *arXiv preprint arXiv:2312.16108*, 2023b.

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. *arXiv preprint arXiv:2208.14437*, 2022.

Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pp. 22352–22369. PMLR, 2023.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2024.

Jiachen Lu, Renyuan Peng, Xinyue Cai, Hang Xu, Hongyang Li, Feng Wen, Wei Zhang, and Li Zhang. Translating images to road network: A non-autoregressive sequence-to-sequence approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23–33, 2023.

Jiachen Lu, Ming Nie, Bozhou Zhang, Renyuan Peng, Xinyue Cai, Hang Xu, Feng Wen, Wei Zhang, and Li Zhang. Translating images to road network: A sequence-to-sequence perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Shen Nie, Feng Zhu, Chao You, Xiaolong Zhang, Jing Ou, Tian Wu, Jun Lin, Yixiao Li, Zhiqiang Ma, and Yang Yu. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Renyuan Peng, Xinyue Cai, Hang Xu, Jiachen Lu, Feng Wen, Wei Zhang, and Li Zhang. Lanegraph2seq: Lane topology extraction with language model via vertex-edge encoding and connectivity enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4497–4505, 2024.

Limeng Qiao, Yongchao Zheng, Peng Zhang, Wenjie Ding, Xi Qiu, Xing Wei, and Chi Zhang. Machmap: End-to-end vectorized solution for compact hd-map construction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Lipeng Wan, Junjie Ma, Jianhui Gu, Zeyang Liu, Xuyang Lu, and Xuguang Lan. Geoblock: Inferring block granularity from dependency geometry in diffusion language models. *arXiv preprint arXiv:2603.26675*, 2026.

Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, et al. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. *Advances in Neural Information Processing Systems*, 36:18873–18884, 2023.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. Locateanything: Fast and high-quality vision-language grounding with parallel box decoding. *arXiv preprint arXiv:2605.27365*, 2026.

Zijie Wang, Weiming Zhang, Wei Zhang, Xiao Tan, Hongxin Liu, Yaowei Wang, and Guanbin Li. Lanediffusion: Improving centerline graph learning via prior injected bev feature generation. *arXiv preprint arXiv:2511.06272*, 2025.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

Wilson Wongso, Lihuan Li, Arian Prabowo, Xiachong Lin, Baiyu Chen, Hao Xue, and Flora D. Salim. Trajdlm: Topology-aware block diffusion language model for trajectory generation. *arXiv preprint arXiv:2605.10020*, 2026.

Chengyue Wu, Shiyi Lan, Yonggan Fu, Sensen Gao, Jin Wang, Jincheng Yu, Jose M. Alvarez, Pavlo Molchanov, Ping Luo, Song Han, Ligeng Zhu, and Enze Xie. Fast-dvlm: Efficient block-diffusion vlm via direct conversion from autoregressive vlm. *arXiv preprint arXiv:2604.06832*, 2026.

Dongming Wu, Jiahao Chang, Fan Jia, Yingfei Liu, Tiancai Wang, and Jianbing Shen. Topomlp: An simple yet strong pipeline for driving topology reasoning. *arXiv preprint arXiv:2310.06753*, 2023.

Mengwei Xie, Shuang Zeng, Xinyuan Chang, Xinran Liu, Zheng Pan, Mu Xu, and Xing Wei. Seqgrowgraph: Learning lane topology as a chain of graph expansions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27166–27175, 2025.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. Unveiling the potential of diffusion large language model in controllable generation. *arXiv preprint arXiv:2507.04504*, 2025.

Zhenhua Xu, Yuxuan Liu, Yuxiang Sun, Ming Liu, and Lujia Wang. Centerlinedet: Centerline graph detection for road lanes with vehicle-mounted sensors by transformer for hd map generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

Yiming Yang, Yueru Luo, Bingkun He, Erlong Li, Zhipeng Cao, Chao Zheng, Shuqi Mei, and Zhen Li. Topo2seq: Enhanced topology reasoning via topology sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9318–9326, 2025a.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025b.

Yujian Yuan, Changjie Wu, Xinyuan Chang, Sijin Wang, Hang Zhang, Shiyi Liang, Shuang Zeng, and Mu Xu. Unimapgen: A generative framework for large-scale map construction from multimodal data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12277–12285, 2026.

Kewei Zhang, Jin Wang, Sensen Gao, Chengyue Wu, Yulong Cao, Songyang Han, Boris Ivanovic, Langechuan Liu, Marco Pavone, Song Han, Daquan Zhou, and Enze Xie. Fast-ddrive: Efficient block-diffusion vlm for autonomous driving. *arXiv preprint arXiv:2605.23163*, 2026. URL https://arxiv.org/abs/2605.23163.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

A Technical Appendices and Supplementary Material

This appendix documents the representation, reported training settings, and decoding procedure of MapDLM. Coordinates are discrete BEV-grid values.

A.1 System Prompt and Input Format

The model is given one BEV image rendered from pose-aligned accumulated LiDAR and a fixed task prompt. The prompt is used in both training and inference; it defines the image-aligned coordinate convention and asks the assistant to emit a lane graph in the MapDLM grammar. No ground-truth lane information is included in the prompt.

> You are a traffic engineer and autonomous driving perception analyst. You are given a bird's-eye-view image rendered from accumulated LiDAR. Predict the directed lane graph as MapDLM structured tokens. The BEV map uses an image-aligned coordinate system with origin at the top-left corner. The horizontal coordinate u increases from left to right and the vertical coordinate v increases from front to back. The ego vehicle is at the image center. Coordinates must remain within the BEV extent and the output must follow the lane-aware topology and geometry grammar.

The experiments use a 100×50 m region at 0.1 m/px, giving $H = 1000$, $W = 500$ and ego image coordinates $(u_0, v_0) = (250, 500)$.

A.2 Lane-Aware Topology and Geometry Serialization

The response follows the 32-token block contract in the main paper:

$$\langle \mathrm{MAP_S} \rangle \; \mathcal{H} \; \mathcal{G} \; \mathcal{A} \; \mathcal{D}_1 \cdots \mathcal{D}_M \; \langle \mathrm{MAP_E} \rangle, \tag{16}$$

where $\mathcal{H}$ is the directed topology header, $\mathcal{G}$ is the predicted lane inventory, and $\mathcal{A}$ aligns the geometry to a block boundary. A header chain is

$$\mathcal{T}_k = \langle \mathrm{CH_S} \rangle \langle \mathrm{C_k} \rangle \langle \mathrm{L_{k,1}} \rangle \cdots \langle \mathrm{L_{k,m_k}} \rangle \langle \mathrm{CH_E} \rangle. \tag{17}$$

Consecutive IDs define directed edges, and repeated IDs across chains express branching or merging. The inventory also allows a lane to be represented without an incident edge. Lane count is implicit in the inventory; redundant lane-count and all-lane summary fields are omitted.

Every geometry block contains two equal-capacity lane records:

$$\mathcal{D}_j = [\mathcal{Q}_{2j-1} \mid \mathcal{Q}_{2j}], \qquad |\mathcal{D}_j| = 32. \tag{18}$$

Each 16-token record begins with $\langle \mathrm{LS} \rangle \langle \mathrm{L_{\ell(a)}} \rangle \langle \mathrm{XY} \rangle$, followed by up to six coordinate pairs and PAD tokens filling its allotted positions. Longer centerlines occupy consecutive records with the same lane ID. An all-PAD record completes the final block if needed. Coordinates belonging to repeated IDs are concatenated in record order to recover a complete lane. At evaluation, this reconstruction uses the generated IDs and coordinate tokens.

A.3 Coordinate System Transformation

For serialization, we use an image-aligned BEV convention in which x_{BEV} points forward and y_{BEV} points to the right after conversion from the dataset's native frame. We render a BEV image with width W, height H, and resolution $\delta = 0.1$ m/px. A point is converted into image-aligned coordinates by

$$u = \left\lfloor \frac{W}{2} + \frac{y_{\mathrm{BEV}}}{\delta} \right\rfloor, \tag{19}$$

$$v = \left\lfloor \frac{H}{2} - \frac{x_{\mathrm{BEV}}}{\delta} \right\rfloor. \tag{20}$$

The resulting frame has its origin at the upper left. For the 100×50 m region used here, the BEV raster has height $H = 1000$ and width $W = 500$ pixels. Points outside the raster are discarded before tokenization.

Every scalar coordinate is represented by one special token, $\langle 0 \rangle$ through $\langle 999 \rangle$. The ordered value index, rather than the tokenizer's internal token-ID order, defines numeric proximity in all coordinate-specific losses. The shared vocabulary covers both axes, while valid horizontal targets occupy only bins 0–499. Vocabulary membership alone therefore does not guarantee an in-range horizontal prediction.

A.4 Special Token Vocabulary

Table 4 lists the registered MapDLM token families. Registration does not require every family to occur in every response; the response structure is specified in Eqs. 16–18. Each token is registered

Table 4: **MapDLM special-token families.** Structural tokens delimit the map, topology header, lane inventory, lane records, relations, and padding regions. Counts correspond to the reported tokenizer configuration.

Family	Token range	Count
Coordinates	$\langle 0\rangle$–$\langle 999\rangle$	1000
Structural	map/header/group/chain/padding tokens	32
Lane IDs	$\langle L_0\rangle$–$\langle L_{99}\rangle$	100
Group IDs	$\langle G_0\rangle$–$\langle G_{63}\rangle$	64
Junction IDs	$\langle J_0\rangle$–$\langle J_{31}\rangle$	32
Chain IDs	$\langle C_0\rangle$–$\langle C_{63}\rangle$	64
Directions	$\langle$DIR_00$\rangle$–$\langle$DIR_15$\rangle$	16
Total defined MapDLM tokens		1308

as an atomic additional special token, preventing subword fragmentation of coordinates, IDs, and structural delimiters.

A.5 Data-to-Token Conversion

The data conversion pipeline has four deterministic stages. First, lane centerlines are transformed into the image-aligned coordinate frame in Section A.3 and clipped to the BEV extent. Second, retained lanes are assigned normalized lane IDs and directed predecessor–successor paths are extracted as Header chains. Third, centerlines are encoded as lane records carrying up to six coordinate pairs, with longer lanes retaining their ID across records. Finally, pairs of records are packed into the 32-token blocks in Eq. 18, and alignment padding is applied to the prefix and final block.

At evaluation time, the inverse parser collects records with a shared lane ID, removes padding, and concatenates their coordinate pairs. Header chains are converted back into directed lane relations. The parser is intentionally tolerant of incomplete output: it first looks for complete delimited blocks and falls back to recoverable geometry fragments when a closing token is absent. This behavior increases the amount of valid geometry available for evaluation, but does not fabricate lanes or edges absent from the generated sequence.

A.6 Coordinate-Value Sinusoidal Encoding (CVSE)

Coordinate values are discrete tokens but have a continuous spatial order. MapDLM therefore augments the learned embedding of coordinate value q with a fixed sinusoidal value feature:

$$\mathbf{e}_{\text{coord}}(q) = \mathbf{e}_{\text{learned}}(q) + \alpha \boldsymbol{\phi}_{\sin}(q), \quad \alpha = 0.3, \tag{21}$$

where, for embedding dimension D,

$$\boldsymbol{\phi}_{\sin}(q, 2j) = \sin\left(q/10000^{2j/D}\right), \tag{22}$$

$$\boldsymbol{\phi}_{\sin}(q, 2j+1) = \cos\left(q/10000^{2j/D}\right). \tag{23}$$

The feature is added only to the input-side embeddings of the 1,000 coordinate tokens. It has no trainable parameters, does not change checkpoint tensor shapes, and does not affect the mask-token embedding. Since its inner product depends on coordinate separation, it provides a value-indexed basis for both local and long-range numeric relationships.

Coordinate values are supervised by *Anchor-Point Gaussian Loss* (APG Loss); separate geometric terms supervise lane geometry. Let z_i be the prediction logits for a supervised coordinate position and q_i^* its target value. We use a local one-dimensional Gaussian target over coordinate bins,

$$\tilde{p}(q \mid q_i^*) \propto \mathbf{1}[|q - q_i^*| \le 2] \exp\left(-\frac{(q-q_i^*)^2}{2\sigma^2}\right), \quad \sigma = 1. \tag{24}$$

Interior coordinate positions use a hard/soft mixture with $\alpha_i = 0.2$, allowing small nearby-bin deviations; the first and last coordinate pair of each polyline use $\alpha_i = 0$, i.e., hard cross-entropy only, to retain exact target labels at connectivity-sensitive endpoints. Endpoints are defined on the complete lane after records are reassembled, not at every record boundary. The sparse formulation uses the five-bin neighborhood in Eq. 24, not a dense two-dimensional soft target.

For a coordinate distribution p_i over the 1,000 value bins, the expected scalar coordinate is $\hat{q}_i = \sum_{q=0}^{999} q\, p_i(q)$. The expected-value term regresses $\hat{q}_i$ to q_i^*; this is a differentiable expectation, rather

Table 5: **Single-distribution SANS schedule.** Geometry receives the highest average corruption so the model is exposed to difficult coordinate recovery states.

Section	Mask-ratio distribution	Mean
Geometry	$\mathrm{Beta}(2, 1)$	0.67
Summary	$\mathrm{Beta}(1, 1.5)$	0.40
Topology	$\mathrm{Beta}(1, 2)$	0.33
Shape	$\mathrm{Beta}(0.5, 3)$	0.14

than the committed discrete token used at inference. Geometric terms operate after scalar predictions are paired as 2D points. The symmetric Chamfer component for predicted and target polylines, $\hat{\mathcal{P}}$ and $\mathcal{P}^*$, is

$$\begin{aligned}\mathcal{L}_{\mathrm{ch}} = \frac{1}{2.5}\frac{1}{2}\Bigg(&\frac{1}{|\hat{\mathcal{P}}|}\sum_{\hat{p}}\min\left(\min_{p^*}\|\hat{p}-p^*\|_2, 200\right)\\ &+\frac{1}{|\mathcal{P}^*|}\sum_{p^*}\min\left(\min_{\hat{p}}\|p^*-\hat{p}\|_2, 200\right)\Bigg).\end{aligned} \tag{25}$$

The full model combines hard/soft coordinate loss, expected-coordinate regression, pointwise ℓ_2, clipped Chamfer, tangential direction, and one-sided entropy terms. With $\gamma_{\mathrm{ch}} = 0.1$, token supervision remains mask-selective on the noisy branch, while geometry terms use the complete polyline from the same forward pass.

The complete objective matches Eq. 11:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\mathrm{base}} &+ \lambda_{\mathrm{sec}}\mathcal{L}_{\mathrm{sec}}\\ &+ \lambda_{\mathrm{lane}}(\mathcal{L}_{\mathrm{lane}}^{\mathrm{clean}} + \lambda_{\mathrm{noisy}}\mathcal{L}_{\mathrm{lane}}^{\mathrm{noisy}}) + \lambda_{\mathrm{pad}}\mathcal{L}_{\mathrm{pad}},\end{aligned} \tag{26}$$

with $\lambda_{\mathrm{sec}} = 0.1$, $\lambda_{\mathrm{lane}} = 0.2$, $\lambda_{\mathrm{noisy}} = 0.5$, and $\lambda_{\mathrm{pad}} = 0.02$. Base token CE is retained. The separate PAD term teaches recovery of masked padding without including PAD in the base CE supervision set. The AR controls use token CE and a coordinate-space L1 objective named `coord_soft`; that regression objective should not be confused with APG's Gaussian soft-label CE.

A.7 Semantic-Adaptive Noise Scheduling

Fast-dVLM denoises a physical block from masked states. MapDLM assigns a semantic type to each effective training partition using a majority vote over its token roles, including scaffold positions. The mask ratio is sampled from the distribution for its dominant type. In the final training stage, this partition is the full 32-token block. The single-distribution SANS configuration is shown in Table 5.

The full model (F) uses a mixture for geometry; E uses the single-Beta geometry schedule. The full-model geometry schedule is

$$0.3\,\mathrm{Beta}(2, 1) + 0.2\,\mathrm{Beta}(5, 1) + 0.5\,\mathcal{U}(0.9, 1.0), \tag{27}$$

which explicitly covers the nearly fully masked states encountered at the beginning of iterative decoding. The section-weighted CE applies weights 6.0, 3.0, 1.5, and 1.0 to geometry, topology, shape, and summary tokens, respectively, with $\lambda_{\mathrm{sec}} = 0.1$.

A.8 Block-Size Annealing Curriculum

Serialization, padding, and the free-running decoder are all organized around the fixed 32-token block, but training does not start from the full block size. Let $r \in [0, 1]$ be the training progress reported by the trainer,

$$r = \mathrm{clip}\Big(\frac{g}{G},\, 0,\, 1\Big), \tag{28}$$

where g is the optimizer step and G the planned maximum number of steps. For runs scheduled by an epoch budget rather than a step budget, g/G is replaced by e/E, the fractional number of completed epochs. The effective denoising block size is

$$b(r) = 2^{\,2+\min\left(\lfloor\sqrt{r}\,K_{\mathrm{bd}}\rfloor,\, K_{\mathrm{bd}}-1\right)}, \qquad K_{\mathrm{bd}} = \log_2 B - 1, \tag{29}$$

where $B = 32$ is the fixed serialization and inference block size. The effective subwindow $b(r)$ increases during training and reaches B at $r \geq 0.5625$. It controls the attention partition and the

Table 6: **Common training configuration.**

Parameter	Value	Parameter	Value
Backbone	Fast-dVLM-3B	Optimizer	AdamW
Training precision	bf16	LR schedule	cosine
Peak learning rate	2×10^{-5}	Warmup ratio	0.1
GPUs	8	Micro-batch/GPU	1
Gradient accumulation	1	Training epochs	20
Response block size	32	Max. sequence length	3040
Gradient checkpointing	enabled	Denoising steps	12
Lane records per block	2	Max. pairs per record	6
λ_{lane}	0.2	λ_{sec}	0.1
Soft-target σ/radius	1.0/2 bins	Sinusoidal scale α	0.3
PAD CE weight	0.02	Entropy coefficient	0.01

granularity at which mask ratios are sampled, while the serialized lane-record layout remains fixed. Early subwindows can divide a lane record; alignment of the effective training partition with the inference block holds in the final stage. This stage occupies $1 - 0.5625 \approx 44\%$ of the 20-epoch schedule, beginning during epoch 12. All reported checkpoints come from this final stage. The term "32-token block" elsewhere refers to the fixed target unit, not to every intermediate training subwindow.

A.9 Training Configuration

Table 6 records the common training settings. All model components, including the vision encoder, multimodal projector, and language backbone, are trainable. The dynamic loss balancer is implemented for diagnostics but disabled for all reported results.

All main and ablation experiments use 20 epochs and the curriculum in Section A.8.

A.10 Structure-Aware Block-Diffusion Decoding

At inference, each 32-token block is initialized from a causal seed followed by masked positions and denoised for $T = 12$ iterations. At every iteration, the decoder ranks positions by confidence and commits an annealed number of predictions. A slot is classified from its nearest committed left context: slots following $\langle \text{XY} \rangle$, a coordinate, or padding are treated as coordinate slots; all other slots are structural. Type-compatible predictions are committed before type-incompatible ones. This is a left-context type preference, not a hard grammar constraint: in particular, a structural position after a PAD token can receive a coordinate preference. The rule does not guarantee valid record boundaries, inventory consistency, or axis-specific coordinate ranges. The original language-model head is retained, with no extra verification network.

When a block is complete, it becomes causal context for the next block and the decoder reuses the KV cache. Generation stops at $\langle \text{MAP_E} \rangle$ or the response-length limit. The final output is parsed as described above; no learned post-processing or cross-block revision is applied.

A.11 Evaluation Protocol and Reproducibility

We use OpenLane-V2 lane annotations Wang et al. (2023) on AV2 subset A and nuScenes subset B. As stated in Section 4.1, evaluation discards height and compares lane geometry in the BEV plane. Directed topology is scored using the associated lane matches. We report DET_l, TOP_{ll}, and the lane-only aggregate L-OLS defined in Eq. 14. Both components are normalized to $[0, 1]$ before aggregation; all reported scores are on the percentage scale. For in-house systems, these scores describe the BEV adaptation rather than the full-task or official 3D OpenLane-V2 evaluation. The BEV input is rendered at 0.1 m/px from pose-aligned accumulated LiDAR; centerlines and directed links are transformed and clipped to the same region before serialization.

UniMapGen's class-token confidence construction is not assumed to apply to MapDLM's lane-ID and topology-chain grammar; the evaluation differences are described in Section 4.1.

Reproducing the absolute scores requires the same evaluation implementation, including lane and edge confidence construction, resampling and matching settings, temporal accumulation window, and scene-level split. The reported comparisons use the common in-house pipeline; the planar metric names alone do not specify these implementation choices. Decoder throughput excludes LiDAR accumulation and BEV rendering. Interpretation of the reported throughput additionally depends on hardware, precision, batch size, output lengths, and KV-cache behavior.

Table 7: **Published camera-based reference results.** Metric names follow the cited results. Inputs and geometric evaluation differ from our BEV setting; these entries do not support a matched comparison to MapDLM.

	AV2 subset A		nuScenes subset B	
Method	DET_l	TOP_{ll}	DET_l	TOP_{ll}
VectorMapNet Liu et al. (2023)	11.1	2.7	3.5	–
STSU Can et al. (2021)	12.7	2.9	8.2	–
MapTR Liao et al. (2022)	17.7	5.9	15.2	–
TopoNet Li et al. (2023a)	28.6	10.9	24.3	6.7
Topo2D Li et al. (2024)	29.1	22.3	–	–
TopoMLP Wu et al. (2023)	28.5	21.7	26.6	19.8
LaneSegNet Li et al. (2023b)	31.1	25.3	27.5	24.9
TopoLogic Fu et al. (2025)	29.9	23.9	25.9	21.6

A.12 Published Results for Context

Table 7 collects published DET_l/TOP_{ll} components for context. These results retain their respective camera-input and geometric evaluation protocols and are excluded from the main comparison tables. LaneDiffusion, LaneGraph2Seq, and UniMapGen are discussed as related work, not evaluated as rerun baselines in our pipeline.

A.13 Token-Level Validation Diagnostics

Table 8 records coordinate and structure/topology token accuracy on the internal AV2 validation subset. They describe masked/teacher-forced token prediction, not free-running lane reconstruction. Their product is not a measured joint success probability, so only the two directly reported accuracies are retained. These diagnostic configurations are named by their training settings; no additional correspondence to individual-loss ablations is asserted.

Table 8: **Internal token-prediction diagnostics.** These accuracies complement the free-running BEV metrics and do not estimate scene-level or lane-level failure rates.

Configuration	Coordinate accuracy	Structure/topology accuracy
Coord Token + Lane Loss	89.03%	**95.43%**
+ SANS noise	87.10%	95.13%
+ weighted tokens	97.96%	93.39%
Full model	**98.53%**	95.23%

AI use statement

In this work, we used generative AI tools for text polishing and manuscript-wide reformatting. Specifically, these tools were used to improve language clarity, grammar, presentation consistency, and the overall formatting of the paper. We have not used generative AI tools to generate or alter experimental results, datasets, evaluation metrics, or empirical claims, and other required disclosure tasks are not applicable to this work. Additionally, generative AI tools were used to assist with improving the readability and organization of the manuscript. All AI-assisted content was carefully reviewed, verified, and revised by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI. See the ICLR 2027 AI Policy for Authors for more details.

Reproducibility statement

To support reproducibility, the appendix provides the implementation code for the proposed method. The main paper describes the task formulation, model design, lane-aligned block diffusion procedure, training objective, and experimental protocol. The appendix further documents the implementation details needed to reproduce the reported experiments. Together, these materials provide the necessary information for reproducing the method and evaluating it under the reported settings.