

Article

Local Lane Graph Conditioning as a General Inductive Bias for Trajectory Prediction: A Multi-Architecture Study on the Waymo Open Motion Dataset

Xingnan Zhou ¹ and Ciprian Alecsandru ^{1,*}

¹ Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

* Correspondence: ciprian.alecsandru@concordia.ca (C.A.)

Academic Editors: Jinlei Zhang and Wei Ma

Version February 10, 2026 submitted to Sustainability

Abstract: Accurate trajectory prediction is critical for both autonomous driving safety and energy-efficient motion planning in sustainable urban mobility systems. While state-of-the-art methods employ complex architectures with hundreds of input features, the contribution of individual components—particularly local road topology—remains difficult to isolate. This study investigates whether local lane graph conditioning provides a *general, architecture-agnostic* improvement, using only minimal position inputs and a lightweight lane encoder. We propose a *waterflow* lane graph extraction method that constructs an ego-centric lane topology through breadth-first traversal of the HD map, fusing lane features into trajectory encoders via cross-attention. We evaluate across two encoder architectures (LSTM and Transformer), two prediction horizons (3 s and 8 s), and both single- and multi-modal ($K=6$) settings on the Waymo Open Motion Dataset (89,258 scenarios). Lane graph conditioning consistently improves accuracy: +9.3% ADE at 3 s (LSTM, $p=0.0071$), +27.3% minADE at 8 s (LSTM, $K=6$), and +32.0% ADE at 8 s (Transformer)—with approximately 8% additional parameters for LSTM. Error decomposition reveals improvements in both lateral (+26.5%) and longitudinal (+25.4%) components, with endpoint lateral error showing the largest reduction (+30.5%). Our lane-conditioned LSTM achieves a minADE of 1.34 m at 8 s, matching the Waymo official LSTM baseline that uses the full feature set, suggesting that local lane structure can substitute for hand-engineered input features. These findings establish local lane graph conditioning as a lightweight, general-purpose module for trajectory prediction in safety-critical urban environments.

Keywords: trajectory prediction; lane graph; autonomous driving; Waymo Open Motion Dataset; LSTM; Transformer; multi-modal prediction; traffic safety; sustainable transportation

1. Introduction

Urban intersections represent one of the most safety-critical environments in road transportation networks. According to the National Highway Traffic Safety Administration (NHTSA), approximately 40% of all police-reported crashes in the United States occur at or near intersections [1], resulting in significant human, economic, and environmental costs. Each collision-induced traffic disruption propagates congestion through the surrounding network, increasing vehicle idle times, fuel consumption, and greenhouse gas emissions [2]. For connected and automated vehicles, Taiebat et al. [3] identify trajectory prediction as a key enabling technology with cascading implications for energy efficiency, emissions reduction, and traffic safety. Consequently, improving the ability of

30 intelligent transportation systems to anticipate vehicle movements is a key pathway toward more
31 sustainable urban mobility [4,5].

32 Trajectory prediction—the task of forecasting future positions of traffic agents given their observed
33 motion history and environmental context—is a foundational capability for autonomous vehicles
34 and advanced driver-assistance systems [6,7]. Accurate trajectory forecasts enable proactive safety
35 interventions such as emergency braking and cooperative maneuver planning, reducing collision risk
36 and its downstream sustainability impacts [8].

37 A key insight driving recent progress is that road map information—particularly lane geometry
38 and connectivity—provides strong *relational inductive biases* [9] for predicting where vehicles are
39 likely to travel [10–12]. Methods such as LaneGCN [11] use lane graphs as the primary backbone
40 representation, while HiVT [13] and PGP [14] hierarchically fuse lane topology with agent dynamics.
41 State-of-the-art methods such as MTR [15] and QCNet [16] achieve remarkable performance by jointly
42 encoding rich agent features (velocity, acceleration, heading, bounding box dimensions) with dense
43 map representations using large Transformer architectures. However, the entanglement of multiple
44 input features and architectural innovations makes it difficult to isolate the specific contribution of
45 lane topology information.

46 This paper addresses this gap through a controlled study that isolates the effect of local lane
47 graph conditioning. Unlike LaneGCN [11], which builds the entire architecture around lane graph
48 representations, we treat lane conditioning as a *modular plug-in* and evaluate it across multiple
49 backbone architectures to assess its generality. We deliberately employ simple backbones—an LSTM
50 encoder-decoder and a vanilla Transformer encoder—with minimal input features (2D position only),
51 so that any observed improvement can be directly attributed to the lane conditioning module. Our
52 key research question is: *Does local lane graph conditioning provide a consistent, architecture-agnostic*
53 *improvement for trajectory prediction?*

54 We answer this question affirmatively through comprehensive experiments on the Waymo
55 Open Motion Dataset [17], evaluating across two architectures, two prediction horizons, and both
56 single-modal and multi-modal prediction settings. Our contributions are as follows:

- 57 1. We propose a *waterflow* lane graph extraction method that constructs a local, ego-centric lane
58 topology through breadth-first traversal of the HD map, and a lightweight lane encoder with
59 graph message passing and cross-attention fusion.
- 60 2. We demonstrate that lane conditioning provides a **consistent, architecture-agnostic**
61 **improvement**: +9.3% ADE reduction for LSTM at 3 s ($p=0.0071$), +27.3% minADE reduction for
62 multi-modal LSTM at 8 s, and +32.0% ADE reduction for Transformer at 8 s.
- 63 3. We show that the benefit of lane conditioning **increases with prediction horizon** (from +9.3% at
64 3 s to +27.3% at 8 s), confirming that lane structure becomes increasingly valuable as kinematic
65 extrapolation degrades over longer horizons.
- 66 4. Through error decomposition analysis, we reveal that lane conditioning improves both lateral
67 (+26.5%) and longitudinal (+25.4%) error components, with the strongest improvement at
68 trajectory endpoints (lateral FDE: +30.5%).
- 69 5. We show that our lane-conditioned LSTM with $K=6$ modes achieves a minADE of 1.34 m at 8 s,
70 **matching the Waymo official LSTM baseline** [17] that uses the full feature set, while using only
71 2D position inputs plus local lane features.

72 By demonstrating that a lightweight lane conditioning module (<700,000 parameters) can match
73 the accuracy of models requiring rich hand-engineered features, our results point toward more
74 computationally efficient prediction systems—a direct contribution to sustainable autonomous driving
75 through reduced onboard energy consumption and accessible deployment on resource-constrained
76 platforms.

77 2. Related Work

78 2.1. Recurrent Approaches to Trajectory Prediction

79 The application of LSTM networks [18] to trajectory prediction was pioneered by Alahi et al. [19],
80 who introduced Social LSTM with a social pooling mechanism for pedestrian interactions. Park et
81 al. [20] adapted the encoder-decoder LSTM architecture for vehicle trajectory prediction, demonstrating
82 that sequence-to-sequence models effectively capture temporal dynamics. Deo and Trivedi [21]
83 proposed convolutional social pooling for highway lane-change prediction. These methods established
84 the core encoder-decoder paradigm upon which our LSTM baseline builds.

85 Graph-based extensions to recurrent models have been proposed to capture dynamic agent
86 interactions. Chandra et al. [22] employed graph-LSTMs with spectral clustering, while Li et al. [23]
87 introduced EvolveGraph for dynamic relational reasoning. Mo et al. [24] combined graph neural
88 networks with recurrent architectures for highway prediction. These works demonstrate the value of
89 structured relational reasoning but primarily target highway scenarios.

90 2.2. Transformer-Based Approaches

91 Transformer architectures [25] have achieved state-of-the-art performance on large-scale motion
92 forecasting benchmarks. Scene Transformer [26] proposed a unified multi-agent prediction architecture.
93 Wayformer [27] demonstrated efficient attention-based forecasting. MTR [15] introduced motion
94 transformers with global intention localization, and QCNet [16] proposed query-centric prediction.
95 HiVT [13] uses a hierarchical architecture that combines local agent–lane interactions with global
96 scene-level attention. These methods typically combine Transformer attention with rich input features
97 and map representations, making it difficult to attribute improvements to specific components. Notably,
98 Zeng et al. [28] showed that simple models can outperform Transformers for time series forecasting,
99 suggesting that the Transformer’s advantage depends on the availability of rich contextual information.
100 Our work complements these efforts by isolating the lane conditioning component within a simple
101 Transformer encoder.

102 2.3. Map-Aware and Lane-Conditioned Methods

103 The integration of HD map information has emerged as a critical factor in prediction performance.
104 VectorNet [10] proposed a unified vectorized representation for agent trajectories and map elements.
105 LaneGCN [11] introduced lane graph representations with graph convolutions [29] along lane
106 connectivity structures. LaneRCNN [30], from the same group, extended this with distributed
107 lane-centric representations and actor–lane interaction graphs. TNT [12] and DenseTNT [31] leveraged
108 lane centerlines as target candidates. LaPred [32] explicitly conditioned predictions on lane-level
109 features. PGP [14] conditions multi-modal predictions on discrete lane-graph traversals, treating
110 lane connectivity as a tree of possible goals—an approach philosophically similar to our workflow
111 extraction but using graph traversals for mode generation rather than feature conditioning. GANet [33]
112 uses lane-level goal areas for multi-modal forecasting, demonstrating that lane structure constrains
113 plausible endpoints.

114 While these methods demonstrate the value of lane information, they employ complex
115 architectures where the lane component is deeply integrated with the rest of the model, making
116 it difficult to isolate the lane contribution. In particular, LaneGCN builds the entire architecture around
117 the lane graph as the primary backbone, whereas we treat lane conditioning as a *modular plug-in* that
118 can be attached to arbitrary backbone architectures. This distinction is important: our controlled
119 study tests whether lane conditioning generalizes across fundamentally different encoder architectures
120 (recurrent and attention-based), using a message passing formulation [34] for lane feature propagation
121 and cross-attention [35] for fusion. By analogy with the relational inductive bias framework of Battaglia

¹²² et al. [9], our lane graph provides a structured prior that constrains the model’s hypothesis space
¹²³ without dictating the overall architecture.

¹²⁴ 2.4. Multi-Modal Prediction

¹²⁵ Vehicle trajectories at intersections are inherently multi-modal, as drivers may turn left, go
¹²⁶ straight, or turn right at the same intersection. Gupta et al. [36] introduced Social GAN, using
¹²⁷ generative adversarial training to produce diverse trajectory samples. Winner-takes-all (WTA)
¹²⁸ training [37] provides a simpler alternative, assigning each ground truth to the closest prediction
¹²⁹ mode and backpropagating through that mode only. Salzmann et al. [38] used conditional variational
¹³⁰ autoencoders in Trajectron++ for probabilistic multi-modal prediction. The Waymo Motion Prediction
¹³¹ Challenge evaluates methods using minADE and minFDE (minimum over K modes), making
¹³² multi-modal prediction essential for benchmark evaluation. We adopt WTA training with $K=6$
¹³³ modes to enable direct comparison with Waymo benchmarks.

¹³⁴ 3. Methodology

¹³⁵ This section describes the problem formulation, the waterflow lane graph, and the model
¹³⁶ architectures.

¹³⁷ 3.1. Problem Formulation

¹³⁸ Given the observed trajectory of an ego vehicle over $T_{\text{obs}} = 11$ timesteps (1.1 s at 10 Hz), the
¹³⁹ observed trajectories of up to $N = 10$ neighboring agents within a 30 m radius, and optionally a
¹⁴⁰ local lane graph, the goal is to predict the ego vehicle’s future trajectory. We evaluate two prediction
¹⁴¹ horizons: $T_{\text{pred}} = 30$ (3.0 s) and $T_{\text{pred}} = 80$ (8.0 s). All positions are in a bird’s-eye-view coordinate
¹⁴² frame centered on the ego vehicle’s last observed position and aligned with its heading direction.

¹⁴³ In the single-modal setting, the model outputs one predicted trajectory $\hat{\mathbf{Y}} \in \mathbb{R}^{T_{\text{pred}} \times 2}$. In
¹⁴⁴ the multi-modal setting, the model outputs $K = 6$ trajectory hypotheses $\{\hat{\mathbf{Y}}_k\}_{k=1}^K$ with associated
¹⁴⁵ confidence scores $\{c_k\}_{k=1}^K$, where $\sum_k c_k = 1$.

¹⁴⁶ 3.2. Waterflow Lane Graph Extraction

¹⁴⁷ To incorporate local road topology, we extract a structured lane graph from the HD map. We
¹⁴⁸ term this the *waterflow* graph because the extraction models the directional propagation of traffic
¹⁴⁹ flow potential from the ego vehicle outward through the lane connectivity network. Unlike standard
¹⁵⁰ undirected graph expansions, the waterflow traversal respects the directionality of lane successors
¹⁵¹ (forward connectivity) while also capturing lateral alternatives, mirroring how traffic flow possibilities
¹⁵² radiate outward from a vehicle’s current position.

¹⁵³ 3.2.1. Graph Construction

¹⁵⁴ The extraction proceeds in four stages:

- ¹⁵⁵ 1. **Ego Lane Identification.** The lane whose centerline passes closest to the vehicle’s last observed
¹⁵⁶ position (within 5 m) is selected as the ego lane. If no lane centerline lies within this threshold, the
¹⁵⁷ system falls back to the nearest lanes by Euclidean distance. In our signal-controlled subset, valid
¹⁵⁸ ego lanes are identified in >98% of scenarios, ensuring the lane conditioning module is active in
¹⁵⁹ the vast majority of cases.
- ¹⁶⁰ 2. **Breadth-First Expansion.** Starting from the ego lane, a 3-hop BFS traverses successor lanes
¹⁶¹ (forward connectivity), left-adjacent lanes, and right-adjacent lanes.
- ¹⁶² 3. **Truncation.** The subgraph is truncated to $L_{\text{max}} = 16$ lanes, prioritized by topological proximity.
- ¹⁶³ 4. **Feature Extraction.** For each lane ℓ , a feature vector $\mathbf{f}_\ell \in \mathbb{R}^{26}$ is computed.

¹⁶⁴ Algorithm 1 formalizes this procedure. The BFS traversal uses a visited set to naturally handle
¹⁶⁵ cycles (e.g., roundabouts), and the L_{max} bound in the outer loop ensures that the 16-lane limit takes

¹⁶⁶ precedence over the 3-hop limit when many adjacent lanes are present at wide intersections. Ties
¹⁶⁷ among equidistant lanes are broken by BFS visit order (first-in, first-out).

Algorithm 1: Waterflow Lane Graph Extraction

```

Input: HD map  $\mathcal{M}$ , ego position  $\mathbf{p}_{ego}$ ,  $h_{max}=3$ ,  $L_{max}=16$ 
Output: Lane features  $\mathbf{F} \in \mathbb{R}^{L_{max} \times 26}$ , adjacency  $\mathbf{A} \in \{0, 1\}^{L_{max} \times L_{max}}$ , mask  $\mathbf{m}$ 

1  $\ell_0 \leftarrow \arg \min_{\ell \in \mathcal{M}} \text{dist}(\text{centerline}(\ell), \mathbf{p}_{ego})$ ; // Ego lane
2 if  $\text{dist}(\text{centerline}(\ell_0), \mathbf{p}_{ego}) > 5 \text{ m}$  then
3   |  $\ell_0 \leftarrow \text{nearest lane by Euclidean distance}$ ; // Fallback
4 end
5  $\mathcal{Q} \leftarrow \{(\ell_0, 0)\}$ ; // Queue: (lane, hop count)
6  $\mathcal{V} \leftarrow \{\ell_0\}$ ; // Visited set
7 while  $\mathcal{Q} \neq \emptyset$  and  $|\mathcal{V}| < L_{max}$  do
8   |  $(\ell, h) \leftarrow \mathcal{Q}.\text{dequeue}()$ ;
9   | if  $h < h_{max}$  then
10    |   | foreach  $\ell' \in \text{Succ}(\ell) \cup \text{LeftAdj}(\ell) \cup \text{RightAdj}(\ell)$  do
11    |   |   | if  $\ell' \notin \mathcal{V}$  and  $|\mathcal{V}| < L_{max}$  then
12    |   |   |   |  $\mathcal{V} \leftarrow \mathcal{V} \cup \{\ell'\}$ ;
13    |   |   |   |  $\mathcal{Q}.\text{enqueue}((\ell', h + 1))$ ;
14    |   | end
15   | end
16 end
17 end
18 foreach  $\ell \in \mathcal{V}$  do
19   |  $\mathbf{f}_\ell \leftarrow [\mathbf{c}_\ell \parallel \mathbf{d}_\ell \parallel \mathbf{s}_\ell \parallel \mathbf{b}_\ell]$ ; // Eq. (1)
20 end
21 Construct  $\mathbf{A}$  from lane connectivity within  $\mathcal{V}$ ; set  $\mathbf{m}$ ;
22 return  $\mathbf{F}, \mathbf{A}, \mathbf{m}$ 

```

¹⁶⁸ Figure 1 illustrates the progressive expansion of the waterflow graph from the ego lane through
¹⁶⁹ three hops, showing how local lane topology is incrementally captured.

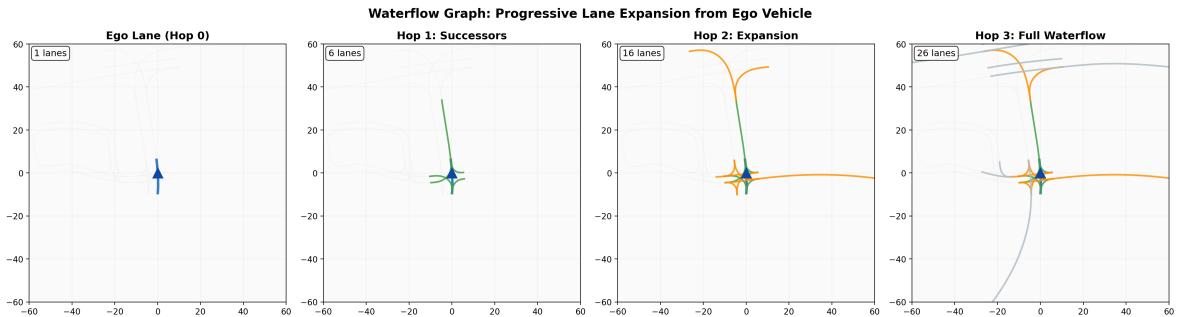


Figure 1. Waterflow lane graph extraction via breadth-first expansion. Starting from the ego lane (Hop 0), the graph progressively incorporates successor and adjacent lanes through 3 hops, capturing the local road topology relevant to trajectory prediction.

¹⁷⁰ 3.2.2. Lane Feature Representation

¹⁷¹ Each lane feature vector comprises:

$$\mathbf{f}_\ell = [\mathbf{c}_\ell \parallel \mathbf{d}_\ell \parallel \mathbf{s}_\ell \parallel \mathbf{b}_\ell], \quad (1)$$

172 where $\mathbf{c}_\ell \in \mathbb{R}^{20}$ contains the flattened (x, y) coordinates of 10 resampled centerline points in the
 173 ego-centric frame, $\mathbf{d}_\ell \in \mathbb{R}^2$ is the normalized direction vector, $s_\ell \in \mathbb{R}$ is the normalized lane length,
 174 and $\mathbf{b}_\ell \in \{0, 1\}^3$ encodes three boolean flags: ego lane, traffic signal controlled, and stop sign present.
 175

176 The lane adjacency matrix $\mathbf{A} \in \{0, 1\}^{L_{\max} \times L_{\max}}$ encodes undirected connectivity, and a validity
 mask $\mathbf{m}_\ell \in \{0, 1\}^{L_{\max}}$ indicates which slots contain valid lanes.

177 3.3. Architecture Overview

178 Figure 2 provides an overview of the four model variants evaluated in this study. All models
 179 share the same neighbor encoder, fusion layer, and CV-residual decoder; only the ego encoder (LSTM
 180 vs. Transformer) and the optional lane conditioning module differ. The lane conditioning module
 181 (highlighted in green) is a modular plug-in that can be attached to either backbone architecture,
 182 enabling a controlled ablation of its effect.

183 3.4. Model A: LSTM Baseline

184 The baseline model follows a standard encoder-decoder architecture with three components.

185 3.4.1. Ego Trajectory Encoder

186 The ego history is projected to a 64-dimensional embedding and processed by a 2-layer LSTM
 187 with hidden dimension $d_h = 128$:

$$\mathbf{e}_t = \text{ReLU}(\mathbf{W}_e \mathbf{x}_t + \mathbf{b}_e), \quad \mathbf{h}_t, \mathbf{c}_t = \text{LSTM}(\mathbf{e}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}). \quad (2)$$

188 3.4.2. Neighbor Context Encoder

189 Each neighbor's trajectory is independently encoded using a smaller 1-layer LSTM (hidden
 190 dimension 64). Neighbor representations are aggregated via masked max-pooling:

$$\mathbf{n}_{\text{ctx}} = \max_{i:m_i=1} (\text{LSTM}_{\text{nbr}}(\mathbf{X}_i^{\text{nbr}})) \in \mathbb{R}^{64}. \quad (3)$$

191 3.4.3. Fusion and CV-Residual Decoder

192 The ego hidden state and neighbor context are concatenated and fused:

$$\mathbf{z} = \text{ReLU}(\mathbf{W}_f [\mathbf{h}_{T_{\text{obs}}} \| \mathbf{n}_{\text{ctx}}] + \mathbf{b}_f) \in \mathbb{R}^{128}. \quad (4)$$

193 A three-layer MLP maps \mathbf{z} to displacement residuals $\Delta \mathbf{Y} \in \mathbb{R}^{T_{\text{pred}} \times 2}$, added to a constant-velocity
 194 baseline to form a CV-residual prediction:

$$\hat{\mathbf{y}}_t = (\mathbf{p}_{\text{last}} + \mathbf{v}_{\text{last}} \cdot t) + \Delta \mathbf{y}_t. \quad (5)$$

195 Here $\mathbf{v}_{\text{last}} = \mathbf{p}_{T_{\text{obs}}} - \mathbf{p}_{T_{\text{obs}}-1}$ is the velocity estimated from the last two observed positions. This
 196 CV-residual formulation provides a strong inductive bias for approximately linear motion segments,
 197 following the insight of Schöller et al. [39] that constant-velocity prediction serves as a surprisingly
 198 strong baseline.

199 3.5. Model B: Lane-Conditioned LSTM

200 The lane-conditioned model extends the LSTM baseline by incorporating local lane graph
 201 information through two additional components: a lane encoder with graph message passing [34]
 202 and cross-attention pooling [35]. The ego encoder, neighbor encoder, and CV-residual decoder remain
 203 identical, enabling a controlled ablation where any performance difference is directly attributable to
 204 the lane conditioning module.

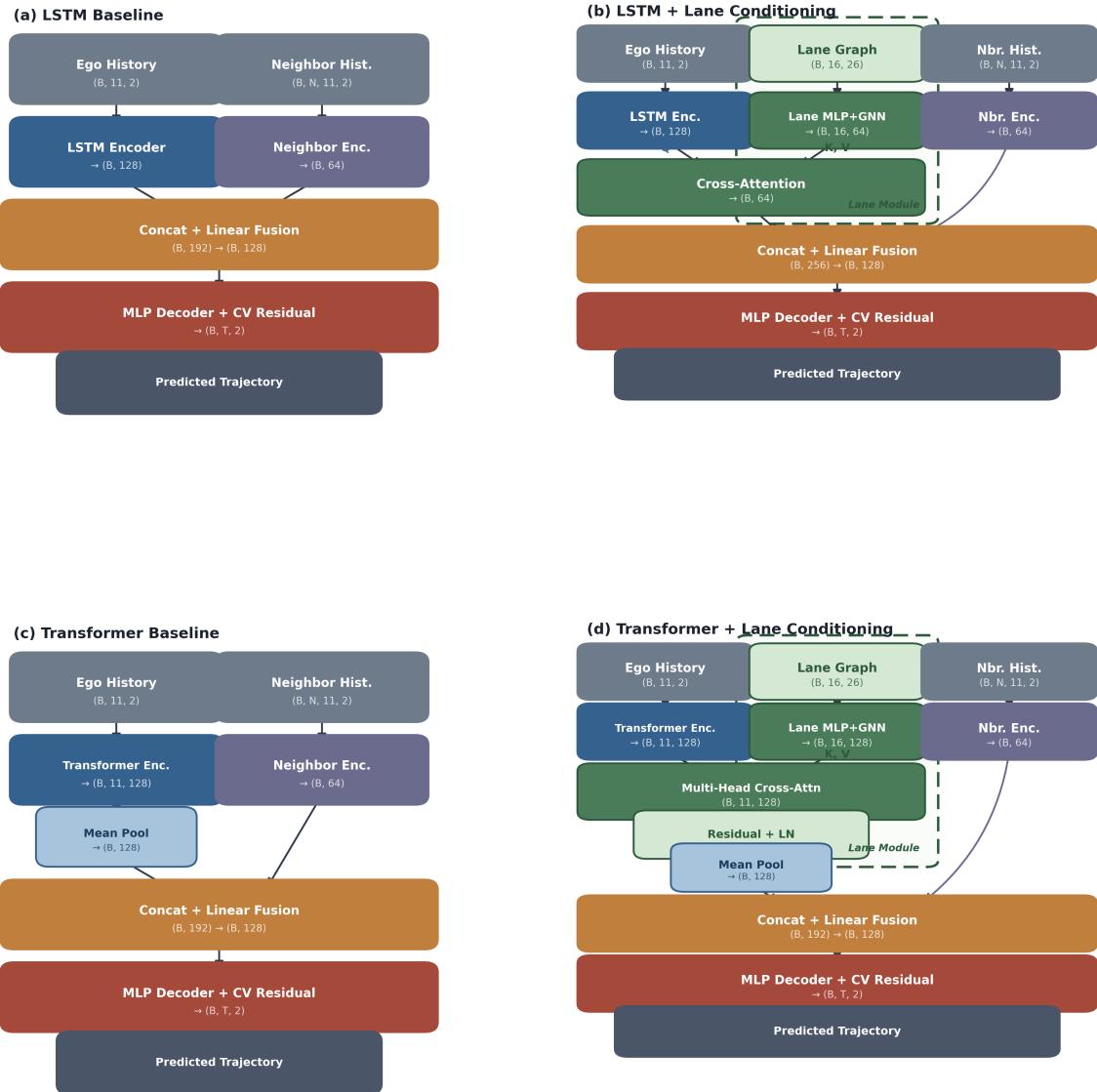


Figure 2. Architecture overview of the four model variants. **Left column:** baselines without lane information. **Right column:** lane-conditioned variants with the lane conditioning module (green dashed box) added as a modular plug-in. The LSTM variants (top row) use single-vector cross-attention pooling, while the Transformer variants (bottom row) use multi-head cross-attention over the full ego sequence to capture timestep-level lane–trajectory interactions.

205 3.5.1. Lane Encoder with Message Passing

206 Each lane feature vector is embedded into a 64-dimensional representation through a two-layer
207 MLP. Two rounds of message passing propagate structural information along the lane graph:

$$\mathbf{l}_\ell^{(k+1)} = \text{ReLU} \left(\mathbf{W}^{(k)} [\mathbf{l}_\ell^{(k)} \| (\tilde{\mathbf{A}} \mathbf{L}^{(k)})_\ell] \right), \quad (6)$$

208 where $\tilde{\mathbf{A}} = \mathbf{D}^{-1} \mathbf{A}$ is the degree-normalized adjacency matrix.

209 3.5.2. Cross-Attention Pooling

210 Lane embeddings are aggregated using cross-attention with the ego hidden state as the query:

$$\alpha_\ell = \frac{\exp(\mathbf{q}^\top \mathbf{k}_\ell / \sqrt{d_k})}{\sum_{j:m_j=1} \exp(\mathbf{q}^\top \mathbf{k}_j / \sqrt{d_k})}, \quad \mathbf{l}_{\text{ctx}} = \sum_\ell \alpha_\ell \mathbf{v}_\ell \in \mathbb{R}^{64}. \quad (7)$$

211 The lane context is concatenated with the ego and neighbor representations before fusion:

$$\mathbf{z} = \text{ReLU}(\mathbf{W}_f [\mathbf{h}_{T_{\text{obs}}} \| \mathbf{l}_{\text{ctx}} \| \mathbf{n}_{\text{ctx}}]) \in \mathbb{R}^{128}. \quad (8)$$

212 3.6. Model C: Transformer Baseline

213 To test the generality of lane conditioning across architecturally distinct backbones, we replace
214 the LSTM ego encoder with a Transformer encoder [25]. This allows us to determine whether the
215 benefits of lane conditioning are specific to the LSTM's sequential inductive bias or transfer to the
216 Transformer's attention-based paradigm.

217 3.6.1. Transformer Ego Encoder

218 The ego history is projected to $d_{\text{model}} = 128$ dimensions, combined with learnable positional
219 embeddings, and processed by a 2-layer Transformer encoder with 4 attention heads:

$$\mathbf{E} = \text{TransformerEncoder}(\mathbf{W}_p \mathbf{X}^{\text{ego}} + \mathbf{P}_{\text{pos}}) \in \mathbb{R}^{T_{\text{obs}} \times d_{\text{model}}}. \quad (9)$$

220 Unlike the LSTM, which compresses the sequence into a single hidden vector, the Transformer
221 produces a full sequence representation. The ego representation is obtained by mean pooling:

$$\mathbf{h}_{\text{ego}} = \frac{1}{T_{\text{obs}}} \sum_{t=1}^{T_{\text{obs}}} \mathbf{E}_t \in \mathbb{R}^{d_{\text{model}}}. \quad (10)$$

222 The neighbor encoder, fusion, and CV-residual decoder are identical to the LSTM baseline.

223 3.7. Model D: Lane-Conditioned Transformer

224 The key advantage of the Transformer for lane conditioning is that it produces a *full sequence*
225 *representation* $\mathbf{E} \in \mathbb{R}^{T_{\text{obs}} \times d_{\text{model}}}$, enabling multi-head cross-attention between all ego timesteps and lane
226 embeddings simultaneously. This contrasts with the LSTM variant (Model B), which compresses the
227 trajectory into a single vector before attending to lane features.

228 3.7.1. Multi-Head Cross-Attention

229 Lane features are projected to $d_{\text{model}} = 128$ dimensions via a two-layer MLP, followed by graph
230 message passing. The ego sequence queries the lane embeddings through multi-head cross-attention:

$$\mathbf{C} = \text{MultiHeadAttn}(\mathbf{Q}=\mathbf{E}, \mathbf{K}=\mathbf{L}, \mathbf{V}=\mathbf{L}) \in \mathbb{R}^{T_{\text{obs}} \times d_{\text{model}}}, \quad (11)$$

²³¹ with 4 attention heads. A residual connection and layer normalization produce the lane-conditioned
²³² sequence:

$$\tilde{\mathbf{E}} = \text{LayerNorm}(\mathbf{E} + \mathbf{C}). \quad (12)$$

²³³ The ego representation is obtained by mean pooling over $\tilde{\mathbf{E}}$, followed by the same neighbor fusion
²³⁴ and decoder.

²³⁵ 3.8. Multi-Modal Extension

²³⁶ At urban intersections, vehicles face multiple plausible futures (e.g., turning left, going straight,
²³⁷ turning right). To capture this multi-modality, we extend the LSTM models with $K = 6$ prediction
²³⁸ heads for the 8-second horizon. Each mode has an independent MLP decoder producing a trajectory
²³⁹ with CV-residual decoding (Equation (5)). A shared confidence head maps the fused representation to
²⁴⁰ mode probabilities via softmax.

²⁴¹ We use WTA loss where only the mode closest to the ground truth receives gradient:

$$k^* = \arg \min_k \frac{1}{T} \sum_t \|\hat{\mathbf{y}}_{k,t} - \mathbf{y}_t^*\|_2, \quad \mathcal{L} = \text{SmoothL1}(\hat{\mathbf{Y}}_{k^*}, \mathbf{Y}^*) - \log c_{k^*}, \quad (13)$$

²⁴² where SmoothL1 denotes the Huber loss [40].

²⁴³ 3.9. Model Complexity

²⁴⁴ Table 1 summarizes the parameter counts for all model variants. Lane conditioning adds
²⁴⁵ fewer than 50,000 parameters (approximately 8%) to the LSTM backbone, primarily from the
²⁴⁶ lane MLP projection, message passing weights, and cross-attention parameters. The Transformer
²⁴⁷ lane-conditioned variant shows a larger overhead (+33.3%) because the lane encoder dimension
²⁴⁸ matches the Transformer’s wider $d_{\text{model}}=128$. All models remain under 700,000 parameters—orders
²⁴⁹ of magnitude smaller than state-of-the-art methods.

Table 1. Model parameter counts.

Model	Parameters	Overhead
LSTM Baseline (single)	582,562	Ref.
LSTM Lane-Cond. (single)	629,698	+8.1%
LSTM Lane-Cond. ($K=6$)	679,618	+16.7%*
Transformer Baseline	456,576	Ref.
Transformer Lane-Cond.	608,640	+33.3%

*Overhead relative to LSTM Baseline (single).

²⁵⁰ 4. Experimental Setup

²⁵¹ 4.1. Dataset and Preprocessing

²⁵² We use the Waymo Open Motion Dataset (WOMD) v1.1.0 [17], one of the largest public motion
²⁵³ forecasting benchmarks, containing over 100,000 driving scenarios across diverse urban environments.
²⁵⁴ Each scenario spans 9.1 s at 10 Hz (91 frames), providing both agent trajectories and HD map
²⁵⁵ information including lane centerlines, connectivity, traffic signals, and stop signs.

²⁵⁶ We select 89,258 scenarios containing traffic signal information (the “signal” subset, approximately
²⁵⁷ 87% of the full dataset), ensuring coverage of complex urban environments where lane topology is
²⁵⁸ most relevant for prediction. A 15% random split (seed 42) yields 75,869 training and 13,389 validation
²⁵⁹ scenarios. This custom split differs from the official WOMD train/val/test partition, so absolute
²⁶⁰ numbers should not be directly compared with leaderboard results. However, all internal comparisons
²⁶¹ (baseline vs. lane-conditioned) use identical splits, ensuring fair evaluation.

262 For 3-second prediction, each scenario yields up to six samples using anchor frames at indices
 263 $\{10, 20, 30, 40, 50, 60\}$, resulting in approximately 450,000 training samples. For 8-second prediction,
 264 only one anchor (index 10) is valid due to the 9.1 s scenario length, yielding approximately 75,000
 265 training samples. All trajectories are transformed to an ego-centric bird's-eye-view coordinate system
 266 aligned with the ego vehicle's heading. Data augmentation includes random 360° rotation of the entire
 267 scene (trajectories, neighbors, and lane features), which prevents the model from memorizing absolute
 268 orientations.

269 4.2. Training Details

270 All models are trained using AdamW [41] with gradient clipping at norm 1.0. LSTM models
 271 use learning rate 10^{-3} with weight decay 10^{-4} . Transformer models use 5×10^{-4} with weight decay
 272 10^{-2} and 5 epochs of linear warmup. All models use cosine annealing [42] over 100 epochs with early
 273 stopping (patience 20). Batch size is 128.

274 For 3-second experiments, we train with three random seeds (7, 42, 123) and report mean \pm
 275 standard deviation with paired t -test p -values. For 8-second experiments, we use seed 42.

276 All experiments are conducted on a single NVIDIA RTX 4090 GPU with 24 GB VRAM.

277 4.3. Evaluation Metrics

- 278 • **ADE**: mean L_2 distance over all future timesteps.
- 279 • **FDE**: L_2 distance at the last predicted timestep.
- 280 • **minADE / minFDE**: minimum ADE / FDE over K modes.
- 281 • **Miss Rate (MR@dm)**: fraction where best-mode endpoint error exceeds d meters.

282 For error decomposition, we project errors onto the heading direction (longitudinal) and
 283 perpendicular axis (lateral).

284 5. Results

285 5.1. 3-Second Prediction: Multi-Seed Validation

286 We first validate the lane conditioning effect at the shorter 3-second horizon using three random
 287 seeds to establish statistical significance. Table 2 presents the results.

Table 2. Single-modal prediction at 3 s (89K scenes, 100 epochs, 3 seeds).

Model	ADE@3 s (m)	Best Seed	p -value
LSTM Baseline	0.559 ± 0.007	0.552	—
LSTM Lane-Cond.	0.507 ± 0.011	0.496	0.0071
Improvement			+9.3%

288 Lane conditioning achieves a statistically significant 9.3% ADE reduction ($p = 0.0071$, paired
 289 t -test). The improvement is consistent across all three seeds: +9.1% (seed 42), +8.1% (seed 123),
 290 +10.9% (seed 7).

291 5.2. 8-Second Multi-Modal Results

292 The 8-second multi-modal setting ($K=6$) represents the primary benchmark comparison, as it
 293 matches the Waymo Motion Prediction Challenge evaluation protocol. Table 3 presents the results.

294 The improvement at 8 s is substantially larger than at 3 s, with minFDE improving by 33.7% and
 295 miss rate dropping by an absolute 14.5 percentage points. This confirms that lane structure becomes
 296 more valuable at longer horizons.

Table 3. Multi-modal prediction ($K=6$) at 8 s (seed 42, 100 epochs).

Model	minADE (m)	minFDE (m)	MR@5 m
LSTM Baseline	1.839	4.959	33.9%
LSTM Lane-Cond.	1.337	3.289	19.4%
Improvement	+27.3%	+33.7%	+42.7%

297 5.3. Architecture-Agnostic Benefit

298 Table 4 compares the effect of lane conditioning across LSTM and Transformer architectures at 8 s
299 (single-modal).

Table 4. Single-modal prediction at 8 s across architectures (seed 42, 100 epochs).

Model	ADE (m)	FDE (m)	ADE@3 s (m)	LC Improv.
LSTM Baseline	3.781	11.244	0.553	—
LSTM Lane-Cond.	3.075	8.688	0.516	+18.7%
TF Baseline	4.859	13.875	0.828	—
TF Lane-Cond.	3.303	8.956	0.663	+32.0%

300 Lane conditioning provides a consistent improvement for *both* architectures: +18.7% for LSTM
301 and +32.0% for Transformer. The LSTM outperforms the Transformer in absolute terms, which is
302 expected with only 11 input timesteps ($T_{\text{obs}} = 11$): the LSTM’s recurrent inductive bias for short
303 sequential processing likely outweighs the Transformer’s general-purpose attention at this limited
304 sequence length, consistent with findings that simple models can match or outperform Transformers
305 for short time series [20,28].

306 5.4. Horizon-Dependent Improvement

307 Table 5 summarizes the improvement from lane conditioning across all experimental settings.

Table 5. Lane conditioning improvement across horizons and settings.

Setting	Horizon	Metric	Improvement
LSTM, single, 3 seeds	3 s	ADE	+9.3% ($p=0.0071$)
LSTM, single	8 s	ADE	+18.7%
LSTM, $K=6$	8 s	minADE	+27.3%
LSTM, $K=6$	8 s	minFDE	+33.7%
LSTM, $K=6$	8 s	MR@5 m	+42.7%
Transformer, single	8 s	ADE	+32.0%

308 The improvement increases consistently from 3 s to 8 s and from ADE to FDE to miss rate, revealing
309 that lane conditioning most strongly benefits trajectory *endpoints*—consistent with the intuition that
310 lane structure constrains where vehicles can plausibly end up.

311 Figure 3 visualizes the error growth over the 8-second horizon, showing how the gap between
312 baseline and lane-conditioned models widens progressively. Figure 4 further decomposes the
313 improvement by horizon and error axis, confirming that both longitudinal and lateral components
314 benefit, with the improvement growing monotonically from 1 s to 7 s before a slight plateau at 8 s.

315 5.5. Error Decomposition

316 Table 6 decomposes errors into lateral and longitudinal components for the multi-modal LSTM
317 models at 8 s.

318 Both components improve substantially, with the strongest improvement in endpoint lateral error
319 (+30.5%). This is significant for intersection safety: lateral errors correspond to lane departures and

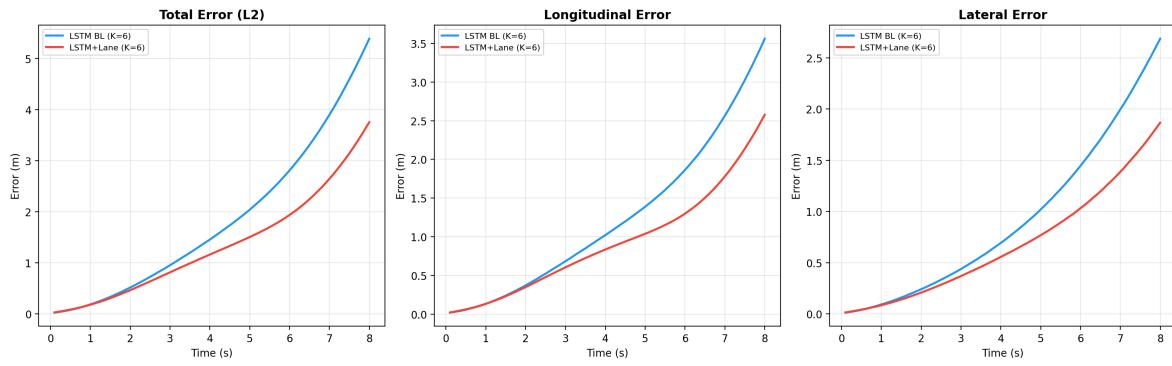


Figure 3. Error growth over prediction horizon for LSTM baseline vs. lane-conditioned model ($K=6$, 8 s). Left: total L2 error. Center: longitudinal error. Right: lateral error. The gap between models widens with horizon, confirming that lane conditioning becomes increasingly valuable for longer-term prediction.



Figure 4. Percentage improvement from lane conditioning decomposed by prediction horizon and error component ($K=6$). Both longitudinal (blue) and lateral (red) improvements grow monotonically from 1 s to 7 s, with lateral improvement slightly dominant at shorter horizons.

Table 6. Lateral / longitudinal error decomposition ($K=6$, 8 s).

Component	Baseline (m)	Lane-Cond. (m)	Improvement
Avg. Longitudinal	1.238	0.924	+25.4%
Avg. Lateral	0.919	0.675	+26.5%
Endpoint Longitudinal	3.561	2.577	+27.6%
Endpoint Lateral	2.687	1.867	+30.5%

³²⁰ potential conflicts with adjacent traffic. The balanced improvement across both axes indicates that lane
³²¹ conditioning provides a comprehensive geometric prior.

³²² Figure 5 provides a visual complement, showing the absolute longitudinal and lateral errors at
³²³ 2 s, 4 s, 6 s, and 8 s horizons for both models.

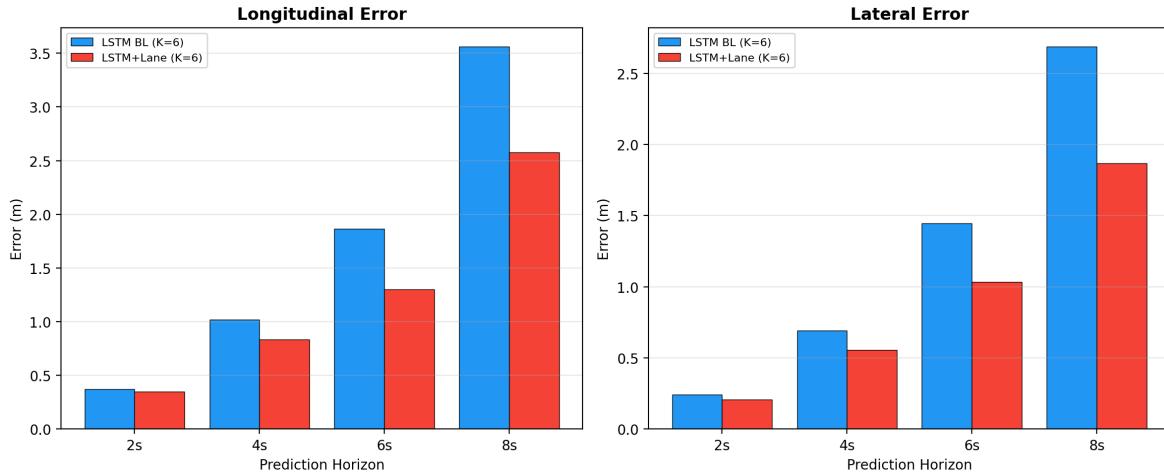


Figure 5. Absolute longitudinal (left) and lateral (right) errors at multiple horizons. The lane-conditioned model (red) consistently reduces both error components, with the gap growing at longer horizons.

³²⁴ 5.6. Qualitative Analysis

³²⁵ Figure 6 presents a representative side-by-side comparison of trajectory predictions from the
³²⁶ baseline and lane-conditioned LSTM models ($K=6$, 8 s prediction) at an intersection with a curved
³²⁷ lane. In the baseline panel (left), predicted modes diverge in multiple directions without regard for
³²⁸ road structure, resulting in a minADE of 5.68 m and minFDE of 12.79 m. In the lane-conditioned panel
³²⁹ (right), the same scene yields tightly clustered predictions that closely follow the curved lane structure,
³³⁰ achieving a minADE of 1.77 m and minFDE of 3.34 m—a 68.9% ADE improvement. The waterflow lane
³³¹ graph is visible in both panels: the ego lane (blue), successor lanes (green), and adjacent lanes (amber).
³³² The lane-conditioned model’s predictions align closely with these structural cues, demonstrating how
³³³ lane features constrain predictions to geometrically plausible trajectories even at complex intersections.

³³⁴ 5.7. Comparison with Waymo Baselines

³³⁵ Table 7 contextualizes our results within published Waymo benchmarks.

Table 7. Context with published Waymo benchmarks at 8 s (vehicle minADE, meters).

Method	minADE@8s	Input Features
Waymo LSTM [†]	2.63	agent state (pos, vel, bbox)
Our LSTM Baseline ($K=6$)	1.84	position + neighbors
Waymo LSTM + rg + ts + hi [†]	1.34	agent state + map + signals + interactions
Our LSTM-LC ($K=6$)	1.34	position + lane graph

[†]Waymo baselines from [17], Table 2 (vehicle class, standard val set).

rg = road graph, ts = traffic signals, hi = high-order interactions.

³³⁶ Our lane-conditioned model (1.34 m) matches the Waymo official LSTM using the *full feature set*
³³⁷ (agent state + road graph + traffic signals + high-order interactions), while using only 2D position
³³⁸ inputs plus local lane features. This demonstrates that **local lane structure can effectively substitute**
³³⁹ **for hand-engineered kinematic features**. We note that the Waymo baselines are evaluated on the

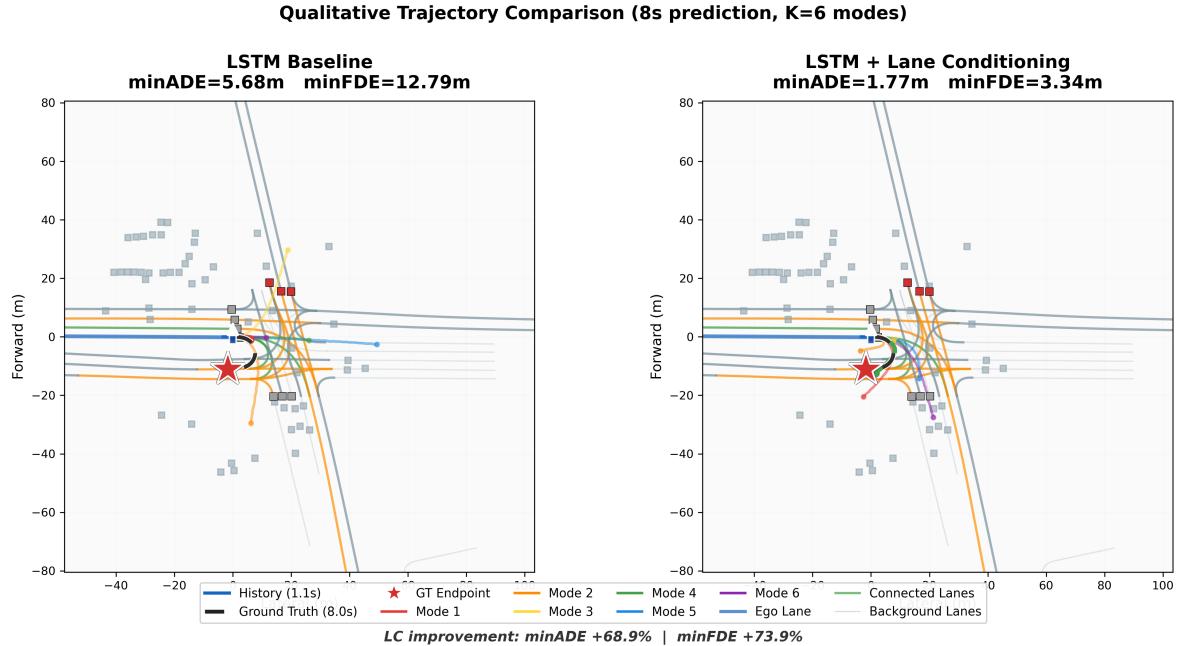


Figure 6. Qualitative trajectory comparison: LSTM Baseline (left) vs. LSTM + Lane Conditioning (right) at an intersection ($K=6$ modes, 8 s prediction). Colored lines show predicted modes; black dashed line shows ground truth; red star marks the ground-truth endpoint. The lane-conditioned model produces tightly clustered predictions along the curved lane, while baseline modes scatter widely across the scene.

340 official validation set, while our results use a custom 85/15 split of the signal-controlled subset; this
 341 limits direct numerical comparison but supports the qualitative conclusion that lane structure provides
 342 comparable information to rich input features. Integrating lane conditioning into state-of-the-art
 343 architectures such as MTR [15] or MTR++ [43], which achieve substantially lower errors through rich
 344 features, dense map encodings, and iterative refinement, is a promising direction for future work.

345 6. Discussion

346 6.1. Why Lane Conditioning Helps More at Longer Horizons

347 The improvement increases from +9.3% at 3 s to +27.3% at 8 s. This can be explained by the
 348 interplay of two information sources:

- 349 1. **Kinematic signal:** The vehicle’s history provides strong short-term predictions through velocity
 350 and acceleration. This signal decays rapidly—by 8 s, it is essentially uninformative about lane
 351 occupancy.
- 352 2. **Structural signal:** Lane geometry constrains physically plausible future positions. This constraint
 353 is invariant to prediction horizon.

354 As kinematic information degrades, lane structure becomes the dominant useful signal, explaining
 355 the increasing benefit at longer horizons.

356 6.2. Architecture-Agnostic Nature

357 The consistent improvement across LSTM (+18.7%) and Transformer (+32.0%) provides
 358 strong evidence that lane conditioning is not architecture-specific. The LSTM uses single-vector
 359 cross-attention, while the Transformer uses full-sequence multi-head cross-attention. Both yield
 360 substantial improvements, suggesting that even simple lane fusion mechanisms are effective.

³⁶¹ The Transformer shows a larger relative improvement (+32% vs. +19%), possibly because
³⁶² full-sequence cross-attention enables richer lane-trajectory correspondences—associating early
³⁶³ positions with upstream lanes and later positions with downstream lanes. Additionally, as discussed in
³⁶⁴ Section 6.6, lane conditioning provides an implicit regularization effect that is particularly pronounced
³⁶⁵ for the Transformer, further contributing to the larger improvement.

³⁶⁶ 6.3. On the Gap to State-of-the-Art Methods

³⁶⁷ We emphasize that **the goal of this study is not to achieve state-of-the-art absolute performance,**
³⁶⁸ **but to validate a general design principle.** Specifically, we demonstrate that local lane graph
³⁶⁹ conditioning provides a consistent 19–32% improvement *regardless of the backbone architecture*. This
³⁷⁰ finding has direct practical implications: any existing or future trajectory prediction system can
³⁷¹ potentially benefit from adding a lightweight lane conditioning module.

³⁷² To contextualize the gap to state-of-the-art: methods such as MTR [15] and MTR++ [43] employ
³⁷³ (i) rich input features (velocity, acceleration, heading, bounding box, traffic light states), (ii) dense global
³⁷⁴ map encodings with hundreds of polylines, (iii) multi-scale attention with millions of parameters,
³⁷⁵ and (iv) iterative refinement decoding. Our models use only 2D position with fewer than 700,000
³⁷⁶ parameters. The fact that our lane-conditioned model matches the Waymo official LSTM baseline [17]
³⁷⁷ that uses the *full feature set* ($\text{minADE} = 1.34 \text{ m}$)—despite using only position inputs plus local lane
³⁷⁸ features—demonstrates that local lane structure can substitute for hand-engineered kinematic features.
³⁷⁹ This substitution property is arguably more valuable as a design principle than incremental gains on a
³⁸⁰ specific leaderboard, because it generalizes across architectures, horizons, and deployment constraints.

³⁸¹ 6.4. Implications for Safety and Sustainable Urban Mobility

³⁸² The World Health Organization reports that road traffic injuries remain a leading cause of death
³⁸³ globally, with intersection-related crashes constituting a disproportionate share [44]. The miss rate
³⁸⁴ reduction from 33.9% to 19.4% at the 5 m threshold is particularly significant for intersection safety.
³⁸⁵ A 33.9% miss rate means one in three trajectory predictions ends more than 5 m (approximately
³⁸⁶ two lane widths) from the true endpoint—a potentially dangerous error for planning and collision
³⁸⁷ avoidance systems. Reducing this to 19.4% substantially improves the reliability of downstream
³⁸⁸ safety applications, directly contributing to UN Sustainable Development Goal 3 (Good Health and
³⁸⁹ Well-Being) through reduced road fatalities, and to the goal of reducing the 40% of crashes that occur
³⁹⁰ at intersections [1].

³⁹¹ From a sustainability perspective, improved trajectory prediction is expected to have cascading
³⁹² benefits for urban mobility, supporting UN SDG 11 (Sustainable Cities and Communities). Taiebat
³⁹³ et al. [3] identify prediction-enabled cooperative driving as one of the key mechanisms through
³⁹⁴ which connected and automated vehicles can reduce energy consumption and emissions. Barth and
³⁹⁵ Boriboonsomsin [2] estimate that congestion-related stop-and-go driving can increase CO₂ emissions
³⁹⁶ by up to 40% compared to free-flow conditions. While we do not directly measure downstream
³⁹⁷ planning or emission outcomes, a 42.7% reduction in miss rate at the 5 m threshold is expected to
³⁹⁸ reduce false-positive emergency braking events and improve the reliability of cooperative maneuver
³⁹⁹ planning, indirectly contributing to smoother traffic flow and lower fuel consumption. These benefits
⁴⁰⁰ remain indirect: a complete sustainability assessment would require integration with downstream
⁴⁰¹ planning modules and real-world or simulation-based driving experiments, which we leave for future
⁴⁰² work.

⁴⁰³ Moreover, the lightweight nature of our lane conditioning module (<700,000 total parameters)
⁴⁰⁴ is amenable to deployment on resource-constrained edge devices rather than power-hungry cloud
⁴⁰⁵ infrastructure. For fleet-scale deployment of autonomous vehicles, the per-vehicle computational cost
⁴⁰⁶ becomes a significant factor in the total environmental footprint. A model that achieves competitive
⁴⁰⁷ accuracy with orders-of-magnitude fewer parameters than SOTA methods (which employ millions of

parameters) aligns with UN SDG 9 (Industry, Innovation and Infrastructure) by facilitating affordable, energy-efficient prediction systems suitable for urban environments.

6.5. Computational Efficiency and Green AI

Beyond prediction accuracy, the computational footprint of trajectory prediction models has direct sustainability implications aligned with the principles of Green AI [45]. State-of-the-art methods such as MTR++ [43] employ architectures with \sim 15 million parameters, requiring substantial computational resources for both training and inference. For electric autonomous vehicles (EVs), the energy consumed by onboard computation directly reduces driving range—a critical concern for sustainable deployment at scale.

Our approach offers a favorable accuracy-efficiency trade-off. The lane conditioning module adds fewer than 50,000 parameters (approximately 8% overhead for LSTM) while providing 19–32% accuracy improvement. The total model size ($<700,000$ parameters, approximately $20\times$ smaller than SOTA) enables real-time inference on resource-constrained edge devices without dedicated GPU hardware. In our experiments, all models were trained on a single consumer-grade GPU (NVIDIA RTX 4090), with each 100-epoch training run completing in approximately 8–12 GPU-hours—a fraction of the computational budget required for SOTA systems that train on multi-GPU clusters. This makes the approach particularly suitable for large-scale fleet deployment, where per-vehicle computational cost is a significant factor in the total environmental footprint of autonomous driving operations.

6.6. Training Dynamics and Implicit Regularization

An important practical finding is that lane-conditioned models require longer training. Baseline models converge within 30–50 epochs, while lane-conditioned models continue improving until 80–100 epochs. Short training runs can be misleading, showing no benefit or even degradation from lane conditioning.

This effect is most striking for the Transformer architecture. The Transformer baseline reaches its best validation ADE at epoch 29 (4.859 m) but then *overfits progressively*, degrading to 6.154 m by epoch 99—a 26.6% increase in error. In contrast, the lane-conditioned Transformer improves monotonically throughout training, reaching its best ADE at epoch 94 (3.282 m) with no sign of overfitting. This suggests that lane conditioning acts as an *implicit regularizer* [46]: the structural prior from lane features constrains the model’s hypothesis space—an instance of the relational inductive biases described by Battaglia et al. [9]—preventing the Transformer’s flexible attention mechanism from memorizing training-set artifacts. This regularization benefit is practically significant—it reduces the need for extensive hyperparameter tuning (e.g., dropout, weight decay) and makes training more robust, which in turn reduces wasted computation from failed experiments, aligning with Green AI principles [45].

6.7. Limitations

This study has several limitations. First, the 8-second experiments use a single random seed; while the 3-second results establish statistical significance across three seeds ($p = 0.0071$), additional seeds for the 8-second setting would strengthen the conclusions. Second, our models use only 2D position inputs; incorporating velocity, heading, and bounding box features would likely improve absolute performance, though this would complicate the isolation of the lane conditioning effect. Third, the waterflow lane graph is limited to 16 lanes within a 3-hop neighborhood, which may be insufficient for very large or complex intersection topologies. Fourth, evaluation is limited to the ego vehicle; joint multi-agent prediction would provide a more complete assessment of intersection safety. Finally, while our lane conditioning module achieves substantial relative improvements, the remaining gap to state-of-the-art methods indicates that lane conditioning alone does not substitute for the full suite of innovations (rich input features, dense global map encoding, iterative refinement) employed in methods like MTR++.

455 7. Conclusions

456 This paper presented a controlled study of local lane graph conditioning across two encoder
457 architectures, two prediction horizons, and both single-modal and multi-modal settings on the Waymo
458 Open Motion Dataset. The key findings are:

- 459 1. **Consistent, architecture-agnostic improvement.** Lane conditioning improves both LSTM (+9.3%
460 to +27.3%) and Transformer (+32.0%) backbones.
- 461 2. **Horizon-dependent benefit.** Improvement increases from +9.3% at 3 s to +27.3% at 8 s,
462 confirming that lane structure becomes more valuable as kinematic extrapolation degrades.
- 463 3. **Balanced error reduction.** Both lateral (+26.5%) and longitudinal (+25.4%) errors improve, with
464 endpoint lateral error showing the strongest reduction (+30.5%).
- 465 4. **Feature substitution.** Our lane-conditioned LSTM with $K=6$ modes achieves minADE = 1.34 m,
466 matching the Waymo official LSTM using the full feature set.

467 These results establish local lane graph conditioning as a lightweight, general-purpose module
468 for trajectory prediction. Future work will explore: (1) integration of the lane conditioning module
469 into state-of-the-art architectures such as MTR to assess whether improvements transfer at higher
470 performance levels; (2) joint multi-agent prediction leveraging shared lane graph representations;
471 (3) cross-dataset generalization to Argoverse [47] and nuScenes [48]; and (4) quantifying the
472 downstream impact of improved prediction accuracy on autonomous vehicle energy efficiency and
473 safety outcomes.

474 **Author Contributions:** Conceptualization, X.Z. and C.A.; methodology, X.Z.; software, X.Z.; validation, X.Z.;
475 formal analysis, X.Z.; investigation, X.Z.; resources, C.A.; data curation, X.Z.; writing—original draft preparation,
476 X.Z.; writing—review and editing, X.Z. and C.A.; visualization, X.Z.; supervision, C.A.; project administration,
477 C.A. All authors have read and agreed to the published version of the manuscript.

478 **Funding:** This research received no external funding.

479 **Data Availability Statement:** The trajectory prediction models and training code developed in this
480 study are available from the corresponding author upon reasonable request. The Waymo Open Motion
481 Dataset used for training and evaluation is publicly available at <https://waymo.com/open/data/motion/> under the Waymo Dataset License Agreement.

483 **Informed Consent Statement:** Not applicable.

484 **Acknowledgments:** The authors acknowledge the use of the Waymo Open Motion Dataset for the experiments
485 presented in this work. Computational resources were provided by Concordia University.

486 **Conflicts of Interest:** The authors declare no conflicts of interest.

487 Abbreviations

488 The following abbreviations are used in this manuscript:

ADE	Average Displacement Error
BEV	Bird's-Eye View
BFS	Breadth-First Search
CV	Constant Velocity
EV	Electric Vehicle
FDE	Final Displacement Error
HD	High Definition
490 LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MIR	Miss Rate
SDG	Sustainable Development Goal
TF	Transformer
WOMD	Waymo Open Motion Dataset
WTA	Winner-Takes-All

491 References

- 492 1. Choi, E.H. Crash Factors in Intersection-Related Crashes: An On-Scene Perspective. *National Highway*
493 *Traffic Safety Administration (NHTSA)* **2010**, *811*, 366.
- 494 2. Barth, M.; Boriboonsomsin, K. Traffic Congestion and Greenhouse Gases. *ACCESS Magazine* **2009**, *1*, 2–9.
- 495 3. Taiebat, M.; Brown, A.L.; Safford, H.R.; Qu, S.; Xu, M. A Review on Energy, Environmental, and
496 Sustainability Implications of Connected and Automated Vehicles. *Environmental Science & Technology*
497 **2018**, *52*, 11449–11465.
- 498 4. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibáñez, J.A. Internet of Vehicles: Architecture, Protocols, and
499 Security. *IEEE Internet of Things Journal* **2018**, *5*, 3701–3709.
- 500 5. Guerrero-Ibáñez, J.; Zeadally, S.; Contreras-Castillo, J. Sensor Technologies for Intelligent Transportation
501 Systems. *Sensors* **2018**, *18*, 1212.
- 502 6. Huang, Y.; Du, J.; Yang, Z.; Zhou, Z.; Zhang, L.; Chen, H. A Survey on Trajectory-Prediction Methods for
503 Autonomous Driving. *IEEE Transactions on Intelligent Vehicles* **2022**, *7*, 652–674.
- 504 7. Mozaffari, S.; Al-Jarrah, O.Y.; Dianati, M.; Jennings, P.; Mouzakitis, A. Deep Learning-Based Vehicle
505 Behaviour Prediction for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* **2020**,
506 *23*, 33–47.
- 507 8. Wang, C.; Xu, C.; Xia, J.; Qian, Z.; Lu, L. A Review of Surrogate Safety Measures and Their Applications in
508 Connected and Automated Vehicles Safety Modeling. *Accident Analysis & Prevention* **2021**, *157*, 106157.
509 Published online 2021.
- 510 9. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti,
511 A.; Raposo, D.; Santoro, A.; Faulkner, R.; others. Relational Inductive Biases, Deep Learning, and Graph
512 Networks. *arXiv preprint arXiv:1806.01261* **2018**.
- 513 10. Gao, J.; Sun, C.; Zhao, H.; Shen, Y.; Anguelov, D.; Li, C.; Schmid, C. VectorNet: Encoding HD Maps and
514 Agent Dynamics from Vectorized Representation. Proceedings of the IEEE/CVF Conference on Computer
515 Vision and Pattern Recognition (CVPR), 2020, pp. 11525–11533.
- 516 11. Liang, M.; Yang, B.; Hu, R.; Chen, Y.; Liao, R.; Feng, S.; Urtasun, R. Learning Lane Graph Representations
517 for Motion Forecasting. European Conference on Computer Vision (ECCV). Springer, 2020, pp. 541–556.
- 518 12. Zhao, H.; Gao, J.; Lan, T.; Sun, C.; Sapp, B.; Varadarajan, B.; Shen, Y.; Shen, Y.; Chai, Y.; Schmid, C.; others.
519 TNT: Target-driven Trajectory Prediction. Conference on Robot Learning (CoRL). PMLR, 2021, pp. 895–904.
- 520 13. Zhou, Z.; Ye, L.; Wang, J.; Wu, K.; Lu, K. HiVT: Hierarchical Vector Transformer for Multi-Agent Motion
521 Prediction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
522 (CVPR), 2022, pp. 8823–8833.
- 523 14. Deo, N.; Wolff, E.; Beijbom, O. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals.
524 Conference on Robot Learning (CoRL). PMLR, 2022, pp. 203–212.
- 525 15. Shi, S.; Jiang, L.; Dai, D.; Schiele, B. Motion Transformer with Global Intention Localization and Local
526 Movement Refinement. Advances in Neural Information Processing Systems (NeurIPS), 2022, Vol. 35, pp.
527 6531–6543.
- 528 16. Zhou, Z.; Wang, J.; Li, Y.H.; Huang, Y.K. Query-Centric Trajectory Prediction. Proceedings of the IEEE/CVF
529 Conference on Computer Vision and Pattern Recognition (CVPR), 2023.
- 530 17. Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C.R.; Zhou, Y.;
531 Yang, Z.; Chou, A.; Sun, P.; Ngiam, J.; Vasudevan, V.; McCauley, A.; Shlens, J.; Anguelov, D. Large Scale
532 Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. Proceedings
533 of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9710–9719.
- 534 18. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
- 535 19. Alahi, A.; Goel, K.; Raber, V.; Sadeghian, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory
536 Prediction in Crowded Spaces. Proceedings of the IEEE Conference on Computer Vision and Pattern
537 Recognition (CVPR), 2016, pp. 961–971.
- 538 20. Park, S.H.; Kim, B.; Kang, C.M.; Chung, C.C.; Choi, J.W. Sequence-to-Sequence Prediction of Vehicle
539 Trajectory via LSTM Encoder-Decoder Architecture. 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE,
540 2018, pp. 1672–1678.
- 541 21. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. Proceedings of the
542 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 1468–1476.

- 543 22. Chandra, R.; Guan, T.; Panber, S.; Manocha, D. Forecasting Trajectory and Behavior of Road-Agents Using
544 Spectral Clustering in Graph-LSTMs. *IEEE Robotics and Automation Letters* **2020**, *5*, 4882–4890.
- 545 23. Li, J.; Yang, F.; Tomizuka, M.; Choi, C. EvolveGraph: Multi-Agent Trajectory Prediction with Dynamic
546 Relational Reasoning. Advances in Neural Information Processing Systems (NeurIPS), 2020, Vol. 33, pp.
547 19783–19794.
- 548 24. Mo, X.; Huang, Z.; Xing, Y.; Lv, C. Graph and Recurrent Neural Network-Based Vehicle Trajectory
549 Prediction for Highway Driving. *IEEE Transactions on Intelligent Transportation Systems* **2022**,
550 *23*, 17534–17547.
- 551 25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I.
552 Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS), 2017, Vol. 30.
- 553 26. Ngiam, J.; Caine, B.; Vasudevan, V.; Zhang, Z.; Chiu, H.L.; Pierce, A.; Truong, Y.; Dao, T.D.; Sapp, B.; Qi, C.;
554 others. Scene Transformer: A Unified Architecture for Predicting Multiple Agent Trajectories. International
555 Conference on Learning Representations (ICLR), 2022.
- 556 27. Nayakanti, N.; Al-Rfou, R.; Zhou, A.; Goel, K.; Refaat, K.S.; Sapp, B. Wayformer: Motion Forecasting via
557 Simple & Efficient Attention Networks. 2023 IEEE International Conference on Robotics and Automation
558 (ICRA). IEEE, 2023, pp. 2187–2193.
- 559 28. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? Proceedings
560 of the AAAI Conference on Artificial Intelligence, 2023, Vol. 37, pp. 11121–11128.
- 561 29. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. International
562 Conference on Learning Representations (ICLR), 2017.
- 563 30. Zeng, W.; Liang, M.; Liao, R.; Urtasun, R. LaneRCNN: Distributed Representations for Graph-Centric
564 Motion Forecasting. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and
565 Systems (IROS). IEEE, 2021, pp. 532–539.
- 566 31. Gu, J.; Sun, C.; Zhao, H. DenseTNT: End-to-End Trajectory Prediction from Dense Goal Sets. Proceedings
567 of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 15303–15312.
- 568 32. Kim, B.; Park, S.H.; Lee, S.; Khoshimjonov, E.; Kum, D.; Kim, J.; Kim, J.S.; Choi, J.W. LaPred: Lane-Aware
569 Prediction of Multi-Modal Future Trajectories of Dynamic Agents. Proceedings of the IEEE/CVF
570 Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14636–14645.
- 571 33. Wang, M.; Zhu, X.; Yu, C.; Li, W.; Ma, Y.; Jin, R.; Ren, X.; Li, D.; Yin, M.; Wang, W. GANet: Goal Area
572 Network for Motion Forecasting. Proceedings of the IEEE International Conference on Robotics and
573 Automation (ICRA). IEEE, 2023, pp. 10784–10790.
- 574 34. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum
575 Chemistry. International Conference on Machine Learning (ICML). PMLR, 2017, pp. 1263–1272.
- 576 35. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate.
577 *arXiv preprint arXiv:1409.0473* **2015**.
- 578 36. Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; Alahi, A. Social GAN: Socially Acceptable Trajectories with
579 Generative Adversarial Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and
580 Pattern Recognition (CVPR), 2018, pp. 2255–2264.
- 581 37. Lee, S.; Purushwalkam, S.; Cogswell, M.; Crandall, D.; Batra, D. Stochastic Multiple Choice Learning for
582 Training Diverse Deep Ensembles. Advances in Neural Information Processing Systems (NeurIPS), 2016,
583 Vol. 29.
- 584 38. Salzmann, T.; Ivanovic, B.; Chakravarty, P.; Pavone, M. Trajectron++: Dynamically-Feasible Trajectory
585 Forecasting with Heterogeneous Data. European Conference on Computer Vision (ECCV). Springer, 2020,
586 pp. 683–700.
- 587 39. Schöller, C.; Aravantinos, V.; Lay, F.; Knoll, A. What the Constant Velocity Model Can Teach Us About
588 Pedestrian Motion Prediction. *IEEE Robotics and Automation Letters* **2020**, *5*, 1696–1703.
- 589 40. Huber, P.J. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* **1964**, *35*, 73–101.
- 590 41. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. International Conference on Learning
591 Representations (ICLR), 2019.
- 592 42. Loshchilov, I.; Hutter, F. SGDR: Stochastic Gradient Descent with Warm Restarts. International Conference
593 on Learning Representations (ICLR), 2017.

- 594 43. Shi, S.; Jiang, L.; Dai, D.; Schiele, B. MTR++: Multi-Agent Motion Prediction with Symmetric Scene
595 Modeling and Pair-Wise Context. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2024**,
596 46, 3543–3558.
- 597 44. World Health Organization. Global Status Report on Road Safety 2023. *World Health Organization* **2023**.
- 598 45. Schwartz, R.; Dodge, J.; Smith, N.A.; Etzioni, O. Green AI. *Communications of the ACM* **2020**, 63, 54–63.
- 599 46. Neyshabur, B.; Tomioka, R.; Srebro, N. In Search of the Real Inductive Bias: On the Role of Implicit
600 Regularization in Deep Learning. ICLR 2015 Workshop Track, 2015.
- 601 47. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan,
602 D.; others. Argoverse: 3D Tracking and Forecasting with Rich Maps. Proceedings of the IEEE/CVF
603 Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8748–8757.
- 604 48. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liang, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom,
605 O. nuScenes: A Multimodal Dataset for Autonomous Driving. Proceedings of the IEEE/CVF Conference
606 on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11621–11631.

607 © 2026 by the authors. Submitted to *Sustainability* for possible open access publication
608 under the terms and conditions of the Creative Commons Attribution (CC BY) license
609 (<http://creativecommons.org/licenses/by/4.0/>).