$SE(2)$-Aware Conditional Distribution Transport for Vehicle Trajectory Generation

Di Wen[1,2,4] **Zhaocheng He**[1,2,*] **Shuhui Wang**[3,*]
[1]Shenzhen Campus of Sun Yat-sen University [4]The Hong Kong Polytechnic University
[2]Guangdong Provincial Key Laboratory of Intelligent Transportation Systems
[3]State Key Lab of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences

Abstract

Vehicle trajectory generation is a fundamental task in autonomous driving, with applications in traffic simulation, safety evaluation and decision making. Existing data-driven methods achieve accurate trajectory generation yet provide limited consideration of heterogeneous characteristics across different behaviors and the inherent physical properties of vehicles, such that metric-level fidelity does not necessarily ensure physically plausible generation at the scene level. To address these limitations, we propose SEA, a novel **SE**(2)-**A**ware generation framework that models vehicle trajectories in complementary data and physical spaces. In the **data space**, SEA learns context conditioned trajectory distributions by incorporating traffic context and inter-vehicle interactions into continuous distribution evolution, enabling heterogeneous trajectory patterns to be characterized within a unified generative process. In the **physical space**, motivated by the **optimal transport** view of distribution interpolation, SEA exploits the planar rigid body structure of vehicles and realizes the generative evolution through geometry consistent geodesic interpolation on $SE(2)$, preserving position-orientation coherence throughout generation. Extensive experiments on large-scale real-world driving datasets demonstrate that SEA generates accurate and diverse trajectories while exhibiting physical characteristics consistent with real-world vehicle trajectories.

1 Introduction

Intelligent Transportation Systems (ITS) enhance travel efficiency, reduce traffic incidents and alleviate congestion [1]. Vehicle trajectories provide critical data for ITS. However, due to privacy

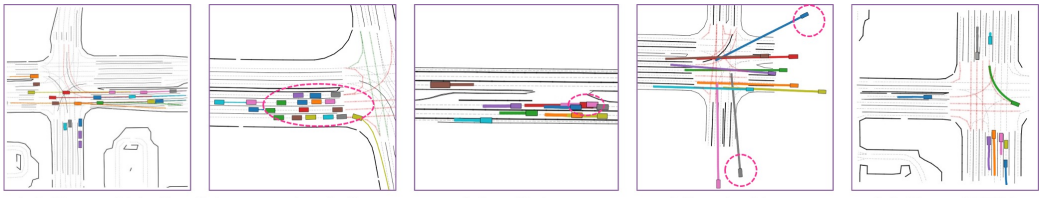


Figure 1: Comparison of generation results. In the training dataset (a), trajectories are mostly straight or stationary. As a result, the baseline methods (b)-(d) [2, 3, 4] sometimes generate physically implausible trajectories such as leaving lane boundaries, colliding with other vehicles or remaining stationary. In contrast, our method SEA (e) generates more physically plausible trajectories.

*✉ Joint corresponding authors: hezhch@mail.sysu.edu.cn and wangshuhui@ict.ac.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

concerns and the high cost of data collection [5], obtaining large scale, realistic trajectory datasets is difficult. Vehicle trajectory generation therefore serves as a core capability for addressing this need. While simulation tools [6, 7] are widely used for developing and validating trajectory generation methods, deep generative models offer an alternative by synthesizing multi-vehicle trajectories jointly rather than vehicle by vehicle. For example, TrafficGen [2] introduces a data driven autoregressive generative model. LCTGen [3] and InteractTraj [4] incorporate language as an additional supervision signal for generation, using Transformer based decoders and explicit vehicle to vehicle interaction.

However, existing studies rely on data driven strategies and ignore the physical laws that govern vehicle motion during operation. This limitation weakens model generalization beyond observed data and increases vulnerability in complex scenarios. For example, straight driving and stationary states are common vehicle motions in daily life, but rare yet critical motions such as left and right turns are also needed. As illustrated in Figure 1, the generated trajectories are unsafe and unrealistic. One possible reason is that data driven generators may overemphasize dominant motion patterns in imbalanced datasets, while lacking geometric priors to regularize the generated pose evolution. As a result, they can produce low quality trajectories in rare or interaction intensive scenarios.

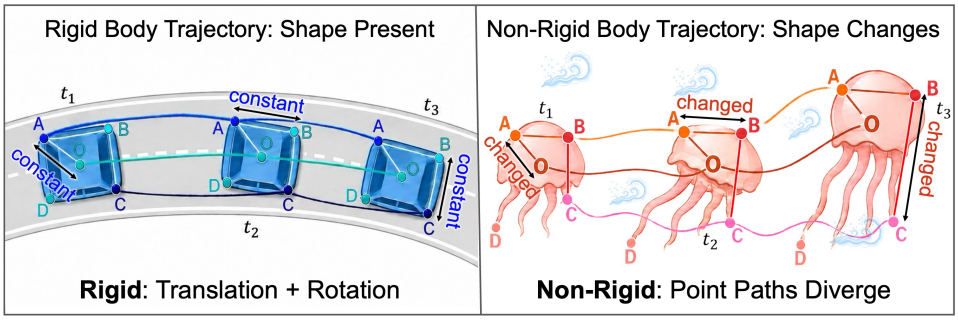


Figure 2: Difference between rigid body and non-rigid trajectories. A rigid body (left, e.g., a vehicle) preserves its shape and relative pointwise distances over time t_*, whereas a non-rigid body (right, e.g., a jellyfish) exhibits time varying distances due to deformation.

In this paper, we model vehicle trajectories through per step poses on $SE(2)$, the matrix Lie group of planar rigid body transformations. Figure 2 illustrates the rigid body assumption at the pose level: although a vehicle moves along a trajectory, its body fixed coordinate frame and relative pointwise geometry remain unchanged. Therefore, each time step vehicle state can be modeled as a planar rigid body pose in $SE(2)$. This design is motivated by the optimal transport principle that interpolation in distribution space should be compatible with the metric geometry of the underlying state space. Rather than directly "solving" the long tail distribution problem, we use physical laws as priors to guide generation, thereby ensuring that rare motions remain governed by the same dynamics learned from the dataset. Our contributions are summarized as follows:

- We propose SEA, a geometry-aware framework for vehicle trajectory generation that realizes distribution transport according to the underlying state space geometry of rigid body motion.
- We couple context conditioned distribution learning in the data space with $SE(2)$-consistent interpolation in the physical space, enabling the generative process to capture heterogeneous traffic distributions while preserving vehicle pose geometry.
- We show on Waymo and nuPlan that this geometry aware realization improves realism, diversity and physical consistency, especially for rare but safety critical cases.

2 Related Work

Mobility (vehicle) trajectory data generation attracts sustained attention, and prior surveys [8, 9] categorize existing theories and tools into two main paradigms: (1) knowledge/model driven and (2) learning/data driven. Consequent studies consistently adopt the same terminology system that separates trajectory generation methods along this axis.

Knowledge/Model Driven Trajectory Generation. Knowledge/model driven methods typically instantiate explicit priors from traffic theory [10, 11, 12], domain expertise [13, 14, 15] and simulator mechanisms [16, 17, 18] to produce synthetic trajectories. This paradigm is particularly motivated by the difficulty of collecting real world trajectory data and the associated privacy risks. In this setting, real data primarily support statistical estimation (e.g., calibrating model parameters) rather than feature learning. Simulation tools generate large scale long-term datasets, albeit with reliance on supplementary information and traffic model assumptions.

Data/Learning-Driven Trajectory Generation. On the one hand, extensions of deep generative models yield higher-fidelity microscopic trajectory generation [19]. For example, WcDT [20] encodes agent motion with a denoising diffusion probabilistic model and fuses latent features with a Transformer to capture scene-level interactions among agents. GoalFlow [21] employs flow matching and incorporates a refined scoring mechanism to generate multimodal trajectories. On the other hand, large language models also support vehicle trajectory generation by translating semantic commands into dynamical constraints and enabling new frameworks for trajectory synthesis [22, 23, 24].

Research Gap. Both paradigms exhibit distinct strengths and limitations. Knowledge/model-driven methods incorporate explicit domain knowledge and traffic theoretic priors, but their predefined assumptions may limit their ability to capture the complex dynamics of real-world traffic. In contrast, data/learning-driven methods provide greater flexibility in modeling complex trajectory distributions but typically require large-scale real-world data for training and offer limited interpretability.

3 Preliminary

3.1 Flow Matching

Yaron Lipman et al. [25] propose Flow Matching (FM), a framework that learns a time-dependent vector field v_t along continuous paths from a simple base distribution p_0 to a target distribution p_{tar}. Specifically, FM defines a flow map $\varphi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ as the solution to the ordinary differential equation $\frac{d}{dt}\varphi_t(x) = v_t[\varphi_t(x)]$ with initial condition $\varphi_0(x) = x$. Here $v_t = v_t(x;\theta)$ is represented by a neural network, where $\theta \in \mathbb{R}^p$ denotes the set of learnable parameters [26]. The induced density path p_t satisfies the pushforward relation[2] $p_t = [\varphi_t]_* p_0$. The objective of FM is defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_{\text{tar}}(x)} \| v_t(x;\theta) - u_{\text{tar}}(x) \|_2^2, \tag{1}$$

where u_{tar} denotes the target vector field associated with p_{tar}. Conditional Flow Matching (CFM [25]) yields gradients identical to those of FM in Equation (1), but replaces its path with a conditional probability $p_t(x|x_{\text{tar}})$. Rectified Flow Matching (RFM [27, 28]) straightens the transport path by using linear interpolation between x_0 and x_{tar}, so the target vector field aligns with the constant displacement $x_{\text{tar}} - x_0$ along this rectified path.

3.2 $SE(2)$ and Lie Group

Rigid body motion in the physical space $\mathbb{R}^n$ is described by the special Euclidean group $SE(n)$, which consists of orientation preserving isometries obtained by composing a rotation in $SO(n)$ with a translation in $\mathbb{R}^n$ [29, 30]. Under the standard bird's-eye-view (BEV) representation, each vehicle state is modeled as a planar pose whose configuration space is $SE(2)$, defined as follows [31]:

Definition 3.1 *The special Euclidean group $SE(2)$ is the set of all 3×3 real matrices of the form*

$$SE(2) = \left\{ \begin{bmatrix} R & U \\ 0 & 1 \end{bmatrix} \middle| \, R \in SO(2),\ U \in \mathbb{R}^2 \right\}, \tag{2}$$

where 0 is a row vector of two zeros, R and U denote planar rotation and translation respectively.

[2]The push-forward operator $*$ is defined as follows with p_0 is pure noise e.g. the standard normal distribution.

$$[\phi_t]_* p_0 = p_0 \left[\phi_t^{-1}(x)\right] \det \left[\frac{\partial \phi_t^{-1}}{\partial x}(x)\right].$$

This set is closed under matrix multiplication and inversion and thus forms a matrix Lie group G [32]. To characterize geometry-consistent interpolation on G, we use the exponential and logarithm maps on $SE(2)$. We consider a left-invariant Riemannian metric[3] g [33]. A smooth curve $\gamma : I \to G$ is a Riemannian geodesic if $D_t\dot{\gamma}(t) = 0$ [34]. We define its length L_γ and adopt the following

$$L_\gamma = \int_0^1 \|\gamma'^{(t)}\|_{\gamma(t)}\, dt, \quad \gamma\,(t) = g_0 \exp\left[t \log(g_0^{-1} g_1)\right]. \tag{3}$$

Since $SE(2)$ admits no bi-invariant Riemannian metric[4] [35], we adopt γ as a geometry-consistent interpolation on G, not necessarily a Riemannian geodesic under general left-invariant metrics.

4 Methodology

4.1 Overview

In this work, we present SEA, an $SE(2)$-aware framework for vehicle trajectory generation that incorporates planar rigid body pose geometry, as illustrated in Figure 3. Section 4.2 formulates the generation task, while Section 4.3 introduces traffic context encoding. Section 4.4 models context conditioned distribution transport in the data space and Section 4.5 realizes the transport through $SE(2)$-consistent interpolation between endpoint vehicle poses in the physical space. Finally, Section 4.6 presents the training objectives.

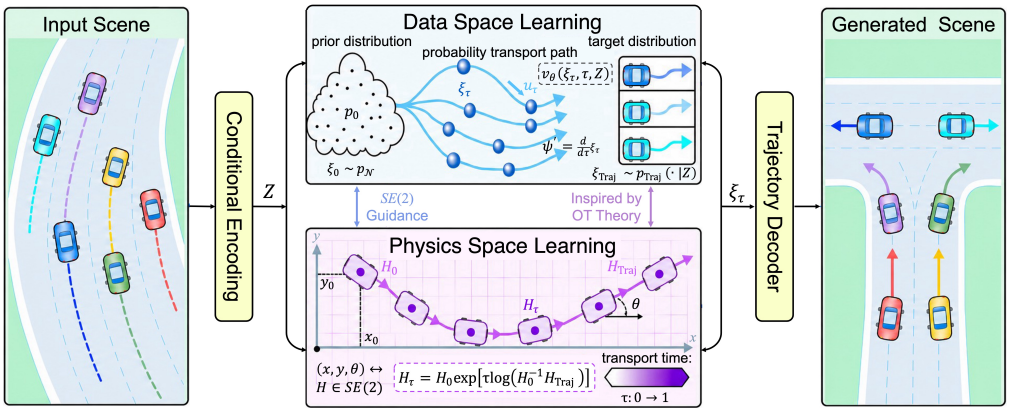


Figure 3: Overview of the proposed SEA for vehicle trajectory generation. The SEA realizes trajectory generation through data space distribution learning and an $SE(2)$-aware conditional transport path, where intermediate states are constructed on the physical pose space.

4.2 Task Definition

Vehicle trajectory generation aims to synthesize realistic trajectories of traffic participants over a time horizon T. Specifically, we generate a traffic scenario $\mathcal{S} = \{Y_1, \ldots, Y_N\}$ consisting of N jointly generated vehicle trajectories, conditioned on the observed states of all vehicles in the scene. Each trajectory is defined as $Y_i = \{s_i^t\}_{t=1}^T$, where s_i^t denotes the state of vehicle i at discrete time step t and comprises its position, heading, velocity and size.

In this paper, we use ξ (instead of x in Section 3.1) to denote a trajectory-level sample from a distribution p. To distinguish from trajectory time step t, we use $\tau \in [0, 1]$ to denote interpolation time along the probability transport path and ξ to denote a scene-level sample comprising joint

[3] A metric $\langle -, - \rangle$ on G is called left-invariant (resp. right-invariant) iff

$$\langle u, v \rangle_b = \left\langle (dL_a)_b\, u, (dL_a)_b\, v \right\rangle_{ab} \quad \text{resp.} \quad \langle u, v \rangle_b = \left\langle (dR_a)_b\, u, (dR_a)_b\, v \right\rangle_{ba}$$

for all $a, b \in G$ and all $u, v \in T_bG$.

[4] A Riemannian metric that is both left and right-invariant is called a bi-invariant metric.

trajectories. Specifically, the source sample ξ_0, ground truth joint trajectory sample ξ_{Traj} and continuous probability path $\mathcal{P}$ connecting their distributions are given by

$$\xi_0 \sim p_{\mathcal{N}}, \;\; \xi_{\text{Traj}} \sim p_{\text{Traj}}(\cdot \mid Z), \;\; \mathcal{P} = \{p_\tau\}_{\tau \in [0,1]}, \;\; p_0 = p_{\mathcal{N}}, \;\; p_1 = p_{\text{Traj}}(\cdot \mid Z), \tag{4}$$

where $p_{\mathcal{N}}$ denotes a simple source distribution (e.g., Gaussian) and $p_{\text{Traj}}(\cdot \mid Z)$ denotes the target joint trajectory distribution conditioned on scene representation Z.

4.3 Context Conditional Encoding

We first encode the lane geometry and observed states into map E_l and vehicle E_{veh} representations:

$$E_l = \Phi_l\left(l_j\right), \quad E^i_{\text{veh}} = \Phi_s\left[\text{PE}\left(s_i\right)\right] + \Phi_q\left(q_i\right),$$

where Φ_l is implemented using Multi-Context Gating [36] while Φ_s and Φ_q denote MLP projections. Here, l_j denotes the positional representation of lane segment j, where $j \in 1, \ldots, M$ and M is the total number of lane segments. Additionally, q_i denotes a learnable query inspired by [37].

We then aggregate map context into vehicle representations through cross-attention CA, followed by self-attention SA to capture inter-vehicle dependencies:

$$\widetilde{Z^i} = \text{CA}\left(E^i_{\text{veh}}, E_l, E_l\right), \quad \widetilde{Z} = \left\{\widetilde{Z^i}\right\}_{i=1}^{N}, \quad Z = \text{SA}\left(\widetilde{Z}\right), \tag{5}$$

The resulting joint representation $Z = \{Z_i\}_{i=1}^{N}$ serves as the deterministic condition for the joint multi-vehicle trajectory distribution $p_{\text{Traj}}(\cdot \mid Z)$.

4.4 Data Space Learning

The context representation Z provides scene-level conditions for modeling the joint trajectory distribution and its continuous evolution in the data space.

Long Tail Vehicle Motions Existing vehicle trajectory generation models are largely data-driven. For example, given a historical trajectory X and scene context information c, a typical model generates a trajectory as $Y = \text{MLP}(X, c)$. However, such architectures often lack appropriate inductive biases [38], making them vulnerable to distribution shifts. As a result, they often fail to capture the full spectrum of motion modes and instead collapse toward dominant patterns in the training data.

Figure 4 illustrates this dataset characteristic on the Waymo dataset [39]: Class 1 (straight) accounts for nearly 70% of the samples, and the remaining classes represent turning or stationary motions (see Figure 8 for definitions), yielding a pronounced long tail motion distribution.

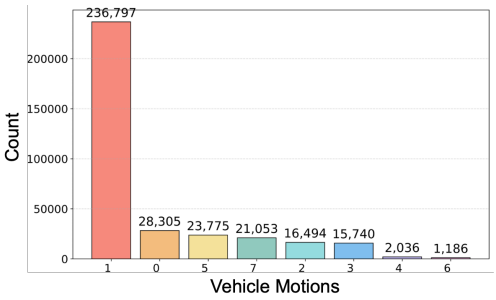


Figure 4: Motion class distribution in the Waymo dataset. Class 1 denotes straight while Class 0 denotes stationary vehicles, the remaining classes correspond to turning (see Figure 8 for definitions).

Distribution Evolution Based on the probability transport path $\mathcal{P}$ defined in Equation (4), we further characterize its marginal evolution through sample-level conditional probability paths. Following [25], the marginal path is induced by

$$p_\tau\left(\xi \mid Z\right) = \int p_\tau\left(\xi \mid \xi_{\text{Traj}}, Z\right) p_{\text{Traj}}\left(\xi_{\text{Traj}} \mid Z\right) d\xi_{\text{Traj}}.$$

Here, $p_\tau(\xi \mid \xi_{\text{Traj}}, Z)$ denotes the conditional probability path associated with ξ_{Traj}, whose marginalization over ξ_{Traj} yields $p_\tau(\xi \mid Z)$ in $\mathcal{P}$. For each ξ_{Traj}, we denote by ψ_τ the flow map associated with the corresponding conditional path. Accordingly, the sample-level evolution follows

$$\xi_\tau = \psi_\tau\left(\xi_0 \mid \xi_{\text{Traj}}, Z\right) \sim p_\tau\left(\xi \mid \xi_{\text{Traj}}, Z\right). \tag{6}$$

The corresponding interpolation vector field u_τ satisfies

$$\begin{cases} \frac{d}{d\tau}\xi_\tau = u_\tau\left(\xi_\tau \mid \xi_{\text{Traj}}, Z\right) \\ \xi_{\tau=0} = \xi_0. \end{cases}, \tag{7}$$

The following physical-space formulation further constrains this sample-level transport according to the inherent geometric structure of vehicle motion.

4.5 Physical Space Learning

Building on the transport formulated in Section 4.4, a question arises: how should intermediate transport states be realized while respecting the physical structure of vehicles? We address this by constructing these states through an $SE(2)$-consistent interpolation between endpoint vehicle poses.

Limitations of Euclidean Transfer Paths Existing methods [21, 40] formulate the transport process in Euclidean data space, where the interpolation in Equation (4) is defined as:

$$\xi_\tau = \tau\xi_{\text{Traj}} + (1-\tau)\,\xi_0.$$

However, a real-world vehicle motion does not strictly satisfy Euclidean assumptions and exhibits distinct physical characteristics. Such a straight transfer path may deviate from the underlying trajectory distribution, as illustrated on the left side of Figure 6. Meanwhile, prior studies [41, 42, 43] model vehicles as planar rigid bodies undergoing rotation and translation.

Vehicle States as Planar Poses on $SE(2)$ Following [44], we examine a rigid body approximation for vehicle motion using the lateral body frame velocity v_y^{body} and sideslip angle β. The former measures lateral motion relative to the vehicle heading, while the latter measures the angular difference between the heading and velocity directions. These quantities are defined as [45, 46]

$$v_x^{\text{body}} = v_x\cos\theta + v_y\sin\theta, \quad v_y^{\text{body}} = -v_x\sin\theta + v_y\cos\theta, \quad \beta = \text{atan2}\left(v_y^{\text{body}}, v_x^{\text{body}}\right).$$

Here, θ denotes the vehicle heading angle, while v_x and v_y denote the velocity components along the global x- and y-axes, respectively, which define the fixed Cartesian coordinate system of the scene. Figure 5 reports the distributions of v_y^{body} and β on the Waymo dataset [47]. We observe that v_y^{body} is sharply concentrated around zero and that most sideslip angles remain small. These observations indicate that vehicle motion largely follows a near zero lateral slip regime, supporting its approximation as planar rigid body motion, with only a small fraction of samples exhibiting noticeable slip induced deviations.

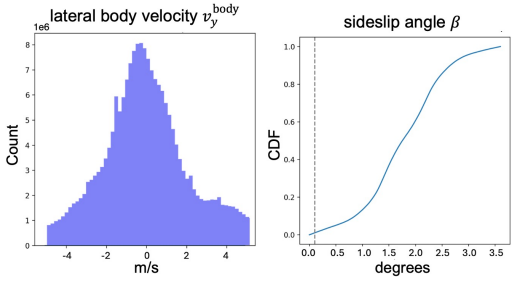


Figure 5: Rigid-body motion characteristics: lateral body frame velocity v_y^{body} is concentrated near zero, and most vehicles exhibit small sideslip angles β, supporting the planar rigid-body approximation of vehicle motion.

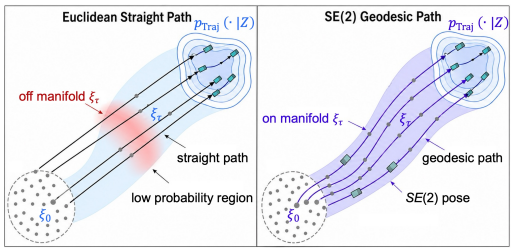


Figure 6: Comparison between the Euclidean straight path (left) and the $SE(2)$ path (right) for distribution transfer. The left path easily deviates from the underlying state space geometry, whereas the right path remains consistent with the geometry of the pose space.

Geometry Consistent Transport Path Realization At the transport-path level, we preserve the endpoint distributions while modifying the realization of sample-level intermediate states. Specifically, we employ an $SE(2)$-consistent interpolation between endpoint vehicle poses, where each pose is characterized by its position (x, y) and heading angle θ. We further motivate this construction from an optimal transport perspective [48, 49]:

Proposition 4.1 (Wasserstein Geodesics as Displacement Interpolations) *Let $(X, \star)$ be a complete pointed locally compact length space and let $\{\mu_t\}_{t\in[0,1]}$ be a geodesic path in $P_2(X)$. An optimal dynamical transference plan Π exists s.t. $\{\mu_t\}_{t\in[0,1]}$ is its displacement interpolation.*

☆ **Implication.** Inspired by this property, we view $p_{\mathcal{N}}$, $p_{\mathrm{Traj}}(\cdot \mid Z)$ and $p_\tau(\cdot \mid Z)$ as elements of $P_2(X)$, where X denotes the vehicle trajectory state space. Proposition 4.1 highlights the role of the underlying state-space geometry in distribution interpolation, rather than implying that our probability path itself constitutes a Wasserstein geodesic (see Definition A.1).

For clarity, we omit the vehicle index i and trajectory-time index t in the following formulation. For each pose component of the endpoint trajectory samples $(\xi_0, \xi_{\mathrm{Traj}})$, we extract the position (x, y) and heading angle θ and denote the corresponding pose coordinates as

$$\zeta_0 = (x_0, y_0, \theta_0), \quad \zeta_{\mathrm{Traj}} = (x_{\mathrm{Traj}}, y_{\mathrm{Traj}}, \theta_{\mathrm{Traj}}) \in \mathbb{R}^3.$$

Thus, ζ_0 and ζ_{Traj} denote pose-level components extracted from the trajectory-level samples ξ_0 and ξ_{Traj}, respectively. The $SE(2)$ construction operates on these pose components, while velocity and size remain in the full trajectory representation and conditional encoding. Details are provided in Appendix A.3. Following Definition 2, we represent the endpoint poses ζ_0 and ζ_{Traj} by $H_0, H_{\mathrm{Traj}} \in SE(2)$, respectively, and construct their $SE(2)$-consistent interpolation according to Eq. (3):

$$H_\tau = H_0 \exp\left[\tau \log\left(H_0^{-1} H_{\mathrm{Traj}}\right)\right], \ \ \tau \in [0, 1]. \tag{8}$$

The pose coordinates ζ_τ are recovered from H_τ and incorporated into the trajectory state ξ_τ in Equation (6). The resulting representation at $\tau = 1$ is then passed to a Transformer decoder to generate vehicle trajectories. Further geometric interpretation and a local W_2 bound between the $SE(2)$-induced path and optimal interpolation are provided in Appendix A.4.

4.6 Loss Function

For each vehicle i in scenario ω with generation query q_i, we retain the conditional supervision term:

$$L^i_{\text{data}}(\omega) = L^i_{\text{cro}}(q_i, \omega) + L^i_{\text{GMM}}(q_i, \omega) + L^i_{\text{MSE}}(q_i, \omega). \tag{9}$$

The detailed definitions of these terms are provided in Appendix C. We retain data space supervision, while the generation head is trained to match the velocity induced by the fixed $SE(2)$-consistent transport path:

$$L^i_{\text{d-p}}(\theta) = \left\| v^i_\theta(\xi_\tau, \tau, Z) - \frac{d}{d\tau}\psi^i_\tau(\xi_0 \mid \xi_{\text{Traj}}, Z) \right\|_2^2.$$

In implementation, the target velocity is obtained by differentiating the intermediate trajectory state ξ_τ, whose pose components are recovered from the $SE(2)$-consistent interpolation in Equation (8). This derivative is computed through automatic differentiation.

For a scenario containing N vehicles, the overall training objective is defined as

$$L = \frac{1}{N}\sum_{i=1}^{N}\left(L^i_{\text{data}} + \lambda L^i_{\text{d-p}}\right).$$

5 Experiments

Datasets. We evaluate our method on two large-scale real-world datasets: the Waymo Open Dataset (Waymo) [47] and nuPlan [50]. Both datasets are collected across multiple cities and contain diverse vehicle motion patterns. We split each dataset into training and test sets in an 8:2 ratio and generate 5-second trajectories at 10Hz, *i.e.*, $T = 50$ time steps.

Metrics. Given ground truth trajectories, we quantify the realism of generated trajectories following the definitions in [4]. These metrics include: (1) distance errors, including minimum Average/Final Displacement Error (minADE/minFDE) and Average/Final Displacement Error (mADE/mFDE); (2) Hausdorff Distance (HD) and (3) Scenario Collision Rate (SCR). Due to space limitations, the detailed formulations of these metrics are provided in Appendix D.

5.1 Quantitative Results

We evaluate our method on the Waymo Open Motion Dataset (WOMD) [47] and nuPlan [50]. Table 1 reports results using the metrics defined above. For all metrics, lower is better.

Compared with InteractTraj [4], our method achieves the best mFDE on Waymo, reducing it from 2.190 to 2.176 (0.64% relative improvement), while mADE increases from 1.067 to 1.082 (+1.41%) remaining competitive overall. For the per scenario best sample, minADE and minFDE improve by 55.80% and 67.19%, respectively. Among all metrics, the largest improvement is observed on minFDE indicating SEA substantially improves the quality of the best generated trajectory. Meanwhile, HD decreases 2.88%, indicating trajectories closer to the real distribution and SCR decreases slightly by 2.86%, suggesting improved realism and accuracy without sacrificing safety.

Table 1: Evaluation of trajectory generation results on WOMD and nuPlan. Lower values indicate better performance, the **best** and second-best results are highlighted.

Dataset	Method	mADE	mFDE	minADE	minFDE	SCR	HD
Waymo	TrafficGen [2]	9.531	20.106	1.440	3.690	0.086	5.733
	MotionDiffuser [51]	8.271	9.478	1.064	1.128	0.174	2.616
	GoalFlow [21]	4.868	12.013	0.310	0.164	0.246	2.236
	SceneDiffuser++ [52]	1.381	3.477	0.464	1.258	0.098	2.025
	WcDT [20]	2.045	-	1.472	-	-	-
	LCTGen [3]	1.262	2.696	0.224	0.463	0.072	1.295
	InteractTraj [4]	**1.067**	2.190	0.181	0.320	0.070	1.076
	SEA	1.082	**2.176**	**0.080**	**0.105**	**0.068**	**1.045**
nuPlan	TrafficGen [2]	9.418	19.686	1.416	3.627	0.082	5.874
	LCTGen [3]	1.161	2.497	0.218	0.448	0.074	1.301
	InteractTraj [4]	0.962	1.987	0.160	**0.321**	0.067	**1.129**
	SEA	**0.901**	**1.770**	**0.146**	0.339	**0.060**	1.133

On nuPlan, compared with the best baseline InteractTraj [4], our method attains the best performance on four out of six metrics: mADE drops from 0.962 to 0.901 (6.34%), mFDE from 1.987 to 1.770 (10.92%), minADE from 0.160 to 0.146 (8.75%), and SCR from 0.067 to 0.060 (10.45%). The remaining two metrics are highly competitive: minFDE is 0.339 vs. 0.321 (5.61% higher) and HD is 0.35% higher, indicating that we improve accuracy and safety substantially while matching the strongest baseline on distributional realism.

5.2 Generation Visualization

Figure 7 visualizes diverse multi-vehicle interactions generated by SEA, including cooperative (left) and competitive (right) driving behaviors, where colored rectangles represent vehicles and trailing curves denote their generated trajectories. Specifically, in (a), the green, purple and gray vehicles behind the red vehicle exhibit adapt their motions to available traffic gaps. Panels (b) and (e) illustrate contrasting lane changing behaviors: in (b), the green vehicle changes lanes cooperatively alongside the red vehicle, whereas in (e), the orange vehicle attempts to merge into a lane with closely spaced surrounding blue and red vehicles. In (c), the purple vehicle yields to the straight moving pink vehicle while turning right. Panels (d) and (g) demonstrate contrasting crossing behaviors: in (d), the pink vehicle coordinates its left turn with approaching traffic, whereas in (g), the gray vehicle crosses the paths of other vehicles, creating a potential conflict. In (f), the green vehicle cuts into the red vehicle's path, while in (h), the green and blue vehicles follow intersecting trajectories through a shared merging area. These results demonstrate that SEA generates diverse multi-vehicle interactions, including competitive behaviors, while preserving the geometric consistency of vehicle motion. This highlights the complementary roles of data-driven distribution learning and geometry-consistent physical realization. Additional qualitative results are provided in Appendix G.

5.3 Ablation Study

To systematically evaluate the effectiveness and generation capabilities of SEA, we conduct ablation studies on the Waymo [47] dataset from three perspectives:

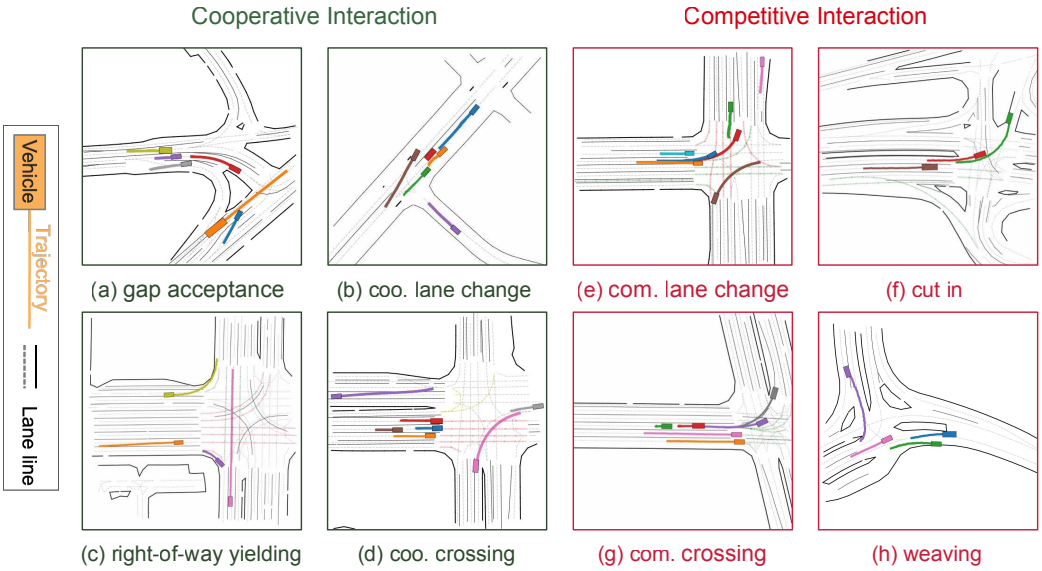


Figure 7: Visualization of generated trajectories across diverse driving scenarios. The left and right panels illustrate cooperative interactions (coo.) and competitive interactions (com.), respectively. Rectangles represent vehicles while trailing curves indicate their generated trajectories.

(i) **Data Space**: Alternative strategies for handling heterogeneous motion distributions (Section 5.3.1) and fine-grained motion dynamics (Appendix B.1).

(ii) **Physical Space**: Generation across different motion patterns (Section 5.3.2) and alternative manifold geometries (Appendix B.2).

(iii) **Safety and Robustness**: Traffic rule compliance (Appendix B.3), robustness to noise perturbations (Appendix B.4) and robustness to masked inputs (Appendix B.5).

5.3.1 Long Tail Method Compare for Data Space Learning

Long tail motion distributions are common in real world traffic scenarios. We therefore compare SEA with these methods by replacing our data space learning module with their core mechanisms on the same backbone. Specifically, BNS [53]: Before generation by learning compact vehicle motion and re-sampling informative tailed motion trajectories to mitigate head motion bias. LoTNext [54]: At the loss level via its Long-Tailed Loss Adjustment mechanism. RATD [55]: Before decoding, reference-guided trajectories are retrieved and fused as additional features. PIR [56]: After the decoder as a post-hoc refinement that identifies high error trajectories and revises them using contextual features. To more clearly evaluate how closely the generated trajectories match real vehicle trajectories, we compute several vehicle dynamics metrics [57] for these methods and the ground truth (GT). The results are summarized in Table 2.

Table 2: Dynamic metric comparison between generated trajectories and Ground Truth (GT). Values closest to GT are highlighted, lower Wasserstein distance W_d indicates a distribution closer to GT.

Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
GT	18.111	2.208	0.060	0.007	0.003	-0.341	14.337	118.889	-
BNS [53]	18.712	16.539	0.036	-0.001	0.001	-0.386	72.748	70.253	2.369
LoTNext [54]	18.578	20.261	0.012	0.011	0.002	-1.422	69.758	45.138	1.708
RATD [55]	19.564	40.018	0.073	-0.005	0.001	-0.526	72.203	52.789	3.918
PIR [56]	18.911	43.003	0.063	0.018	0.001	-6.429	79.326	43.787	3.037
SEA	18.285	15.469	0.077	0.029	0.002	-0.369	68.656	67.949	1.646

These metrics are velocity v, acceleration a, curliness (cur), heading θ, heading-rate R_θ, jerk j, space headway S_h, time headway T_h and Wasserstein distance W_d. Detailed definitions are provided in Appendix B.1. Compared with the GT dynamics, SEA produces the closest values on $5/8$ comparable

metrics. Across all comparable metrics, SEA obtains the smallest average absolute deviation from GT, i.e., 14.970, outperforming BNS (15.257), LoTNext (18.603), RATD (20.430) and PIR (23.474). The closest individual deviation of SEA appears for R_θ with an absolute gap of only 0.001 from GT, while its deviations for v and j remain small (0.174 and 0.028, respectively). SEA achieves lower acceleration deviation than competing methods, indicating that $SE(2)$-aware transport better preserves the dynamic characteristics of real-world vehicle trajectories.

5.3.2 Comparison Across Motion Classes for Physical Space Learning

To evaluate the effectiveness of $SE(2)$-aware path realization across different motion patterns, we compare SEA with LCTGen [3] across fine-grained motion classes, as illustrated in Figure 8. For visualization, radial values are symmetrically scaled around a common center using the amplified relative difference $(x_{\text{LCTGen}} - x_{\text{SEA}})/\max(|x_{\text{LCTGen}}|, |x_{\text{SEA}}|)$, where larger radii indicate better performance.

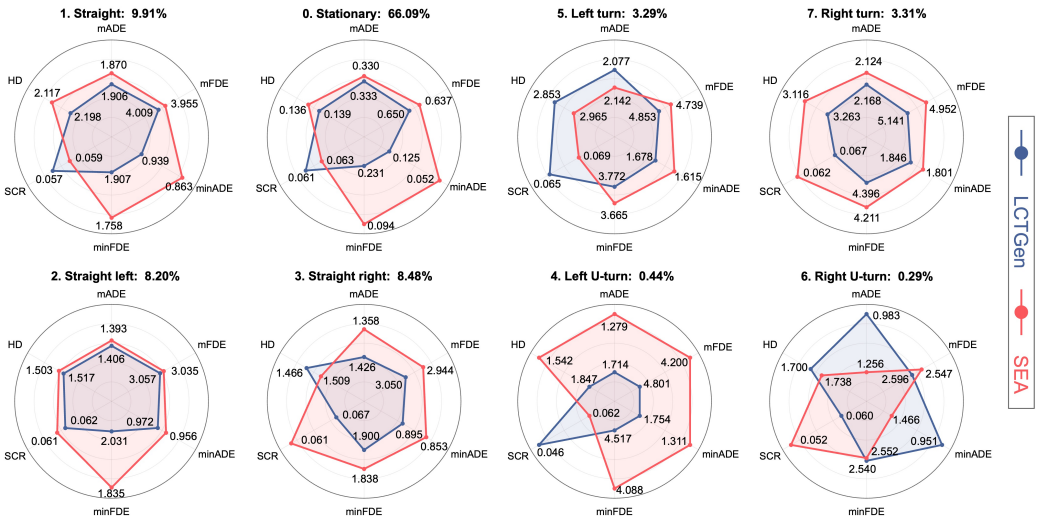


Figure 8: Motion wise generation performance of LCTGen [3] and SEA. Radial values are normalized such that larger radii indicate better performance. Motion-class frequencies are reported as percentages, with class indices corresponding to the dataset labels.

Across heterogeneous motion patterns, SEA improves trajectory generation performance for prevalent and turning motions. Compared with LCTGen, SEA reduces minADE by 8.1% and 58.4% for Straight and Stationary motions, respectively, with similar improvements across turning motions. For less frequent U-turns (0.73%), SEA reduces mADE by 25.4% for left U-turns and improves mFDE and HD for right U-turns. Regarding collision safety, SEA reduces SCR for straight right, right turn and right U-turn motions, although SCR increases in other classes, particularly left U-turns (from 0.046 to 0.062). Overall, these results highlight SEA's effectiveness in modeling heterogeneous motion distributions while indicating that collision avoidance remains challenging for certain motions.

6 Conclusion and Discussion

We introduced an $SE(2)$-aware (SEA) conditional transport path realization for vehicle trajectory generation, which injects planar rigid body pose geometry into the intermediate states of the generative transport process, improving pose-level geometric consistency and empirical dynamic realism. Experiments on large scale datasets show that SEA outperforms state-of-the-art methods, better respects vehicle dynamic constraints and consistently reduces errors across motion classes. More broadly, this perspective suggests a general geometry aware recipe for generative modeling on structured state spaces: when the target states lie on a task specific manifold, the transport process can be realized through manifold consistent intermediate states, thereby providing a principled way to incorporate domain geometry into distribution generation beyond the vehicle trajectory setting.

Acknowledgments and Disclosure of Funding

This research was supported by the National Key Research and Development Program of China (2023YFB4301900), the Shenzhen Science and Technology Program (JCYJ20240813151445059), the National Natural Science Foundation of China (62236008), the Natural Science Foundation of Beijing (L251082) and the Shenzhen Key Industry Research Program (ZDCY20250901112501002).

References

[1] Kashif Naseer Qureshi and Abdul Hanan Abdullah. A survey on intelligent transportation systems. *Middle-East Journal of Scientific Research*, 15(5):629–642, 2013.

[2] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575, 2023.

[3] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. In *7th Annual Conference on Robot Learning*, 2023.

[4] Junkai Xia, Chenxin Xu, Qingyao Xu, Yanfeng Wang, and Siheng Chen. Language-driven interactive traffic trajectory generation. *Advances in Neural Information Processing Systems*, 37:77831–77859, 2024.

[5] Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Zhao Huang, Yipeng Wu, Die Zuo, Jibin Peng, Ziyuan Zhong, Xin Wang, et al. Trajectory-llm: A language-based data generator for trajectory prediction in autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[6] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. Ieee, 2018.

[7] Martin Fellendorf and Peter Vortisch. Microscopic traffic flow simulator vissim. In *Fundamentals of traffic simulation*, pages 63–93. Springer, 2010.

[8] Xiangjie Kong, Qiao Chen, Mingliang Hou, Hui Wang, and Feng Xia. Mobility trajectory generation: a survey. *Artificial Intelligence Review*, 56(Suppl 3):3057–3098, 2023.

[9] Xin Chen, Chengrui Huang, Chenhao Wang, and Lisi Chen. Trajectory generation: a survey on methods and techniques. *GeoInformatica*, pages 1–26, 2025.

[10] Tonglong Wei, Youfang Lin, Shengnan Guo, Yan Lin, Yiheng Huang, Chenyang Xiang, Yuqing Bai, and Huaiyu Wan. Diff-rntraj: A structure-aware diffusion model for road network-constrained trajectory generation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[11] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4676–4687, 2024.

[12] Qichao Zhang, Yinfeng Gao, Yikang Zhang, Youtian Guo, Dawei Ding, Yunpeng Wang, Peng Sun, and Dongbin Zhao. Trajgen: Generating realistic and diverse trajectories with reactive and feasible agent behaviors for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24474–24487, 2022.

[13] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024.

[14] Jia Jia, Ximing Li, Ning Wang, Binsi Cai, Xu Kang, Xu Zhang, and Pengfei Qiu. High-quality trajectory generation via domain-knowledge enhanced gans. *IEEE Internet of Things Journal*, 2025.

[15] Zhongcai Cao, Kang Liu, Xin Jin, Li Ning, Ling Yin, and Feng Lu. Stage: a spatiotemporal-knowledge enhanced multi-task generative adversarial network (gan) for trajectory generation. *International Journal of Geographical Information Science*, 39(5):1100–1127, 2025.

[16] Domenico Bianchi, Alessandro Borri, Federico Cappuzzo, and Stefano Di Gennaro. Quadrotor trajectory control based on energy-optimal reference generator. *Drones*, 8(1):29, 2024.

[17] Rami Ammourah, Pedram Beigi, Bingyi Fan, Samer H Hamdar, John Hourdos, Chun-Chien Hsiao, Rachel James, Mohammdreza Khajeh-Hosseini, Hani S Mahmassani, Dana Monzer, et al. Introduction to the third generation simulation dataset: Data collection and trajectory extraction. *Transportation Research Record*, 2679(1):1768–1784, 2025.

[18] Yueyuan Li, Wei Yuan, Songan Zhang, Weihao Yan, Qiyuan Shen, Chunxiang Wang, and Ming Yang. Choose your simulator wisely: A review on open-source simulators for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 9(5):4861–4876, 2024.

[19] Farnaz Behnia, Dominik Karbowski, and Vadim Sokolov. Deep generative models for vehicle speed trajectories. *Applied Stochastic Models in Business and Industry*, 39(5):701–719, 2023.

[20] Chen Yang, Yangfan He, Aaron Xuxiang Tian, Dong Chen, Jianhui Wang, Tianyu Shi, Arsalan Heydarian, and Pei Liu. Wcdt: World-centric diffusion transformer for traffic scene generation. *arXiv preprint arXiv:2404.02082*, 2024.

[21] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1602–1611, 2025.

[22] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

[23] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023.

[24] Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Zhao Huang, Yipeng Wu, Die Zuo, Jibin Peng, Ziyuan Zhong, Xin Wang, et al. Trajectory-llm: A language-based data generator for trajectory prediction in autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[26] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[28] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.

[29] Valentin Sonneville, Alberto Cardona, and Olivier Brüls. Geometrically exact beam finite element formulated on the special euclidean group se(3). *Computer Methods in Applied Mechanics and Engineering*, 268:451–474, 2014.

[30] Pu You, Zhuyong Liu, and Ziqi Ma. A two-dimensional corotational beam formulation based on the local frame of special euclidean group se (2). *Journal of Computational and Nonlinear Dynamics*, 18(5):051005, 2023.

[31] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017.

[32] Claude Chevalley. *Theory of Lie groups*. Courier Dover Publications, 2018.

[33] Jean Gallier and Jocelyn Quaintance. *Metrics, Connections, and Curvature on Lie Groups*. Springer International Publishing, 2020.

[34] Joachim Hilgert and Karl-Hermann Neeb. *Structure and geometry of Lie groups*. Springer Science & Business Media, 2011.

[35] Marcos M Alexandrino and Renato G Bettiol. Lie groups with bi-invariant metrics. In *Lie groups and geometric aspects of isometric actions*, pages 27–47. Springer, 2015.

[36] Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, et al. Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821. IEEE, 2022.

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

[38] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

[39] Waymo Research. motion_metrics_utils.cc – evaluation code for the waymo open dataset metrics. `https://github.com/waymo-research/waymo-open-dataset/blob/master/src/waymo_open_dataset/metrics/motion_metrics_utils.cc`, 2023.

[40] Yuxiang Fu, Qi Yan, Lele Wang, Ke Li, and Renjie Liao. Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17282–17293, 2025.

[41] Steffen Hagedorn, Marcel Milich, and Alexandru P Condurache. Pioneering se (2)-equivariant trajectory planning for automated driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2097–2102. IEEE, 2024.

[42] Jiaying Chen, Han Wang, Minghui Hu, and Ponnuthurai Nagaratnam Suganthan. Versatile lidar-inertial odometry with se (2) constraints for ground vehicles. *IEEE Robotics and Automation Letters*, 8(6):3486–3493, 2023.

[43] Li-Yu Lin, James Goppert, and Inseok Hwang. Log-linear dynamic inversion control with provable safety guarantees in lie groups. *IEEE Transactions on Automatic Control*, 69(8):5591–5597, 2024.

[44] Haiping Du, Nong Zhang, and Fazel Naghdy. Velocity-dependent robust control for improving vehicle lateral dynamics. *Transportation research part C: emerging technologies*, 19(3):454–468, 2011.

[45] Haiping Du, Nong Zhang, and Fazel Naghdy. Velocity-dependent robust control for improving vehicle lateral dynamics. *Transportation research part C: emerging technologies*, 19(3):454–468, 2011.

[46] Jizheng Liu, Zhenpo Wang, Lei Zhang, and Paul Walker. Sideslip angle estimation of ground vehicles: A comparative study. *IET Control Theory & Applications*, 14(20):3490–3505, 2020.

[47] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[48] John Lott and Cédric Villani. Ricci curvature for metric-measure spaces via optimal transport. *Annals of Mathematics*, 169(3):903–991, 2009.

[49] Robert J. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1):153–179, 1997.

[50] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024.

[51] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9644–9653, 2023.

[52] Shuhan Tan, John Lambert, Hong Jeon, Sakshum Kulshrestha, Yijing Bai, Jing Luo, Dragomir Anguelov, Mingxing Tan, and Chiyu Max Jiang. Scenediffuser++: City-scale traffic simulation via a generative world model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1570–1580, 2025.

[53] Fudong Lin and Xu Yuan. Long-tailed recognition via information-preservable two-stage learning. In *Advances in Neural Information Processing Systems*, 2025. NeurIPS 2025.

[54] Xiaohang Xu, Renhe Jiang, Chuang Yang, Kaoru Sezaki, et al. Taming the long tail in human mobility prediction. *Advances in Neural Information Processing Systems*, 37:54748–54771, 2024.

[55] Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. *Advances in Neural Information Processing Systems*, 37:2766–2786, 2024.

[56] Zhiding Liu, Mingyue Cheng, Guanhao Zhao, Jiqian Yang, Qi Liu, and Enhong Chen. Improving time series forecasting via instance-aware post-hoc revision. *Improving Time Series Forecasting via Instance-aware Post-hoc Revision*, 2025.

[57] Malte Risto and Marieke H Martens. Time and space: The difference between following time headway and distance headway instructions. *Transportation research part F: traffic psychology and behaviour*, 17:45–51, 2013.

[58] Jean Gallier and Jocelyn Quaintance. Introduction to manifolds and lie groups. Chapter 2 of CIS 610 lecture notes, University of Pennsylvania, 2013.

[59] Xinhu Zheng, Pengtao Yang, Dongliang Duan, Xiang Cheng, and Liuqing Yang. Real-time driving style classification based on short-term observations. *IET Communications*, 16(12):1393–1402, 2022.

[60] John M Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[61] John G Ratcliffe. *Foundations of hyperbolic manifolds*. Springer, 2006.

[62] Shin-ichi Ohta. Nonlinear geometric analysis on finsler manifolds. *European Journal of Mathematics*, 3(4):916–952, 2017.

[63] Yangdi Lu, Yang Bo, and Wenbo He. Noise attention learning: Enhancing noise robustness by gradient scaling. *Advances in Neural Information Processing Systems*, 35:23164–23177, 2022.

[64] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pages 358–374. Springer, 2022.

[65] Xiang Li, Mingbao Zhang, Adriano Jose Tavares, Hao Xu, and Shijun Ma. Personalized vehicle trajectory prediction method based on driving style classification. *Scientific Reports*, 15(1):31587, 2025.

[66] Yi Xu and Yun Fu. Gaussian mixture conditional variational recurrent neural network for unified trajectory imputation and prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Appendix for $SE(2)$-Aware Conditional Distribution Transport for Vehicle Trajectory Generation

A Geometric Background for Probability Transport on $SE(2)$

A.1 From $SO(n)$ to $SE(n)$: rigid motions as geometric state spaces

Rigid body motions act on the physical space $\mathbb{R}^n$. The set of orientation preserving Euclidean isometries of $\mathbb{R}^n$ is the special Euclidean group $SE(n)$, whose elements combine a rotation in $SO(n)$ with a translation in $\mathbb{R}^n$. Here the special orthogonal group $SO(n)$ is defined by $SO(n) = \mathrm{O}(n) \cap SL(n, \mathbb{R})$.

In homogeneous coordinates, we can view every element of $SE(n)$ as a real $(n+1) \times (n+1)$ matrix H [31]:

$$H = \begin{pmatrix} R & U \\ 0 & 1 \end{pmatrix}, \quad R \in SO(n), \quad U \in \mathbb{R}^n.$$

This embedding of $SE(n)$ into $SL(n+1)$ is a group homomorphism [33, 58], since the group operation on $SE(n)$ corresponds to matrix multiplication in $SL(n+1)$:

$$\begin{pmatrix} RS & RV+U \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R & U \\ 0 & 1 \end{pmatrix} \begin{pmatrix} S & V \\ 0 & 1 \end{pmatrix}. \tag{10}$$

The inverse is given by

$$\begin{pmatrix} R^{-1} & -R^{-1}U \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} R^T & -R^TU \\ 0 & 1 \end{pmatrix}, \tag{11}$$

where the equality follows from $R^{-1} = R^T$ for $R \in SO(n)$. Moreover, this embedding shows that, as a manifold, $SE(n)$ is diffeomorph to $SO(n) \times \mathbb{R}^n$. Thus, $SE(n)$ forms a Lie group with closure under matrix multiplication and inversion given by Equations (10) and (11), respectively.

In the planar case relevant to vehicle trajectory modeling, we represent the vehicle pose at discrete time step t by its position and heading: $\zeta_t = (x_t, y_t, \theta_t)$. Equivalently, the pose is represented by the translation vector $U(t)$ and rotation matrix $R(t)$:

$$U(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \quad R(t) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{bmatrix}.$$

Thus, each time step vehicle pose is represented as an element of $SE(2)$ in homogeneous coordinates.

A.2 Wasserstein space and geometry induced interpolation

We provide the definition of the Wasserstein space used in Proposition 4.1 [48] as follows:

Definition A.1 (Wasserstein space) *Let (X, d) be a metric space and let $P(X)$ denote the space of Borel probability measures on X. Endowed with the Wasserstein distance W_2, the metric space $(P(X), W_2)$ is denoted by $P_2(X)$ and is called the* **Wasserstein space**.

Here, W_2 is defined by the optimal transport cost induced by the ground metric d on X. Definition A.1 shows that the Wasserstein space $P_2(X)$ is induced by the ground metric on the underlying state space (X, d). Therefore, interpolation in distribution space should not be specified independently of the geometry of the state space.

We denote by X the joint vehicle trajectory state space (the $s_i^\star$ in Section 4.2), whose pose components are represented on $SE(2)$. This motivates an $SE(2)$-aware realization of the conditional transport path through pose space consistent interpolation. Specifically, the source distribution $p_{\mathcal{N}}$, the target distribution $p_{\mathrm{Traj}}(\cdot \mid Z)$ and the intermediate distributions $p_\tau(\cdot \mid Z)$ are considered as elements of $P_2(X)$, where Z denotes the joint scene representation. The conditional transport path is realized through $SE(2)$-consistent interpolation between endpoint poses. The resulting pose coordinates ζ_τ are recovered from H_τ in Equation (8) and constitute the pose components of the joint trajectory sample ξ_τ.

A.3 Left-invariant Riemannian metric and $SE(2)$-aware transport state realization

To describe geometry-consistent interpolation on a Lie group, we consider a left-invariant Riemannian metric, under which every left translation $L_a : G \to G$ is an isometry. Since $SE(2)$ does not admit a bi-invariant Riemannian metric, the exponential coordinate interpolation

$$\gamma(\tau) = g_0 \exp\left[\tau \log\left(g_0^{-1} g_1\right)\right], \quad \tau \in [0, 1],$$

is interpreted as a geometry-consistent interpolation rather than a distance minimizing geodesic under an arbitrary left-invariant Riemannian metric. This point clarifies the role of the Riemannian metric in SEA: it provides the geometric basis for an $SE(2)$-aware realization of transport states, rather than defining interpolation in ambient Euclidean coordinates.

Using a Lie group interpolation does not imply that the conditional probability path is a Wasserstein geodesic or that the model directly solves a Riemannian optimal transport problem. Instead, the $SE(2)$-aware construction provides the pose components of intermediate trajectory samples used to train the flow matching vector field.

The proposed framework involves two complementary levels. At the physical-space level, vehicle poses are represented on $SE(2)$, while complete joint trajectories belong to the trajectory state space X. At the distribution level, the model transports probability from $p_{\mathcal{N}}$ to $p_{\mathrm{Traj}}(\cdot \mid Z)$ through intermediate distributions $p_\tau(\cdot \mid Z)$, which are considered in the Wasserstein space $P_2(X)$. This motivates an $SE(2)$-aware realization of the pose components of intermediate transport states.

A.4 Local Bound for $SE(2)$-Aware Distribution Transfer

To further characterize the proposed $SE(2)$-aware path for vehicle trajectory distribution transfer in Section 4.5, we analyze its deviation from the optimal displacement interpolation. While our construction is not assumed to induce a Wasserstein geodesic, the following proposition establishes a local bound at both the pose path and distribution levels.

Proposition A.2 (Local Bound to Optimal Displacement Interpolation) *Let $X = SE(2)$ be equipped with a left-invariant Riemannian distance d and let $\pi^\star \in \mathrm{Opt}(\mu_0, \mu_1)$ be an optimal coupling on $X \times X$, with endpoint pairs (H_0, H_1) contained in a compact region, where the chosen logarithm is smooth and satisfies*

$$\left\|\log\left(H_0^{-1} H_1\right)\right\|_e \leqslant \kappa d\left(H_0, H_1\right),$$

for some $\kappa < \infty$, where d is the distance induced by a left-invariant Riemannian metric and $\|\cdot\|_e$ is the corresponding norm at the identity. Let γ_{SE} be the proposed $SE(2)$-consistent path in Equation (8) and $\gamma_\star(\tau)$ be the constant speed distance minimizing ground space geodesic that induces the displacement interpolation in Proposition 4.1. Then, for some finite constant C_κ,

$$d\left[\gamma_{SE}(\tau), \gamma_\star(\tau)\right] \leqslant C_\kappa \min\{\tau, 1-\tau\}\; d\left(H_0, H_1\right). \tag{12}$$

Consequently, under the same optimal coupling $\pi^\star$, the induced distribution paths satisfy

$$W_2\left(\mu_\tau^{SE}, \mu_\tau^\star\right) \leqslant C_\kappa \min\{\tau, 1-\tau\}\; W_2\left(\mu_0, \mu_1\right). \tag{13}$$

Proof: For an endpoint pair (H_0, H_1) sampled from $\pi^\star$, let

$$A = \log(H_0^{-1} H_1) \quad s.t. \quad \gamma_{SE}(\tau) = H_0 \exp(\tau A)$$

Pointwise path bound. Under the left-invariant Riemannian metric, γ_{SE} has constant body velocity A. Hence,

$$d\left[H_0, \gamma_{SE}(\tau)\right] \leqslant L\left(\gamma_{\mathrm{SE}}|_{[0,t]}\right) = \tau\|A\|_e \leqslant \tau\kappa d\left(H_0, H_1\right).$$

Since $\gamma_\star$ is a constant speed distance minimizing geodesic connecting the same endpoints,

$$d\left[H_0, \gamma_\star(\tau)\right] = \tau \cdot d\left(H_0, H_1\right).$$

The triangle inequality therefore gives

$$d\left[\gamma_{SE}(\tau), \gamma_\star(\tau)\right] \leqslant (\kappa+1)\,\tau\, d\left(H_0, H_1\right). \tag{14}$$

Similarly, from the terminal endpoint H_1,

$$d\left[\gamma_{SE}(\tau), H_1\right] \leqslant (1-\tau)\,\|A\|_e \leqslant (1-\tau)\,\kappa d\left(H_0, H_1\right),$$

while

$$d\left[\gamma_\star(\tau), H_1\right] = (1-\tau)\, d\left(H_0, H_1\right).$$

Thus,

$$d\left[\gamma_{SE}(\tau), \gamma_\star(\tau)\right] \leqslant (\kappa+1)\,(1-\tau)\, d\left(H_0, H_1\right). \tag{15}$$

Setting $C_\kappa = \kappa + 1$ and combining the bounds in Equations (14) and (15), which proves Equation (12).

Distribution level bound. Define the endpoint dependent maps

$$\mathcal{M}_\tau^{SE}(H_0, H_1) = \gamma_{SE}(\tau), \qquad \mathcal{M}_\tau^\star(H_0, H_1) = \gamma_\star(\tau).$$

Then

$$\mu_\tau^{SE} = \left(\mathcal{M}_\tau^{SE}\right)_\# \pi^\star, \qquad \mu_\tau^\star = \left(\mathcal{M}_\tau^\star\right)_\# \pi^\star,$$

and $\left(\mathcal{M}_\tau^{SE}, \mathcal{M}_\tau^\star\right)_\# \pi^\star$ is an admissible coupling between μ_τ^{SE} and $\mu_\tau^\star$. Therefore,

$$\begin{aligned} W_2^2\left(\mu_\tau^{SE}, \mu_\tau^\star\right) &\leqslant \int_{X\times X} d^2\left[\mathcal{M}_\tau^{SE}\left(H_0, H_1\right), \mathcal{M}_\tau^\star\left(H_0, H_1\right)\right] d\pi^\star(H_0, H_1) \\ &\leqslant C_\kappa^2 \min\{\tau, 1-\tau\}^2 \int_{X\times X} d^2(H_0, H_1)\, d\pi^\star(H_0, H_1). \end{aligned} \tag{16}$$

Since $\pi^\star$ is an optimal coupling between μ_0 and μ_1, we have

$$\int_{X\times X} d^2\left(H_0, H_1\right)\, d\pi^\star\left(H_0, H_1\right) = W_2^2\left(\mu_0, \mu_1\right).$$

Substituting this identity into the preceding inequality (16) and taking square roots on both sides yields Equation (13), completing the proof. □

B Additional Ablation Studies

In this section, we present additional experimental results and ablation studies for the proposed approach. We further compare generated trajectories with ground truth in terms of dynamic metrics across baselines in Appendix B.1. In Section 4.5, we analyze the advantages of the $SE(2)$ manifold for physics guided vehicle trajectory generation. We also investigate the effect of alternative geometric manifolds on the generation process, with detailed results reported in Appendix B.2. To estimate the safety of generated trajectories, compute the ratio of trajectories that violate traffic rules in Appendix B.3. Finally, we evaluate the robustness of our proposed SEA under noise perturbation (Appendix B.4) and trajectory reconstruction conditions (Appendix B.5).

B.1 Dynamic Metrics

We compare the proposed method with three baselines which have identical tasks and evaluate metrics in their paper, and inputs and report averaged value results in Table 3. For each metric [57]: velocity v, acceleration a, curliness (cur), heading θ, heading rate R_θ, jerk j, space headway S_h, time headway T_h and Wasserstein distance W_d, we highlight in bold the value that is closest to the ground truth (GT). Table 3 evaluates the dynamic consistency of generated trajectories under different motion categories. Across all motion types, SEA achieves the closest match to GT on $36/64$ dynamic metric entries, substantially outperforming TrafficGen ($6/64$), LCTGen ($18/64$), and InteractTraj ($7/64$). Moreover, SEA obtains the smallest Wasserstein distance W_d on $7/8$ motion categories, indicating that the generated trajectory distribution is generally closer to the real trajectory distribution. Compared with the strongest baseline in each motion category, SEA outperforms the strongest baseline in each motion category by an average margin of 0.888 in W_d.

For the head motion category, i.e., stationary trajectories, SEA is closest to GT on $6/8$ dynamic metrics and also achieves the smallest W_d. Specifically, SEA exactly matches the GT heading rate with zero gap in R_θ, and obtains small deviations on heading, space headway, and time headway, with gaps of 0.003, 0.922, and 21.986, respectively.

For the remaining long-tail motion categories, SEA remains consistently competitive. It is closest to GT on $30/56$ dynamic metric entries and achieves the lowest W_d on $6/7$ tail categories. Across all dynamic entries, the average absolute deviation from GT is 18.844 for SEA, lower than TrafficGen (23.234), LCTGen (20.767), and InteractTraj (28.135). For the competing methods, the absolute deviations from GT range from 0.000 to 449.759, with an average deviation of 24.045, whereas SEA reduces the average deviation to 18.844.

Notation. For a vehicle, let its trajectory position $p\left(t\right) = \left[x\left(t\right), y\left(t\right)\right]$ in time t and sampling interval Δt. Define discrete velocity v, acceleration a and jerk j:

$$v = \frac{p\left(t+1\right) - p\left(t\right)}{\Delta t},\ a = \frac{v\left(t+1\right) - v\left(t\right)}{\Delta t},\ j = \frac{a\left(t+1\right) - a\left(t\right)}{\Delta t}.$$

it is obviously that jerk is the time derivative of acceleration (equivalently, the 3rd derivative of position):

$$j = \frac{1}{\triangle t^3}\left[p\left(t+3\right) - 3p\left(t+2\right) + 3p\left(t+1\right) - p\left(t\right)\right].$$

In evaluation, we typically use the magnitude $j = \left\|j\left(t\right)\right\|_2$ or per axis components.

Table 3: Dynamic metric comparison between generated trajectories and ground truth (GT) under different motion categories, where the percentage denotes the motion ratio. Values closest to GT are highlighted, and a lower Wasserstein distance W_d indicates a distribution closer to GT.

	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
Straight 66.09%	GT	0.208	0.290	0.166	0.000	-0.00005	0.002	5.893	174.229	-
	TrafficGen	4.924	19.475	0.019	0.106	0.04372	-0.979	0.406	0.489	3.389
	LCTGen	0.671	9.796	0.040	-0.013	0.00005	-0.547	7.262	149.782	1.323
	InteractTraj	4.864	55.311	0.058	0.009	0.00052	-16.307	3.360	12.118	2.679
	SEA	0.674	9.771	0.063	0.003	-0.00005	-0.819	6.815	152.243	0.456
Stationary 9.91%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	30.125	3.108	0.003	-0.001	0.00008	-0.567	15.400	12.272	-
	TrafficGen	24.072	24.359	0.007	0.028	0.00958	-2.471	1.226	0.119	7.130
	LCTGen	30.508	16.995	0.004	0.007	0.00008	-1.223	82.198	8.524	3.405
	InteractTraj	28.466	57.533	0.019	0.008	-0.00026	-12.935	39.491	7.387	3.824
	SEA	30.449	20.194	0.004	-0.005	-0.00007	-1.997	83.089	10.272	2.642
Straight left 8.20%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	38.060	3.601	0.006	0.109	0.035	-0.728	16.160	0.518	-
	TrafficGen	28.402	30.729	0.008	0.125	0.026	-1.546	1.843	0.116	8.175
	LCTGen	37.417	21.394	0.007	0.045	0.026	-1.240	77.924	6.043	3.405
	InteractTraj	36.360	96.014	0.010	0.159	0.034	-12.593	37.246	1.865	7.189
	SEA	37.791	15.885	0.006	0.057	0.026	-2.382	17.495	3.835	4.104
Straight right 8.48%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	34.591	3.689	0.009	-0.146	-0.045	-0.403	27.678	7.095	
	TrafficGen	25.376	27.773	0.010	-0.041	0.002	-2.231	0.746	0.154	8.341
	LCTGen	33.723	22.182	0.009	-0.039	-0.035	-1.372	72.509	2.576	4.571
	InteractTraj	34.057	106.377	0.014	-0.110	-0.051	-11.698	43.101	2.155	4.086
	SEA	34.034	16.975	0.010	-0.125	-0.048	-1.290	26.669	2.907	3.474
Left turn 3.29%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	9.981	4.342	0.073	0.317	0.163	0.324	19.064	173.696	
	TrafficGen	13.670	32.303	0.030	0.344	0.097	-1.869	1.279	0.554	4.243
	LCTGen	10.118	21.455	0.057	0.220	0.147	-2.846	80.215	21.850	3.702
	InteractTraj	9.788	74.021	0.076	0.158	0.114	-5.347	31.005	10.570	3.976
	SEA	10.080	15.988	0.059	0.307	0.150	-2.167	66.881	23.826	2.341
Right turn 3.31%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	10.902	4.405	0.044	-0.304	-0.127	-0.138	28.633	303.977	-
	TrafficGen	11.357	25.336	0.029	-0.029	-0.003	-1.966	1.491	0.430	5.085
	LCTGen	10.226	21.760	0.042	-0.264	-0.113	-1.032	49.347	29.709	3.458
	InteractTraj	10.905	65.386	0.058	-0.251	-0.117	-5.796	45.483	17.201	4.366
	SEA	10.709	25.015	0.044	-0.275	-0.104	-2.317	15.751	27.333	2.162
Left U-turn 0.44%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	4.659	3.298	0.037	-0.007	-0.006	0.899	11.131	10.714	-
	TrafficGen	6.944	23.511	0.018	0.173	0.052	-1.976	0.583	0.581	4.489
	LCTGen	4.458	12.832	0.023	-0.024	-0.008	-0.225	62.736	33.256	0.973
	InteractTraj	2.592	35.824	0.084	0.028	-0.009	-4.999	38.217	26.265	2.781
	SEA	4.838	17.482	0.023	-0.060	-0.006	-1.704	54.604	7.967	0.784
Right U-turn 0.29%	Method	v	a	cur	θ	R_θ	j	S_h	T_h	W_d
	GT	3.980	2.428	0.076	0.003	0.0066	0.714	7.571	450.282	
	TrafficGen	5.804	21.749	0.019	0.081	0.0406	-1.100	0.529	0.523	4.085
	LCTGen	3.905	12.322	0.024	0.097	0.0007	-0.811	71.250	107.105	1.783
	InteractTraj	2.260	33.690	0.085	-0.030	0.0005	-4.767	41.565	28.126	2.978
	SEA	3.939	6.346	0.021	0.005	0.0007	-3.243	69.117	62.726	0.652

Heading θ. We compute heading from the velocity direction $\theta = \text{atan2}\,(v_y, v_x)$, where v_x and v_y denote the velocity components along the x- and y-axes, respectively. Here, $\theta \in (-\pi, \pi]$.

Heading-rate R_θ. Heading-rate is the time derivative of heading angle $\theta = \text{atan2}\,(v_y, v_x)$.

Curliness (cur). We measure trajectory curliness as the ratio of path length to end to end displacement:

$$\text{cur} = \frac{\sum_{t=0}^{T-1} \|p\,(t+1) - p\,(t)\|_2}{\|p\,(T) - p\,(0)\|_2 + \epsilon},$$

where $\epsilon > 0$ ensures numerical stability. For non stationary straight trajectories without direction reversals, cur ≈ 1, while larger values indicate greater path tortuosity.

Space headway S_h. For a following vehicle i and its assigned leader k at time t, the space headway is approximated by the Euclidean center-to-center distance minus half the length of each vehicle [59]:

$$S_h^{(i)} = \|p_k\,(t) - p_i\,(t)\|_2 - \frac{1}{2}\left[L_{\text{veh}}^{(k)} + L_{\text{veh}}^{(i)}\right], \tag{17}$$

where L_{veh} denotes vehicle length.

Time headway T_h. Time headway is defined as the time required for vehicle i to reach the front center position of its preceding vehicle when traveling at its current speed:

$$T_h^{(i)} = S_h^{(i)} \Big/ \left[\left\|v^{(i)}\right\|_2 + \varepsilon\right],$$

where $S_h^{(i)}$ is defined in Equation (17).

Wasserstein distance W_d. Given a scalar statistic z (used in Table 3) computed over a set of samples, let μ_z and ν_z denote the empirical distributions of this statistic for the generated and ground-truth (GT) trajectories, respectively. The p-Wasserstein distance is defined as

$$W_p\,(\mu_z, \nu_z) = \left[\inf_{\gamma \in \Gamma(\mu_z, \nu_z)} \mathbb{E}_{(x,y)\sim\gamma}\left(\|x - y\|_p^p\right)\right]^{1/p},$$

where $\Gamma(\mu_z, \nu_z)$ denotes the set of all couplings between μ_z and ν_z. Here $\|\cdot\|_p$ denotes the p-norm. We use $p = 1$ and report $W_d = W_1(\mu_z, \nu_z)$ in our experiments.

B.2 Other Geodesics Transfer Path

In Section 4.5, we model vehicle trajectories on the $SE(2)$ rigid body manifold, use Lie group geodesics as probability transport paths. Now we compare this geometry with Euclidean $\mathbb{R}^n$ and other non-Euclidean manifold choices, including Riemannian (Rie, [60]), hyperbolic (Hyper, [61]), and Finsler geometries [62]. For each geometry, we provide a mathematical definition and describe its physical meaning and how it is instantiated for vehicle trajectory data.

Definition B.1 (Riemannian manifolds) *A* Riemannian manifold *is a pair $(\mathcal{M}, g)$ where $\mathcal{M}$ is a smooth manifold and g is a Riemannian metric, i.e., a smooth assignment to each point $m \in \mathcal{M}$ of an inner product*

$$g_m : T_m\mathcal{M} \times T_m\mathcal{M} \to \mathbb{R}$$

such that for any smooth vector fields X, Y the function $m \mapsto g_m(X_m, Y_m)$ is smooth. The metric g induces norms $\|v\|_m = \sqrt{g_m(v, v)}$ on $T_m\mathcal{M}$ and hence geodesic length, distance, angle and curvature on $\mathcal{M}$.

Physical meaning. This geometry equips the state space with an anisotropic cost metric (e.g., penalizing large curvature, jerk or speed) that reflects the energetic or comfort cost of different motions.

Trajectory generation meaning. The model connects latent noise and trajectories along minimum effort curves, which biases generation toward dynamically smooth and energy efficient vehicle motions.

Definition B.2 (Hyperbolic manifolds) *A* hyperbolic manifold *is a Riemannian manifold $(\mathcal{M}, g)$ that is complete and has constant sectional curvature $K_g \equiv -1$. Equivalently, an n-dimensional hyperbolic manifold is a complete Riemannian space form of constant curvature -1, any simply connected such manifold is isometric to the standard hyperbolic space $\mathbb{H}^n$.*

Physical meaning. This geometry provides a natural structure for hierarchical and tree-like relationships, which represent multi scale traffic organization such as route choices, maneuver hierarchies (go straight/turn/U-turn) and interaction graphs among vehicles.

Trajectory generation meaning. The model interpolates between maneuvers along hierarchy aware shortest paths, which helps generate coherent behaviors while preserving the global structure of the maneuver tree.

Definition B.3 (Finsler manifolds) *A* Finsler manifold *is a pair* $(\mathcal{M}, F)$ *where* $\mathcal{M}$ *is a smooth manifold and* $F : \mathcal{M} \to [0, \infty)$ *is a Finsler metric satisfying: (i) Regularity:* F *is smooth on* $\mathcal{M} \setminus \{0\}$*; (ii) Positive homogeneity:* $F(m, \lambda n) = \lambda F(m, n)$ *for all* $m, n \in \mathcal{M}$ *and* $\lambda > 0$ *(positive 1-homogeneity) and (iii) Strong convexity: the fundamental tensor*

$$g_{ij}(m, n) = \frac{1}{2} \frac{\partial^2}{\partial n^i \partial n^i} \left[F^2(m, n)\right]$$

is positive definite on $T_m\mathcal{M}$ *for every* $(m, n) \in \mathcal{M} \setminus \{0\}$*. When* $F(m, n) = \sqrt{g_m(n, n)}$ *for a Riemannian metric* g*, one recovers the Riemannian case as a special instance of Finsler geometry.*

Physical meaning. This geometry allows the "metric" to depend on both position and direction, which naturally captures direction dependent driving costs such as uphill vs. downhill, lane keeping vs. lane changing or forward vs. reverse.

Trajectory generation meaning. This geometry lets the model encode asymmetric motion preferences (e.g., penalizing sharp lane changes more than gentle cruising), so rare behaviors remain consistent with direction-dependent physical and traffic constraints.

Then we compare the $SE(2)$ geometry with Euclidean and other non-Euclidean geodesic (Geo) on their manifolds in Table 4. It shows that our $SE(2)$ geodesic attains the best results on $5/6$ metrics (83.33%). Compared with the strongest other baseline per metric, $SE(2)$ reduces mADE/mFDE by $0.82\%/0.82\%$, improves minFDE by 17.32%, and further improves SCR and HD by 2.86% and 3.69%, respectively, while being worse on minADE by 2.56%.

Table 4: Comparison of different geometry interpolation paths. We evaluate Euclidean ($\mathbb{R}^n$), Riemannian (Rie), hyperboloid (Hyper), Finsler and ours $SE(2)$. The **best** and second-best results are highlighted.

Geo Path	mADE	mFDE	minADE	minFDE	SCR	HD
$\mathbb{R}^n$	1.091	2.296	0.091	0.248	0.126	1.085
Rie	1.255	2.194	0.085	0.211	0.072	1.142
Hyper	1.296	3.223	**0.078**	0.127	0.070	1.427
Finsler	1.437	3.676	0.122	0.521	0.113	1.344
$SE(2)$	**1.082**	**2.176**	0.080	**0.105**	**0.068**	**1.045**

In this case, the hyperboloid baseline shows selective benefits (e.g., it can be competitive on safety standard) but does not translate those advantages into consistently better mean accuracy or distributional fidelity. This is in line with the typical use of hyperbolic geometry: negative curvature is particularly effective for representing graph structure with low embedding distortion, yet it can introduce a global distance geometry that is not naturally matched to approximately Euclidean road space transport. By contrast, generic Riemannian and Finsler variants may yield improvements on a small subset of criteria, but their performance can be sensitive to how curvature is instantiated, an issue widely noted in geodesic methods where the metric choice is often problem dependent and can affect robustness.

In contrast, we attribute the advantage of $SE(2)$ to its alignment with the physical structure of planar rigid body motion. Specifically, $SE(2)$ couples translation and heading through group composition, so geodesic transport preserves consistent pose evolution and avoids distortions unrelated to vehicle kinematics.

B.3 Traffic Rule Compliance

We further evaluate whether the generated trajectories comply with basic traffic rules. We extract the position coordinates of each generated trajectory, denote as $p_t = (x_t, y_t)$. We compare each generated trajectory with the lane centerline polylines provided by the map. For computation, each centerline polyline is decomposed into a set of directed line segments. For example, the m-th lane segment l_m is consist of the start point s_m and end point e_m, which of them have 2-dimension coordinates (x_m^*, y_m^*). Its midpoint and direction are computed as

$$c_m = (s_m + e_m)/2 \qquad \phi_m = \operatorname{atan2}\left(y_m^e - y_m^s, x_m^e - x_m^s\right).$$

For each trajectory point p_t, we assign it to the nearest lane centerline segment distance to the segment midpoint 1 and the corresponding centerline deviation:

$$\widetilde{m_t} = \arg\min_m \parallel p_t - c_m \parallel_2, \quad d_t = \parallel p_t - \widetilde{c_m} \parallel_2,$$

where $\widetilde{c_m}$ is the midpoint of the nearest lane segment. Thus, d_t measures the lateral proximity of the generated trajectory point to the closest lane centerline segment.

Centerline deviation A trajectory is considered to deviate from the lane centerline if its average distance to the nearest lane centerline exceeds a predefined threshold η_c:

$$\frac{1}{T}\sum_{t=1}^{T} d_t > \eta_c.$$

In our implementation, η_c =1.5m. This metric measures whether the generated vehicle trajectory stays sufficiently close to the lane centerline.

Lane departure A trajectory is considered to leave its lane if its average centerline distance exceeds half of the lane width:

$$\frac{1}{T}\sum_{t=1}^{T} d_t > \frac{w}{2},$$

where w denotes the lane width. We set w=3.5m following a standard lane width assumption, so the lane departure threshold is 1.75m. This criterion checks whether the generated trajectory remains inside the corresponding lane.

Wrong way driving To detect wrong way motion, we compare the instantaneous trajectory heading with the direction of the nearest lane segment. The motion direction between two consecutive points is

$$\psi_t = \operatorname{atan2}\left(y_{t+1} - y_t, x_{t+1} - x_t\right),$$

where near stationary steps with $\|p_{t+1} - p_t\|_{l^2} \leq \varepsilon$ are ignored. The wrapped angular difference is computed as

$$\Delta\psi_t = \min\left(|\psi_t - \phi_{\widetilde{m_t}}|, 2\pi - |\psi_t - \phi_{\widetilde{m_t}}|\right).$$

A trajectory is marked as wrong way driving if the mean angular deviation exceeds η_ψ:

$$\frac{1}{|\mathcal{V}|}\sum_{t\in\mathcal{V}} \Delta\psi_t > \eta_\psi,$$

where $\mathcal{V}$ denotes the valid moving steps. In our implementation, $\eta_\psi = \pi/3$. This metric identifies trajectories whose motion direction is inconsistent with the lane direction.

Table 5 reports the ratio of traffic rule violations among generated trajectories on the labeled Waymo scenarios. Overall, SEA achieves the lowest violation rates for centerline deviation and lane departure, with relative reductions of 5.47% and 24.59% over the strongest baseline, respectively. Although InteractTraj slightly outperforms SEA on wrong way driving (14.77% vs. 15.46%), SEA shows the best overall rule-compliance profile, suggesting safer and more lane-consistent trajectory generation.

Table 5: The ratio of generation trajectories violate the rules.

Method	Centerline deviation	Lane departure	Wrong way
TrafficGen	64.30%	30.65%	18.23%
LCTGen	66.93%	27.56%	25.79%
InteractTraj	63.60%	26.11%	14.77%
SEA	60.12%	19.69%	15.46%

B.4 Noise Analysis

To evaluate robustness [63] during generation, we inject Gaussian noise into the inputs at varying corruption ratios. We set the standard deviations to $\sigma = 0.15$ for positions, $\sigma = 0.02$ for headings, and $\sigma = 0.20$ for lane lines. The results appear in Table 6.

Table 6 demonstrates robustness to corrupted observations: even under substantial Gaussian noise, performance degrades rather than collapsing, indicating the model is not brittle to exact clean inputs an important consideration since real trajectory observations are routinely contaminated by sensor noise. This behavior is consistent with denoising style generative dynamics, which explicitly learn to recover coherent samples from noisy perturbations.

Quantitatively over the three noisy settings, the most stable metric is minADE, whose variance is 8.07×10^{-5} and whose fluctuation range is only 0.022 (0.101–0.123). For the accuracy metrics, mADE and mFDE vary within $[1.873, 3.511]$ and $[2.655, 5.339]$, respectively, while minFDE remains within $[0.063, 0.189]$. For the realism and safety-related metrics, SCR stays in $[0.067, 0.103]$ and HD varies within $[1.309, 1.530]$, supporting that the generator remains reasonably stable under Gaussian input corruption.

Table 6: Robustness to Gaussian input noise. The left column lists the corruption ratio applied to vehicle trajectories, "0" denotes our original method (without added noise).

Noise	mADE	mFDE	minADE	minFDE	SCR	HD
0	1.082	2.176	0.080	0.105	0.068	1.045
10%	3.511	5.339	0.101	0.063	0.103	1.309
30%	1.920	2.729	0.123	0.147	0.067	1.481
50%	1.873	2.655	0.112	0.189	0.078	1.530

B.5 Mask Analysis

Table 7 studies the robustness to Gaussian input noise by corrupting a portion of the input trajectory, which mirrors real world intermittency caused by occlusion and tracking dropouts. Overall, the model exhibits graceful degradation: across 10-50% corruption, mADE and mFDE remain in a bounded range of [1.873, 3.511] and [2.668, 5.366], respectively, and SCR stays comparable at [0.067, 0.103] (vs. 0.068 without noise), indicating the generator does not collapse under partial inputs. Interestingly, mild noise can even be beneficial for best-sample prediction (best minFDE at 10% corruption: 0.063 vs. 0.105 without noise), consistent with the intuition from masked modeling that learning to operate on incomplete observations encourages reliance on global structure rather than brittle local details.

Table 7: The results on masked input trajectories. The ratio represents the fraction of masked trajectory history.

Mask	mADE	mFDE	minADE	minFDE	SCR	HD
0	1.082	2.175	0.080	0.105	0.068	1.045
10%	1.365	3.177	0.176	0.378	0.066	1.052
20%	1.338	3.037	0.080	0.105	0.070	1.238
30%	1.261	2.932	0.059	0.063	0.064	1.421
40%	1.243	2.863	0.069	0.063	0.066	1.342
50%	1.325	3.067	0.123	0.252	0.070	1.258

C Loss Function

The loss function used in the data space in (9) is defined as:

$$L^i_{\text{data}}(\omega) = L^i_{\text{cro}}(q_i, \omega) + L^i_{\text{GMM}}(q_i, \omega) + L^i_{\text{MSE}}(q_i, \omega),$$

where ω denotes the scenario with corresponding generation query q_i for the i-th vehicle. This formulation follows the design principle of [64], which is to learn a shared space for the target modality (traffic scenario in our case) and text embedding. Now we illustrate each component of the L^i_{con}. For clarity, we omit the vehicle index i in the following.

Cross-entropy Loss L_{cro}. We compute the cross-entropy loss between the predicted lane $\widetilde{q}$ and the ground-truth lane index l:

$$L_{\text{cro}}(q, w) = -\log \widetilde{q}(l).$$

Here, l denotes the ground-truth lane (i.e., lane-segment) index.

Log-likelihood Loss L_{GMM}. Following prior work on vehicle trajectory prediction [65, 66], we use a Gaussian Mixture Model (GMM) to model the multimodal uncertainty in trajectory generation. The model predicts the GMM parameters via an MLP, and we maximize the likelihood of the ground-truth targets under the predicted mixtures. The resulting negative log-likelihood loss is

$$\begin{aligned} L_{\text{GMM}}(q, \omega) = &-\log \text{GMM}_{\text{heading}}(\theta) - \log \text{GMM}_{\text{vel}}(v) \\ &- \log \text{GMM}_{\text{box}}(\text{b}) - \log \text{GMM}_{\text{shift}}(s_p). \end{aligned} \tag{18}$$

In (18), there is heading θ, velocity v, bounding-box size b (as in Equation (19)) and position shift s_p. Here, $\text{GMM}_*(\cdot)$ denotes the likelihood of the corresponding target under the predicted GMM.

MSE loss L_{MSE}. This loss measures the reconstruction accuracy between the generated trajectory and the ground-truth trajectory. Conditioned on the generation query q, the model outputs K candidate trajectories with associated probabilities $\widetilde{p_Y}(k)$. Given the ground truth trajectory $\widetilde{Y}$, we select the closest candidate index

$$k_1 = \arg\min_k \sum\nolimits_t \|Y_k^t - \widetilde{Y_1^t}\|_2.$$

We then define the accuracy loss as

$$L_{\text{MSE}}(q, \omega) = -\log p_{k_1} + \sum\nolimits_t \|Y_{k_1}^t - \widetilde{Y^t}\|_2.$$

The total conditional supervision term loss L_{data} sums L_{cro}, L_{GMM} and L_{MSE} over all vehicles $i = 1, \ldots, N$ in scenario ω, which follows the same aggregation strategy as the data-physics loss $L_{\text{d-p}}$.

D Evaluation Metrics

We evaluate our SEA model by following metrics: (1) distance errors: minimum Average/Final Displacement Error (minADE/minFDE) and Average/Final Displacement Error (mADE/mFDE); (2) Hausdorff Distance (HD); and (3) Scenario Collision Rate (SCR). The smaller values mean that the generated trajectories are closer to the ground truth trajectories. Therefore the corresponding model can be considered to generate scenarios closer to the real vehicle trajectory distribution.

The (1) minimum Average/Final Displacement Error (minADE/minFDE) are defined as:

$$\text{minFDE} = \frac{1}{N_{\text{G}}} \sum_{k=1}^{N_{\text{G}}} \min_{i \in N} \left\| Y_{k,i}^{T_{\text{G}}} - \widetilde{Y_{k,i}^{T_{\text{G}}}} \right\|_2,$$

$$\text{minADE} = \frac{1}{N_{\text{G}}} \sum_{k=1}^{N_{\text{G}}} \min_{i \in N} \sum_{t=1}^{T_{\text{G}}} \left\| Y_{k,i}^{t} - \widetilde{Y_{k,i}^{t}} \right\|_2.$$

where $\widetilde{Y_*}$ and Y_* denote the ground truth (GT) and generated trajectories of vehicle i in test scenario k, respectively. N_{G} is the number of test scenarios. The mean errors Average/Final Displacement Error (mADE/mFDE) are defined analogously by replacing the minimization over with an average over i. These four metrics are measure the distance between the generate trajectory and GT trajectory.

For (2) Hausdorff Distance (HD) define as:

$$d_H\left(Y, \widetilde{Y}\right) = \max\left\{ \sup_{s \in Y} \inf_{\widetilde{s} \in \widetilde{Y}} d\left(s, \widetilde{s}\right), \sup_{s \in \widetilde{Y}} \inf_{s \in Y} d\left(\widetilde{s}, s\right) \right\},$$

where d denotes the pointwise distance between a generation s and ground truth state $\widetilde{s}$. We average HD over all vehicles and test scenarios. This measures how far two trajectories of a metric space are from each other when viewed as point sets in a scenario.

The (3) Scenario Collision Rate (SCR) is defined under the threshold δ:

$$\text{SCR} = \frac{1}{N_G} \sum_{g=1}^{N_G} \frac{2}{N(N-1)} \sum_{i<j} \mathbb{1}\left[\max_{t \in T} \text{IoU}\left(\text{b}_i^t, \text{b}_j^t\right) > \delta \right], \tag{19}$$

where $\mathbb{1}$ is the indicator function for collision between two vehicles i and j. The IoU intersection over union between bounding box b_*^t.

E Limitations and Future Work

Limitations We evaluate the proposed SEA primarily using accuracy metrics consistent with the baseline methods, whereas image and video generation typically rely on perceptual quality measures such as Fréchet Inception Distance and Learned Perceptual Image Patch Similarity. From a traffic perspective, trajectory evaluation involves not only accuracy but also diversity and reactivity. Current datasets often omit fine grained vehicle attributes and traffic rules, which limits the breadth of evaluation.

Future Work We aim to design richer traffic aware metrics that more directly capture diversity and reactivity, and to incorporate additional traffic rule and vehicle attribute annotations into the datasets. The Waymo dataset covers diverse weather conditions and a range of urban and suburban environments, while nuPlan includes widely varying traffic patterns. However, these factors are not explicitly annotated at the trajectory level. In the future, we plan to incorporate weather conditions into vehicle trajectory generation, for example, by modeling how vehicles behave under typhoon and rainstorm scenarios.

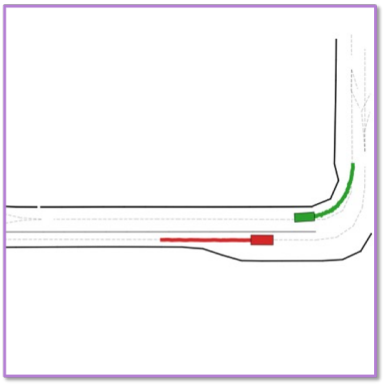
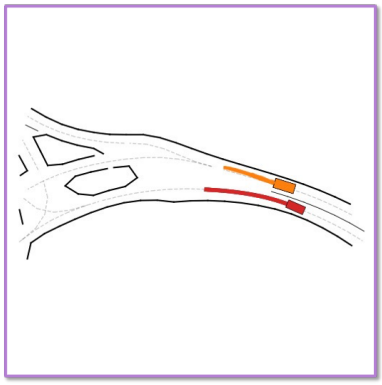
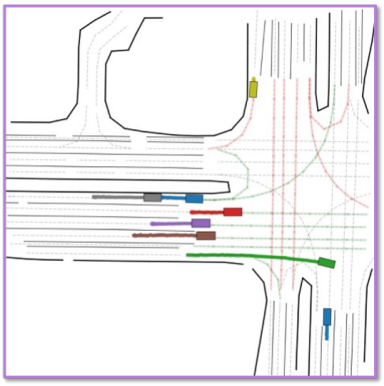
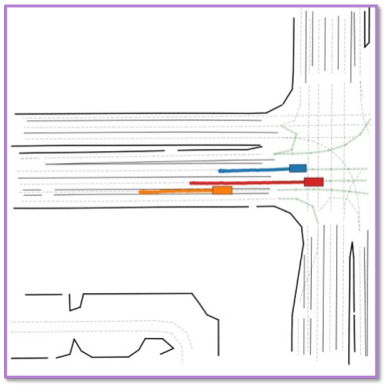
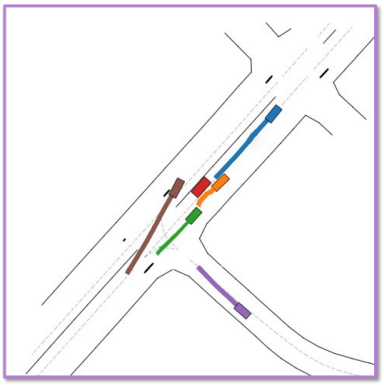
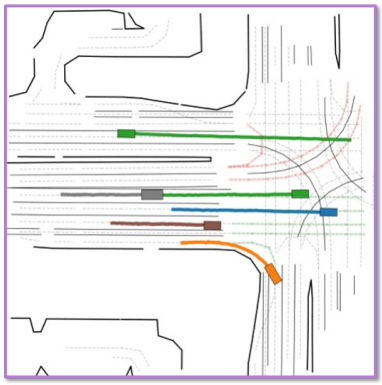
F Border Impact

From SEA we know that physical knowledge can be incorporated into generative modeling not only through loss functions or post processing constraints, but also directly through the geometry of the probability transport path. For example, molecular generation can employ $E(3)$ or $SE(3)$ equivariant denoising to generate atom types and coordinates while preserving the required spatial symmetries. Crystal generation can construct probability paths for lattice parameters, periodic atomic coordinates and element types while preserving rotational, translational, permutation and periodic symmetries. Similarly, robot motion generation can restrict intermediate samples to configuration manifolds or dynamically admissible manifolds, directly producing feasible trajectories rather than repairing invalid motions after sampling.

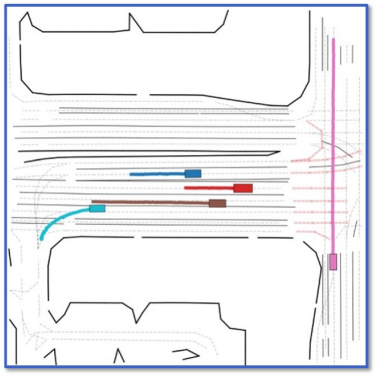
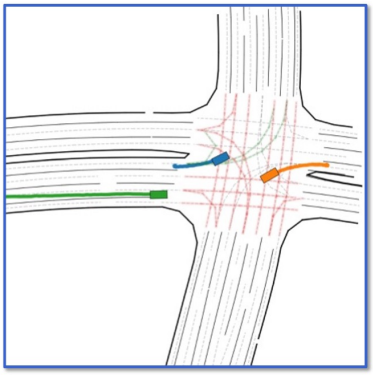
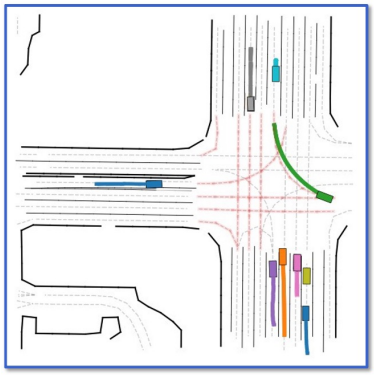
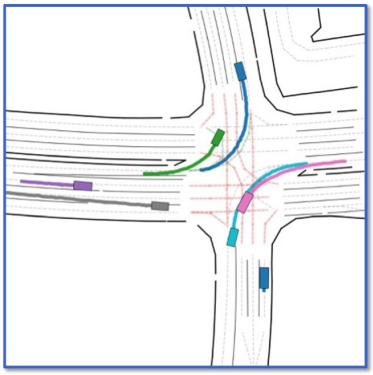
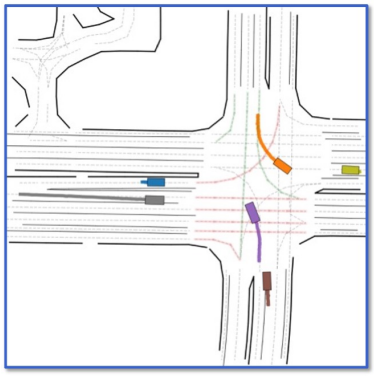
G Generation Visualization

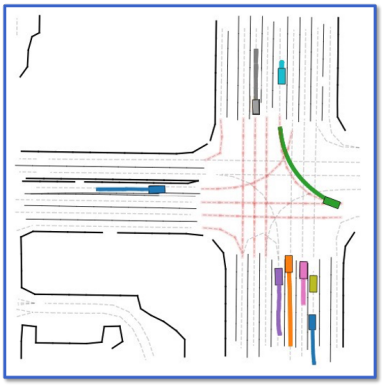
Following Section 5.2, we provide additional visualizations of trajectories generated by SEA across simple and complex traffic scenarios. These examples illustrate diverse driving behaviors, including lane changing, lane keeping, and car following, under varying traffic conditions and multi-vehicle interactions. The results further demonstrate SEA's ability to generate diverse and physically plausible trajectories while preserving the geometric consistency of vehicle motion across different scenarios.

Simple Scenario

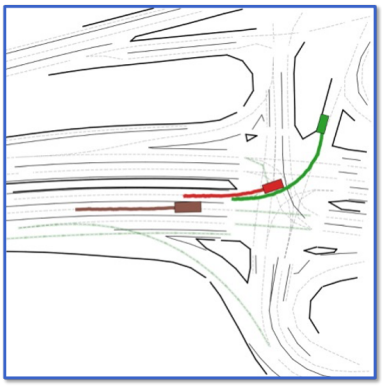







Complex Scenario

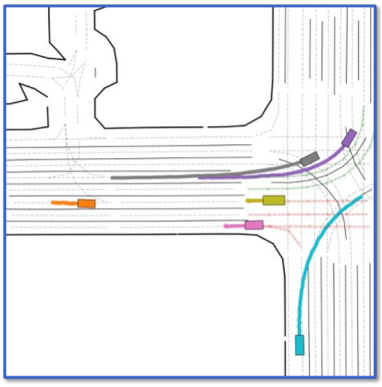






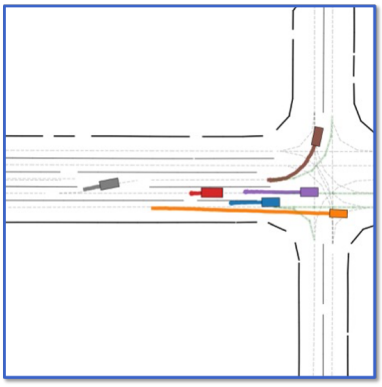



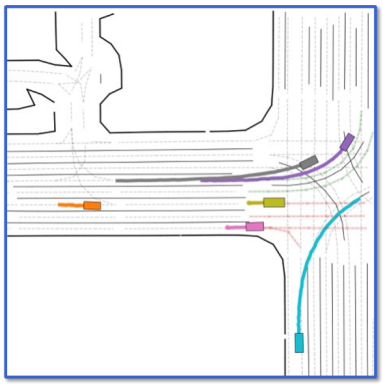
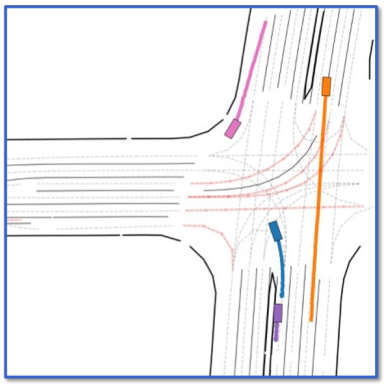
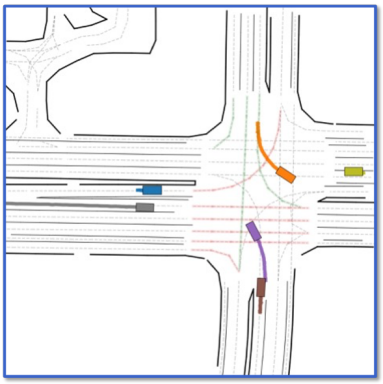
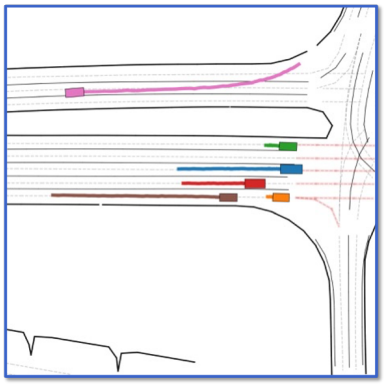


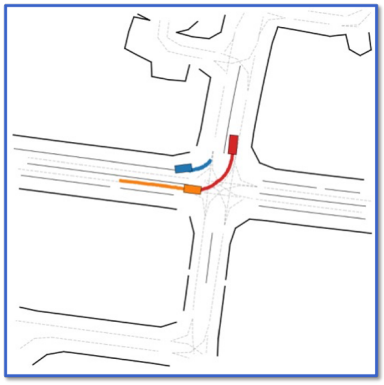


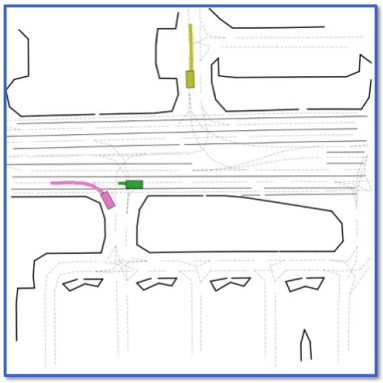


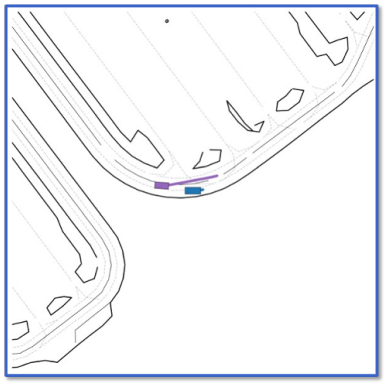




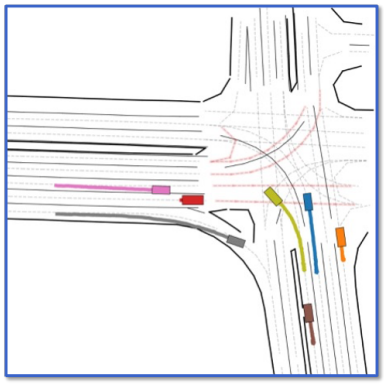





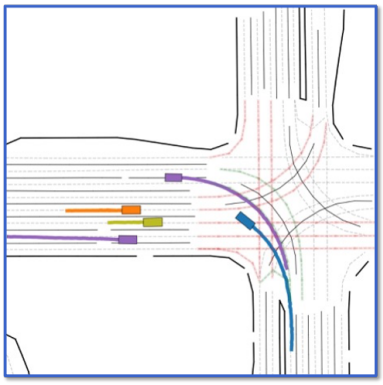


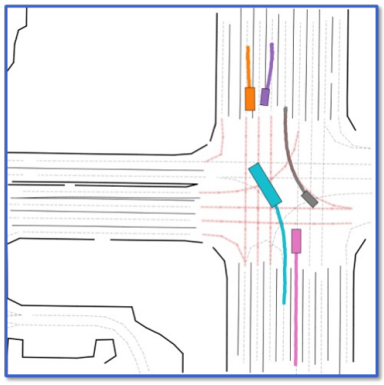


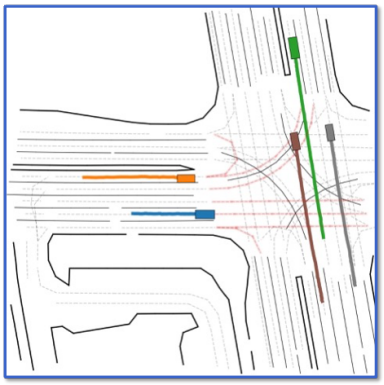
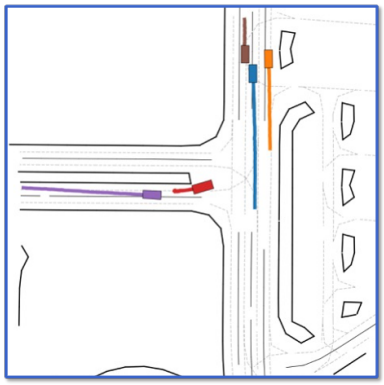


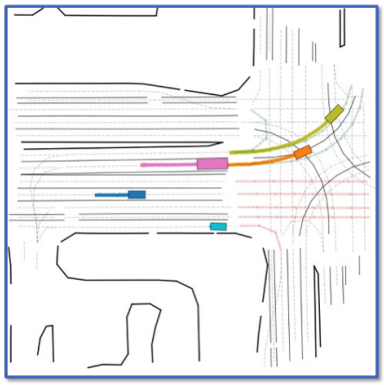
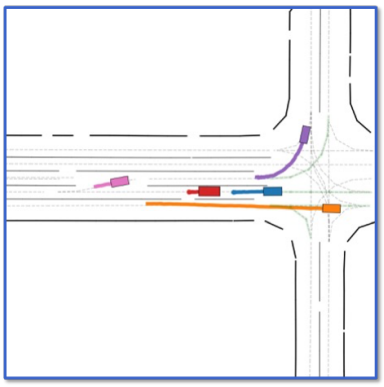









NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are supported by Section 2 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss it in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide related theory in Section 4.5 and Appendix A.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 5

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide it in supplemental material named code.zip.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A].

   Justification: This line of experimentation represents a distinct research direction from ours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See the experiment part and supplemental material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, our research methodology has been conducted in full accordance with the ethical guidelines established by NeurIPS.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss it in Section and 6 and Appendix E.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Null.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all the necessary papers, and their licenses will be made public along with our open-source code and datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not related.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Not related.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Null.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.