Towards Generalizable and Interpretable Motion Prediction: A Deep Variational Bayes Approach

Juanwu Lu
Purdue University
juanwu@purdue.edu

Wei Zhan
UC Berkeley
wzhan@berkeley.edu

Masayoshi Tomizuka
UC Berkeley
tomizuka@berkeley.edu

Yeping Hu
LLNL
yeping_hu@berkeley.edu

Abstract

Estimating the potential behavior of the surrounding human-driven vehicles is crucial for the safety of autonomous vehicles in a mixed traffic flow. Recent state-of-the-art achieved accurate prediction using deep neural networks. However, these end-to-end models are usually black boxes with weak interpretability and generalizability. This paper proposes the Goal-based Neural Variational Agent (*GNeVA*), an interpretable generative model for motion prediction with robust generalizability to out-of-distribution cases. For interpretability, the model achieves target-driven motion prediction by estimating the spatial distribution of long-term destinations with a variational mixture of Gaussians. We identify a causal structure among maps and agents' histories and derive a variational posterior to enhance generalizability. Experiments on motion prediction datasets validate that the fitted model can be interpretable and generalizable and can achieve comparable performance to state-of-the-art results.

1 INTRODUCTION

With the rapid commercialization of autonomous vehicles (AVs), one can foresee increasing AV penetration rates and interactions among AVs and their surrounding human-driven vehicles (HDVs). To safely navigate through mixed and congested traffic, AVs must evaluate and predict future collision risks. A related task is motion prediction, which refers to forecasting the future trajectories of surrounding objects. However, motion prediction can be challenging due to uncertain and multi-modal driver behaviors: the exact history trajectories can point to several possible future paths, conditioned on plausible destinations, road geometry, social interactions, and maneuver differences, as illustrated in Figure 1.

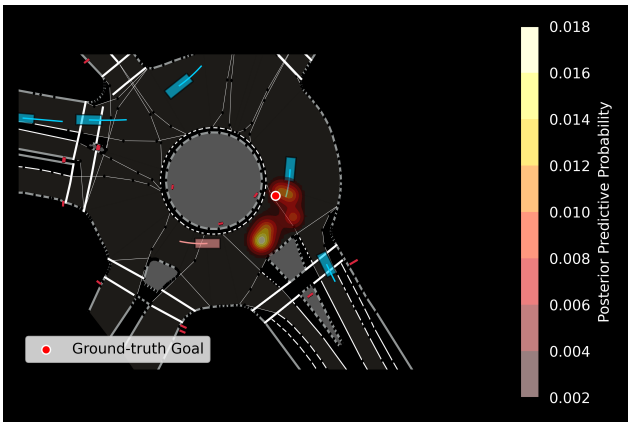


Figure 1: Example case illustrating the multi-modal distribution of long-term goal. The target vehicle can maintain its current cruising speed or accelerate inside the roundabout, leading to a spatial distribution with multiple modes.

Recently, learning-based methods have achieved outstanding prediction accuracy and gained emerging popularity. Nevertheless, most of these models assume that training and test data follow similar statistics and fit their parameters as point estimations derived from maximum likelihood. As a result, they can make overconfident predictions under distributional shifts due to changing road geometry or traffic conditions (Lakshminarayanan et al., 2017; Filos et al., 2020; Bahari et al., 2022). Although meticulously adjusting the architecture, constantly enlarging the number of parameters, or training the model with a sufficiently large dataset that covers a diversified situation can help mitigate the performance degradation facing out-of-distribution (OOD) data (Wang et al., 2022b), the

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

time and computing resources required for data collection and parameter tuning can be intractable. Meanwhile, interpretability is necessary for the robustness and safety of deploying motion prediction algorithms in real-world cases. While end-to-end prediction models can deliver high accuracy, they are primarily black boxes with extremely limited interpretability.

In this work, we address these limitations and propose the Goal-based Neural Variational Agent (*GNeVA*) model. The model follows a target-driven trajectory prediction setting (Zhao et al., 2021; Gu et al., 2021), where the motion prediction consists of two subtasks: predicting a continuous spatial distribution over plausible trajectory endpoints (i.e., *goals*) on the map and completing the intermediate trajectory from the current location to the goals. The source code of our model is open-sourced at `https://github.com/juanwulu/gneva`. The main contribution of this paper is summarized as follows:

- We identify and implement a causal structure where the surrounding physical context features determine the expected locations of goals, and the uncertainty is only sourced from dynamic future interactions.
- We propose a generative model using a variational mixture of Gaussians with learnable prior and variational posterior to model the spatial distribution of goals. The variational posterior is derived from the causal structure.
- We comprehensively evaluate GNeVA on the Argoverse Motion Forecasting (Chang et al., 2019) and the INTERACTION dataset (Zhan et al., 2019), where the model is shown to yield comparable performance to the state-of-the-art motion prediction models. Crucially, the model maintains its performance under cross-scenario and cross-dataset tests, which indicates a promising generalizability. We further showcase predicted intention distributions qualitatively.

2 RELATED WORKS

2.1 Generalizability and Interpretability in Motion Prediction Models

While a few existing works implicitly consider generalization ability and interpretability in their model design (Tran et al., 2021; Gu et al., 2021; Zeng et al., 2021; Gilles et al., 2022), they did not thoroughly investigate these properties or clearly state what kind of design or model structure helped to address them. Other works tried to address these issues explicitly in motion prediction tasks. For example, Hu et al. (2022b) and Wang et al. (2022a) proposed using domain-invariant semantic representation to minimize the joint discrepancy across traffic domains in a shared latent feature space. Hu et al. (2022a) utilized a causal structure to learn temporal invariant representations. These methods, however, still have limited interpretability on the latent correlations between output predictions and input features.

One way to incorporate interpretability is to depict the distribution of plausible future trajectories using a generative model. The classic approach employs flexible implicit distributions of trajectories from which the predictions can be drawn, such as conditional variational autoencoder (Lee et al., 2017b) and generative adversarial networks (Chai et al., 2020). Despite their competitive prediction accuracy, using latent variables for reasoning driver behavior prohibits them from being interpreted and often requires inference-time sampling or ensembling methods to evaluate the uncertainty. Previous works address the limitation by decomposing trajectory prediction into two sub-tasks, intent identification and trajectory completion, based on the assumption that the uncertainties about a sufficiently long future trajectory can be mostly captured by its destination (Dendorfer et al., 2021).

2.2 Deep Generative Model

Driven by the emerging progress in generative models (Sohn et al., 2015; Creswell et al., 2018), there has been a significant shift in the paradigm of predictive modeling. Unlike deterministic regressors, generative models produce a distribution representing potential future behaviors. Notably, deep generative models, which exploit the expressiveness of neural networks for approximating the actual underlying probability distributions, have emerged as leading-edge methods. One primary deep generative approaches that stand out in the field are (Conditional) Variational Autoencoders ((C)VAEs) (Kingma and Welling, 2013; Sohn et al., 2015), which applies variational inference through parameterizing the generative and variational family distributions with neural networks. Such methodologies have seen extensive application in predicting future vehicle trajectories within interactive scenarios (Gupta et al., 2018; Lee et al., 2017a; Kosaraju et al., 2019). However, existing (C)VAE-based models commonly draw random samples from the latent space during inference, which has no guarantee for generalization, and the latent space lacks direct interpretability. Meanwhile, they often adopt uni-modal Gaussian distributions as both generative and variational distributions, which can fail to capture the multi-modality in future trajectories.

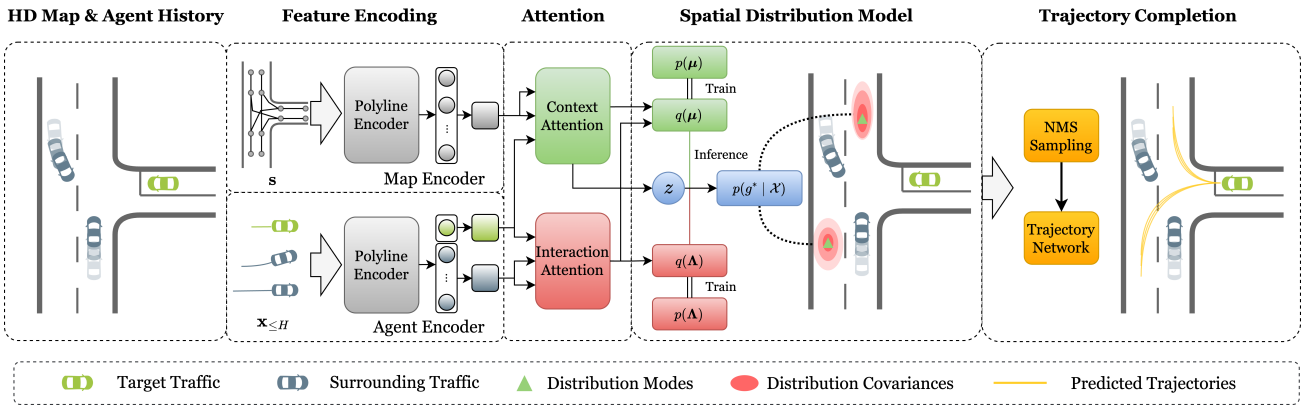


Figure 2: GNeVA model overview. The input HD map and history trajectories of all observed traffic are first encoded through polyline-based Map and Agent encoders, respectively. Encoded vector features associated with road geometry and agents' histories pass the context attention and interaction attention modules to derive posterior distribution parameters of means and precision. We then evaluate and sample goals from the posterior predictive distribution. Finally, a trajectory network completes the intermediate paths from current positions to sampled goals.

3 METHOD

This section first presents the problem statements for the motion prediction problem (Section 3.1). Then, we introduce the formulation of the proposed deep generative model (Section 3.2) with causal structure. In this model, the prior distribution consists of trainable parameters, and the posterior distribution is parameterized by neural network encoders (Section 3.3). To allow sampling from the multi-modal mixture model, we train a proxy network evaluating the mixture assignments (Section 3.4). Finally, we introduce how to sample from the posterior predictive distribution, generate intermediate trajectory(Section 3.5), and train the model (Section 3.6).

3.1 Problem Statement

In this paper, we follow the target-driven trajectory prediction problem setting and formulate the motion prediction as a two-step regression problem. The following lists out key concepts and the notations we use:

- **Environment Semantics** refers to the objects in the surroundings besides traffic participants. Typical environment semantic data include high-resolution maps (HD maps), point clouds, and traffic regulations (e.g., stop signs, traffic lights, etc.). The set of all objects' indexes is denoted as $\mathcal{S}$, and the features of the i-th object observed at time step t is denoted by $\mathbf{s}_t^{(i)},\ i \in \mathcal{S}$.
- **Traffic Participants** consists of individuals or entities interacting in the current traffic, such as pedestrians, cyclists, and pedestrians. The set of all traffic participants' indexes is denoted as $\mathcal{P}$, and the motion states of participant i observed at time step t is marked as $\mathbf{x}_t^{(i)},\ i \in \mathcal{P}$.
- **Target Participants** refers to a set of participants whose possible future locations are to be predicted. The collection of all target participants' indexes is denoted as $\mathcal{T},\ \mathcal{T} \subseteq \mathcal{P}$, and the motion states of participant i observed at time step t is similarly denoted as $\mathbf{x}_t^{(i)},\ i \in \mathcal{T}$.
- **Observation Horizon** is the number of history time steps we have observed for prediction, denoted by H.
- **Prediction Horizon** is the number of future time steps to predict, denoted by T.

Suppose we aim to predict the future trajectory of $|\mathcal{T}| = N$ target vehicles. The objective is to search for an optimal model in model space $\mathcal{F}$ such that it maximizes the joint likelihood of future trajectories conditioned on the observations given by,

$$\max_{f \in \mathcal{F}} \prod_{i=1}^{N} \prod_{t=1}^{T} p\left(\mathbf{x}_{H+t}^{(i)} \mid f(\mathbf{x}_{<H+t}^{(j)}, \mathbf{s}_{<H+t}^{(k)})\right), \quad (1)$$

where $i \in \mathcal{T}$, $j \in \mathcal{P}$, and $k \in \mathcal{S}$. However, direct estimation of the joint likelihood is challenging, and implicit estimations can limit interpretability. The target-driven trajectory prediction assumes that goals satisfy physical constraints and social interactions and capture the most uncertainty in a long-horizon prediction. We can reduce the estimation to maximize the

upper bound of the original objective

$$\max_{f' \in \mathcal{F}} \prod_{i=1}^{N} p\left(\mathbf{x}_{H+T}^{(i)} \mid f(\mathbf{x}_{\leq H}^{(j)}, \mathbf{s}_{\leq H}^{(k)})\right). \tag{2}$$

We further consider a more straightforward case of the problem: to predict the trajectory of a single agent (*i.e.*, $N = 1$) conditioned on its surrounding traffic participants and a static environment (*i.e.*, $\mathbf{s}_t^{(i)} = \mathbf{s}^{(i)}, \forall i \in \mathcal{S}, t = 1, \ldots, H+T$). To achieve this, we propose the GNeVA model (illustrated in Figure 2) to learn and output the distribution representing the likelihood of an agent's goal at the prediction horizon. A trajectory network completes the intermediate path from its current location to the final goal. A spatial distribution model built with a variational mixture of Gaussians is the key component that drives the likelihood evaluation.

3.2 Spatial Distribution Model for Goals

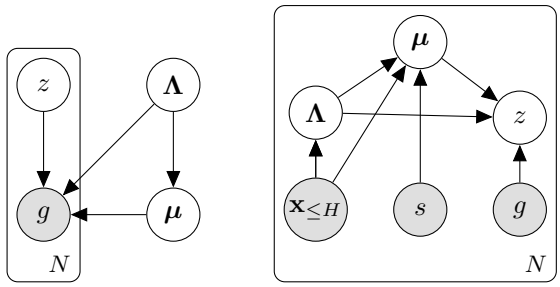
For simplicity, let $g \subset \mathbf{x}_{H+T}^{(i)} \in \mathbb{R}^2$ denote the two-dimensional location of the goal at the prediction horizon for a target participant i. Denote the collection of observed goals in the dataset by $\boldsymbol{g} = \{g_1, \ldots, g_N\}$. For each observation g_n, we can introduce a latent variable $z_n \in \{0,1\}^C$. As illustrated in Figure 3a, we consider the generating process of observed sample g_n in the form

$$z_n \sim \text{Categorical}(\pi), \tag{3}$$

$$g_n \mid z_n, \boldsymbol{\mu}, \boldsymbol{\Lambda} \sim \prod_{c=1}^{C} \mathcal{N}\left(g_n \mid \boldsymbol{\mu}_c, \boldsymbol{\Lambda}_c^{-1}\right)^{z_c}, \tag{4}$$

$$\boldsymbol{\mu}_c, \boldsymbol{\Lambda}_c \sim \mathcal{N}\left(\boldsymbol{\mu}_c \mid \eta_0, (\beta_0 \boldsymbol{\Lambda}_c)^{-1}\right) \mathcal{W}\left(\boldsymbol{\Lambda}_c \mid V_0, \nu_0\right). \tag{5}$$

where C is a tunable number of mixture components and π is the mixing coefficients. The formulation addresses two propositions related to the nature of multi-modality and uncertainty of the goals:



(a) Likelihood Family (b) Variational Family

Figure 3: Graphical model for the GNeVA showing the likelihood family (left) and variational family (right).

Proposition 1. *Goals are multi-modal samples from a mixture of diverse intention distributions, and a single observed goal in the data is a sample from one dominant intention at a specific timestamp.*

Following this proposition, we leverage a mixture indicator variable z to learn which intention mixture component has the dominant effect on the goal. In practice, $z_n \in \{0,1\}^C$ is a one-hot vector sampled from a Categorical distribution parameterized by π.

Proposition 2. *The distribution of goals reflects a joint effect of certainty and uncertainty, where certainty is associated with expectations, and the covariances quantify uncertainty.*

Following this proposition, the mixture components are bivariate Gaussian distributions parameterized by expectations $\boldsymbol{\mu}$ and precisions $\boldsymbol{\Lambda}$ (i.e., *inverse covariances*). In previous works, parameterization is accomplished by either training a neural network to output the point estimations of the mixture distribution parameters (Chai et al., 2020) or using ensemble methods to approximate the covariances (Varadarajan et al., 2022). The common issue is that the output point estimations from the neural networks are optimized using training data. For the unseen or OOD cases, the same set of parameters may no longer be effective, hinging its capability to generalize across scenarios.

We address this limitation by adopting a Bayesian mixture with conjugate priors to account for the epistemic uncertainty of the parameters. The conjugate priors can bring algebraic convenience and closed-form expression to the posterior. We consider the means for all mixture components $\boldsymbol{\mu}$ to follow a bivariate normal prior, where prior mean η_0 and parameter β_0 are trainable. The precisions $\boldsymbol{\Lambda}$ are said to follow a Wishart distribution with trainable parameters V_0 and ν_0.

However, directly learning parameters of the true posterior $p(\boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{g}, \boldsymbol{z})$ from the training dataset can not ensure the model's generalizability. The learned parameters can only express the belief of the latent parameters under the training dataset. Instead, we propose to parameterize a variational posterior distribution $q(\boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{g}, \boldsymbol{z}, s, \mathbf{x}_{\leq H})$ leveraging a *causal structure* between environment and goal.

Mean Posterior Since the expectations should reflect certainty, the variational mean posterior distribution reflects the belief of the expected goal location conditioned by observation. In real-world cases, the posterior variable η_c is strongly bound to the concept of "anchors." For example, if we observed the lane where the target vehicle was positioned was a left-turn lane (contained in feature $\mathbf{s}$). The downstream left-turn exit lane was empty (contained in feature $\mathbf{x}_{\leq H}$),

the most probable goal location somewhere on the exit lane. Meanwhile, $\boldsymbol{\beta}_c$ measures our belief in the uncertainty about this distribution, which the observations of the environment and interaction history should also deduce.

Precision Posterior Different from the mean posterior distribution, the precision posterior is conditioned only on the history traffic states $\mathbf{x}_{\leq H}$. The reason is two-folded:

- We argue that uncertainty is rooted in the ever-changing dynamics of the surroundings rather than the static components. Since the environment is considered static in our problem settings, we drop them from the condition to avoid spurious correlations.
- Meanwhile, uncertainty quantification requires anticipation of the plausible future. To support the anticipation, we incorporate the full history feature as the function input.

Therefore, the joint variational posterior of the latent random variables is in the form

$$\prod_{n=1}^{N}\prod_{c=1}^{C}\mathcal{N}(\boldsymbol{\mu}_c \mid \boldsymbol{\Lambda}_c, \mathbf{x}_{\leq H}, \mathbf{s})\mathcal{W}(\boldsymbol{\Lambda}_c \mid \mathbf{s})q(z_{nc}). \tag{6}$$

where the z-posterior is approximated by

$$q(z_{nc}) = \frac{\pi_i \mathbb{E}_{q(\boldsymbol{\mu}_c,\boldsymbol{\Lambda}_c)}\left[\log p(g_n \mid \boldsymbol{\mu}_c, \boldsymbol{\Lambda}_c)\right]}{\sum_{i=1}^{C} \pi_i \mathbb{E}_{q(\boldsymbol{\mu}_i,\boldsymbol{\Lambda}_i)}\left[\log p(g_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)\right]}. \tag{7}$$

Derivation of equation 6 and 7 are in the supplementary material. By constructing our model this way, we decouple the goal observation with posterior means and precisions (see Figure 3b), which benefits the performance in three ways. First, it prevents the neural network for estimating posterior parameters from overfitting the distribution of goals in the training dataset, which helps guarantee the generalization performance. Meanwhile, since means and precisions are independent of the locations, we can train a proxy z-posterior network (see Section 3.4) to evaluate mixture assignment in unseen scenarios. Finally, we can derive a closed-form objective for training our model (see Section 3.6).

3.3 Feature Encoding and Attention Module

Feature Encoding Following the approaches in existing works (Gao et al., 2020; Zhao et al., 2021), we represent the observation of a traffic scenario as a collection of polylines consisting of map-related polylines and agent history trajectories. Each polyline is broken into a collection of vectors containing origin and destination coordinates, other attributes such as heading, velocity, and dimensions for trajectories, and polyline types for map-related polylines. We encode map polylines and agents' history trajectories using two separate polyline encoders, resulting in three features: map features $\mathbf{m}$, target traffic participant's history feature $\mathbf{e}$, and surrounding participants' history features, $\mathbf{o}$.

Attention Module To capture global interaction and avoid spurious correlations, we use two separate attention modules: the *context attention* and the *interaction attention* module to *map-target* and *surrounding-target* interactions, using target history feature $\mathbf{e}$ as query, and $\mathbf{m}$ and $\mathbf{o}$ as key and value, respectively. Both modules contain a stack of multi-head attention encoders (Vaswani et al., 2017) followed by MLPs. The context feature is the sum of the output from the context and interaction attention module to jointly represent $(\mathbf{x}_{\leq H}, \mathbf{s})$, whereas the interaction feature is the output from the interaction attention module. Details about the attention modules are listed in the supplementary.

3.4 Proxy z-posterior Network

The Proxy z-posterior Network (z-proxy) aims to parameterize the mixture assignment over the components and serve as a proxy for the learned z-posterior $q(\boldsymbol{z} \mid \boldsymbol{g}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$. We model the z-proxy with an MLP conditioned on the context feature $\tilde{p}(\boldsymbol{z}|\mathbf{x}_{\leq H}, \mathbf{s})$. To be noticed, z-posterior and z-proxy are mathematically different, where the former associates a location in space with a mixture, while the latter estimates the assignment conditioned on the current environment and vehicle movement. Similar implementations exist in previous literature (Dendorfer et al., 2021) and have been shown to help maintain performance in unseen and OOD cases.

3.5 Sampling and Trajectory Completion

Recall that each mixture component of the spatial distribution is a multivariate Gaussian with Normal-inverse-Wishart conjugate prior. The posterior predictive distribution for a newly observed location g' is in the form

$$p(g^*) \approx \sum_{c=1}^{C} \tilde{p}(\boldsymbol{z}|\mathbf{x}'_{\leq H}, \mathbf{s}')\tau_{\nu_c - 1}\left(\eta_c, \frac{\beta_c + 1}{\beta_c(\nu_c - 1)}V_c^{-1}\right), \tag{8}$$

where $\tau(\cdot)$ is the multivariate student-t distribution, η_c and β_c are the mean and scaling factor of the mean posterior, V_c and ν_c are the scale matrix and degree of freedom of the precision posterior. We weigh

Table 1: Performance Results on Argoverse validation set.

	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$	$\mathbf{MR}_6$
TPCN (Ye et al., 2021)	<u>0.73</u>	1.15	0.11
mmTrans (Huang et al., 2022)	**0.71**	1.15	0.11
LaneGCN (Liang et al., 2020)	**0.71**	<u>1.08</u>	-
GNeVA (Ours)	0.78	**1.06**	**0.10**

Algorithm 1 Goal Sampling

Require: List of candidate locations G with associated probabilities.
Require: Candidate buffer radius r
Require: Intersection-over-Union (IoU) Threshold γ
Ensure: List of selected candidates D
Sort candidates by probabilities $p(g), g \in G$ in descending order
Initialize an empty list D
while $G \neq \emptyset$ **do**
  Take the most probable candidate g^* from G
  Add g to D
  Create circle c centered at g of radius r
  for each candidate g' in G **do**
    Create circle c' centered at g' of radius r
    if $\text{IoU}(c, c') > \gamma$ **then**
      Remove g' from G
    end if
  end for
  Remove g from G
end while
return D

each component probability using output from the z-proxy network. The goals are sampled using the Non-maximum Suppression (NMS), as described in Algorithm 1. After we obtain the list of goal candidates, we use a trajectory network modeled by a cascade of MLPs to predict and complete the intermediate trajectory given the goal candidates and the context feature.

3.6 Model Training

We train our the generative model by maximizing the log-evidence lower bound (ELBO), given by

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = &\sum_{c=1}^{C} q(\boldsymbol{z}) \mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Lambda})} \left[\log p(g|\boldsymbol{\mu},\boldsymbol{\Lambda},z)\right] \\ &- D_{KL}\left[q(\boldsymbol{\mu},\boldsymbol{\Lambda})|p(\boldsymbol{\mu},\boldsymbol{\Lambda})\right] \\ &- \mathbb{E}_{q(\boldsymbol{\Lambda})q(\boldsymbol{\mu})} D_{KL}\left(q(z|\boldsymbol{\mu},\boldsymbol{\Lambda})|p(\boldsymbol{z})\right),\end{aligned} \tag{9}$$

where $D_{KL}(\cdot)$ is the Kullback–Leibler divergence. Please refer to the supplementary material for the closed-form ELBO loss. Meanwhile, we train the z-proxy network by minimizing the cross-entropy

$$\mathcal{L}_z = -\mathbb{E}_{q(\boldsymbol{z})}\left[\log \tilde{p}(\boldsymbol{z})\right]. \tag{10}$$

The trajectory network is trained to minimize the Huber Loss between the predicted trajectory and ground-truth trajectory given the ground-truth goal. In practice, we train the spatial distribution model and the trajectory network separately.

4 EXPERIMENTS

We empirically evaluate the GNeVA on the popular motion prediction datasets following the experiment settings introduced in section 4.1. Section 4.2 presents performance comparison and analysis with state-of-the-art motion prediction methods. In section 4.3, we evaluate the model's generalization ability. Finally, section 4.4 presents visualizations of predictive distribution in in-distribution and out-of-distribution cases with qualitative analysis of the model's interpretability.

4.1 Experiment Settings

4.1.1 Dataset

We evaluate our model in single-agent motion prediction settings on the Argoverse Motion Forecasting dataset (Chang et al., 2019) and the INTERACTION dataset (Zhan et al., 2019). The Argoverse consists of roughly $323,557$ scenarios, the task of which is to predict future three-second motions given the HD map and the two-second history of the agent. The INTERACTION dataset consists of around $398,409$ cases, and the objective is to predict the future trajectory of target vehicles in three consecutive three seconds given one-second history observations. Additionally, the INTERACTION dataset provides manual labels of different road geometry, including highway merging, intersections, and roundabouts, which is convenient for evaluating OOD performance.

4.1.2 Metrics

We use standard motion prediction metrics, including the minimum average displacement error (mADE_k),

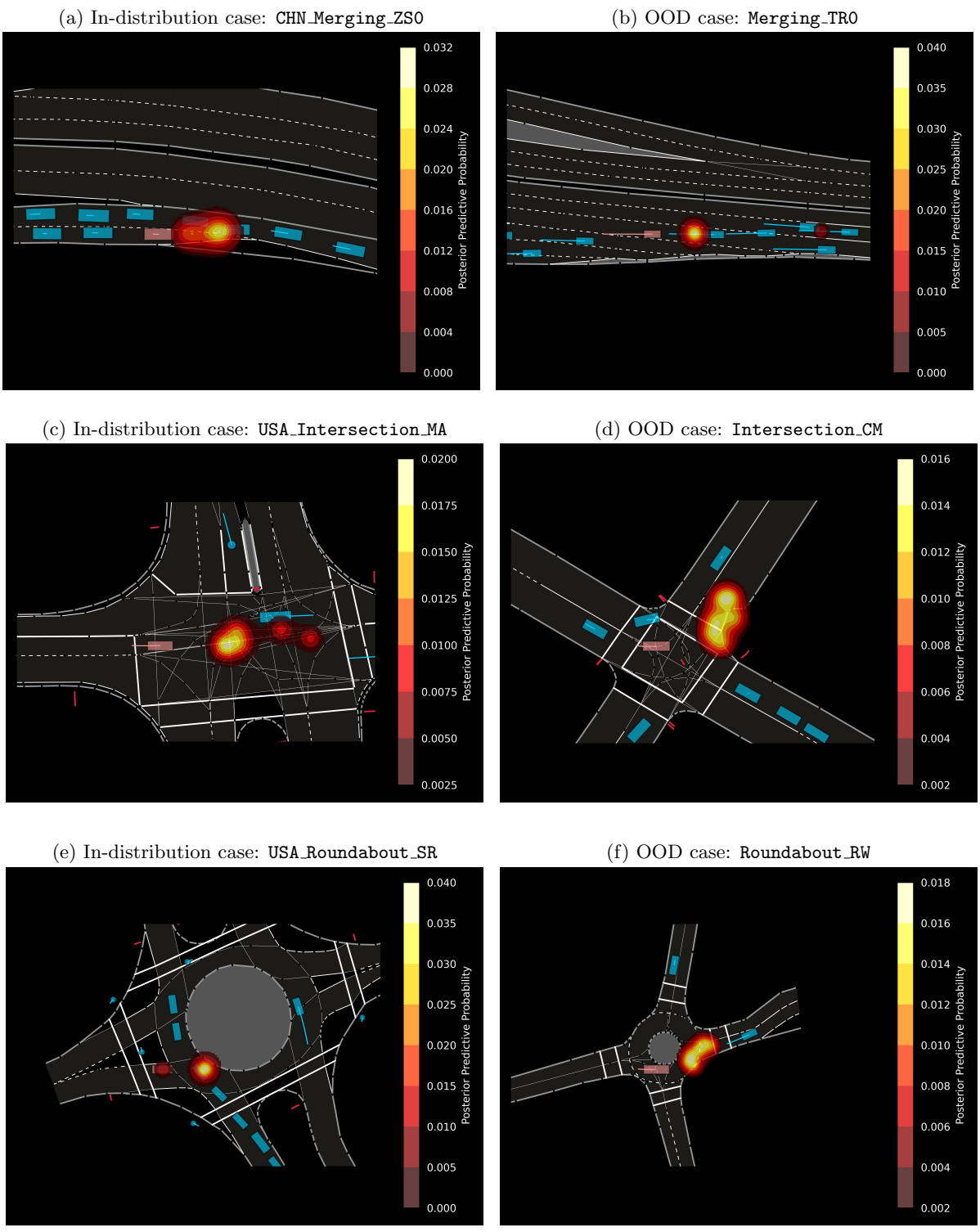


Figure 4: Visualization of posterior predictive goal distributions under selected in-distribution and out-of-distribution cases. All cases are selected from the INTERACTION test dataset.

minimum final displacement error (mFDE_k), and the miss rate (MR_k). The mADE_k and mFDE_k are calculated between top-k trajectory predictions and the ground truth, given as

$$\text{mADE}_k = \min_k \frac{1}{T} \sum_{t=H+1}^{H+T} \sqrt{(\hat{x}_t^k - x_t)^2 + (\hat{y}_t^k - y_t)^2}, \tag{11}$$

$$\text{mFDE}_k = \min_k \sqrt{(\hat{x}^k_{H+T} - x^k_{H+T}) + (\hat{y}^k_{H+T} - y^k_{H+T})}, \tag{12}$$

where T is the number of time steps needed to predict the future. MR_k is only calculated following the definition given by the Argoverse dataset. It is the ratio of cases where minFDE_k is higher than 2 meters.

4.2 Benchmark Results

Table 2: Results on INTERACTION validation set.

	mADE$_6$	**mFDE$_6$**
DESIRE (Lee et al., 2017b)	0.32	0.88
MultiPath (Chai et al., 2020)	0.30	0.99
TNT (Zhao et al., 2021)	**0.21**	<u>0.67</u>
GNeVA (Ours)	<u>0.25</u>	**0.64**

We compare the proposed GNeVA motion prediction model with the state-of-the-art method on the validation split of both datasets. As shown in Table 1, our model achieves state-of-the-art performance measured by minFDE_6 and miss rate on the Argoverse dataset, with a slightly higher (i.e., ≈ 0.08 meters) minADE_6 compared to the others. Table 2 shows the performance result evaluated on the INTERACTION validation set. Our model has the best mFDE_6 and the second-place mADE_6 performance among the four models compared. The GNeVA model can achieve performance comparable to state-of-the-art motion prediction models.

4.3 Generalization Evaluation

4.3.1 Cross-scenario Generalizability

The model's generalization ability can be evaluated directly through a cross-scenario test (Lu et al., 2022), that is, to mimic the distribution shift in a real scenario by training and testing the model under different traffic conditions. The drop in performance can reflect the existence of generalization bottlenecks. Table 3 shows the cross-scenario test results. From the results, we find that the model that is trained solely using data collected from the intersections can achieve similar performance under out-of-domain cases (roundabout), with about 8% in mADE_6 and 10% in mFDE_6. Meanwhile, the model trained solely using data collected from the roundabout can also maintain its performance in OOD cases, with only a 5%

4.3.2 Cross-dataset Generalizaility

Evaluating the generalization performance under different datasets is more challenging than cross-scenario evaluation since distribution shifts exist in underlying traffic conditions and noise due to different sensors. The evaluation results are listed in Table 4. We observed a 0.14 meters increase in mADE_6, a 0.28 meters increase in mFDE_6, and a 5% increase in MR_6 when trained on INTERACTION and evaluated on the Argoverse. On the other hand, the mADE_6 and mFDE_6 are still below 0.5 meters and 1.0 meters when trained on the Argoverse and evaluated on the INTERACTION. Therefore, the performance degradation when the model is trained on the INTERACTION dataset and evaluated on the Argoverse dataset is higher than the opposite. This fits our expectation that the Argoverse dataset has a higher diversity and is collected under a wider urban area than the INTERACTION dataset. Results from this experiment show that our model has the potential generalizability to apply to unseen datasets directly but can face challenges when trained on a significantly less diverse training set. We argue that such a limitation can potentially be rooted in the fact that the neural network used to parameterize the posterior fails to learn a sufficiently generalizable mapping from observed data to posterior parameters.

4.4 Qualitative Analysis

Since the GNeVA model aims to improve interpretability by directly modeling the distribution of goals, it is essential to validate if the predicted distribution fits the properties of driver intention. In Figure 4, we select and visualize six cases from the INTERACTION test dataset, where three of them are in-distribution cases with seen road geometry and similar driver behavior in the training dataset, whereas the other three cases are collected from locations unseen in the training dataset.

In the first row, we compare the cases where the target vehicle is merging into the mainline of a highway section. Plots indicate that our model can successfully distinguish the lanes in the ramp and the mainline and identify the staying-within-ramp intention in the in-distribution case (see Figure 4a), as well as the merging-left-into-mainline intention in the out-of-distribution case (see Figure 4b). More importantly, we observe two separate distribution modes in two adjacent lanes. This indicates that the model successfully anticipates two lane-changing intentions: finish the current lane-change maneuver, stay in the lane, or accelerate and change to an inner adjacent lane for higher speed.

The middle row shows results on two different intersections. We compare two similar cases with potential left-turn target vehicles but at different locations and with different current headings. In the in-distribution case (see Figure 4c), the distribution has three clear

Table 3: Model Performance under Cross-scenario Tests

	Train Scenario					
	Intersection		Roundabout		Full Dataset	
Validate Scenario	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$
Intersection	0.56	1.41	0.56	1.39	0.31	0.73
Roundabout	0.61	1.56	0.44	1.08	0.32	0.76

Table 4: Cross Dataset Evaluation Results.

Dataset	Argoverse (validate)			INTERACTION (validate)	
	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$	$\mathbf{MR}_6$	$\mathbf{mADE}_6$	$\mathbf{mFDE}_6$
Argoverse (train)	0.78	1.06	0.10	0.37	0.91
INTERACTION (train)	0.92	1.34	0.15	0.25	0.64

clusters: the right-most one is associated with going straight, the middle mode potentially allows the target vehicle to turn left and enter the outer lane of the downstream exit, and the left-most cluster shows a diverse intention, including going straight or turning left and entering the inner lane of the exit. In the OOD case (see Figure 4d), the multi-modal distribution concentrates along the only left-turn exit lane since the target vehicle has a clear left-turn heading. The two examples showcase the power of GNeVA to identify multiple plausible intentions, which address the multi-modal property of driver intention.

Visualizations of the two cases in the bottom row demonstrate predictive distributions on two different roundabouts. As shown in Figure 4e, the model can successfully anticipate two plausible intentions: staying behind the stop line and yield (i.e., *the left mode*) and accelerating to enter the roundabout (i.e., *the right mode*). In Figure 4f, we choose the OOD case where there is a small roundabout significantly different from the common large-scale roundabouts in the training data. Despite the model predicting a predictive distribution that fits the constraints of the roundabout geometry, we see that it only identifies the plausible path toward the nearest exit, which shows that the model can be further improved in the future to tackle cases with road geometry that is rare in the training data. Further experiments evaluating the model's sensitivity to road geometry can be found in the supplementary.

5 CONCLUSION

In this paper, we propose the Goal-based Neural Variational Agent (GNeVA), a deep variational Bayes model that evaluates the spatial distribution of the long-term goals for drivers. We postulate the propositions about the multi-modality and a causal structure associated with the goal. Following these propositions, we design the model using a variational mixture of Gaussian distributions with posterior parameterized by the neural network. In the experiment, our model achieved comparative performance as the state-of-the-art models, showcased a promising generalization ability to the unseen cases, and demonstrated the ability to generate predictive distributions that address the constraints from road geometry and reflect the multi-modal property of driver behavior.

Limitations The GNeVA model is inherently a single-agent trajectory prediction model and requires global coordinates to be first projected into the local coordinate frame, which can lead to redundant data preprocessing for multi-agent prediction. Therefore, how to efficiently scale the model to predict multiple agents simultaneously can be an exciting topic. Besides, we reduce the problem by building a goal distribution only conditioned on a static map and the history trajectories. How to handle dynamic environment states and incorporate a single agent's anticipations of the others' future motion as a variable in the model is worth exploring in the future.

References

Bahari, M., Saadatnejad, S., Rahimi, A., Shaverdikondori, M., Shahidzadeh, A. H., Moosavi-Dezfooli, S.-M., and Alahi, A. (2022). Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17123–17133.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Springer.

Chai, Y., Sapp, B., Bansal, M., and Anguelov, D.

(2020). Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In Kaelbling, L. P., Kragic, D., and Sugiura, K., editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR.

Chang, M.-F., Lambert, J. W., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., and Hays, J. (2019). Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65.

Dendorfer, P., Elflein, S., and Leal-Taixé, L. (2021). Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13158–13167.

Filos, A., Tigkas, P., McAllister, R., Rhinehart, N., Levine, S., and Gal, Y. (2020). Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., and Schmid, C. (2020). Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533.

Gilles, T., Sabatini, S., Tsishkou, D., Stanciulescu, B., and Moutarde, F. (2022). Gohome: Graph-oriented heatmap output for future motion estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9107–9114.

Gu, J., Sun, C., and Zhao, H. (2021). Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15303–15312.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264.

Hu, Y., Jia, X., Tomizuka, M., and Zhan, W. (2022a). Causal-based time series domain generalization for vehicle intention prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7806–7813. IEEE.

Hu, Y., Zhan, W., and Tomizuka, M. (2022b). Scenario-transferable semantic graph reasoning for interaction-aware probabilistic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):23212–23230.

Huang, Z., Mo, X., and Lv, C. (2022). Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2605–2611.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., and Savarese, S. (2019). Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M. (2017a). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 336–345.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., and Chandraker, M. (2017b). Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., and Urtasun, R. (2020). Learning lane graph representations for motion forecasting. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, pages 541–556, Cham. Springer International Publishing.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Lu, J., Zhan, W., Tomizuka, M., and Hu, Y. (2022). Generalizability analysis of graph-based trajectory predictor with vectorized representation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13430–13437.

Sohn, K., Lee, H., and Yan, X. (2015). Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.

Tran, H., Le, V., and Tran, T. (2021). Goal-driven long-term trajectory prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 796–805.

Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K. S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C. P., Anguelov, D., and Sapp, B. (2022). Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7814–7821.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, L., Hu, Y., Sun, L., Zhan, W., Tomizuka, M., and Liu, C. (2022a). Transferable and adaptable driving behavior prediction.

Wang, T., Roberts, A., Hesslow, D., Scao, T. L., Chung, H. W., Beltagy, I., Launay, J., and Raffel, C. (2022b). What language model architecture and pretraining objective works best for zero-shot generalization? In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22964–22984. PMLR.

Ye, M., Cao, T., and Chen, Q. (2021). Tpcn: Temporal point cloud networks for motion forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11318–11327.

Zeng, W., Liang, M., Liao, R., and Urtasun, R. (2021). Lanercnn: Distributed representations for graph-centric motion forecasting. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 532–539.

Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., and Tomizuka, M. (2019). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps.

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., Schmid, C., Li, C., and Anguelov, D. (2021). Tnt: Target-driven trajectory prediction. In Kober, J., Ramos, F., and Tomlin, C., editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 895–904. PMLR.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Not Applicable**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Not Applicable**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **Not Applicable**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes**
   (d) Information about consent from data providers/curators. **Not Applicable**

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

A Derivations of Training Objectives

In this section, we provide detailed mathematics of the variational mixture of Gaussians with Normal-Wishart prior and the derivation of z-posterior and training objectives.

A.1 Conjugate Prior

In the GNeVA model, we leverage the Normal-Wishart distribution as the prior for the mean $\boldsymbol{\mu}$ and precision $\boldsymbol{\Lambda}$ the Gaussian distribution. The distribution is in the form

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \eta, \beta, V, \nu) = \mathcal{N}(\boldsymbol{\mu} \mid \eta, (\beta\Lambda)^{-1})\mathcal{W}(\boldsymbol{\Lambda} \mid V, \nu). \tag{13}$$

Herein, μ and V are the *prior mean* and the positive-definite *prior scale matrix*, while β and ν control the strengths of belief in two priors, respectively. The density of the Normal-Wishart distribution is a product of a conditional normal distribution for the mean variable and a Wishart distribution for the precision

$$\begin{aligned}
\mathcal{N}(\mu \mid \eta, (\beta\Lambda)^{-1}) &= \frac{\beta^{\frac{D}{2}} \det(\Lambda)}{2\pi^{\frac{D}{2}}} \exp\left\{-\frac{\beta}{2}(\boldsymbol{\mu} - \eta)^\intercal \Lambda (\boldsymbol{\mu} - \eta)\right\}, \\
\mathcal{W}(\Lambda \mid V, \nu) &= \frac{\det(\Lambda)^{\frac{\nu - D - 1}{2}} \exp\left\{-\frac{1}{2}\mathrm{trace}(\Lambda V^{-1})\right\}}{2^{\frac{\nu D}{2}} \Gamma_D\left(\frac{\nu}{2}\right) \det(V)^{\frac{\nu}{2}}},
\end{aligned} \tag{14}$$

where $\Gamma_D(\cdot)$ is the *multivariate gamma function* and D is the dimensionality of the normal variable.

A.2 KL Divergence

The general objective for training the GNeVA model is to maximize the probability of ground-truth goal g. Following the notations, we have the log-probability given by

$$\begin{aligned}
\log p(g) &= \log \int_{\boldsymbol{\mu}} \int_{\boldsymbol{\Lambda}} \int_z p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z) d\boldsymbol{\mu} d\boldsymbol{\Lambda} dz \\
&= \log \int_{\boldsymbol{\mu}} \int_{\boldsymbol{\Lambda}} \int_z p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z) \frac{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} d\boldsymbol{\mu} d\boldsymbol{\Lambda} dz \\
&= \log \mathbb{E}_{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda} \mid g)} \left[\frac{p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z)}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})}\right]
\end{aligned} \tag{15}$$

According to the Jensen's Inequality,

$$\log \mathbb{E}_{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} \left[\frac{p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z)}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})}\right] \geq \mathbb{E}_{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} \left[\log \frac{p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z)}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})}\right]. \tag{16}$$

Hence, we can reduce the maximizing of log-probability to maximize its evidence lower bound (ELBO). Recall that our variational posterior distribution is given by $q(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) q(\boldsymbol{\Lambda}) q(\boldsymbol{z} \mid \boldsymbol{g}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$, where we drop the observation variables for simplicity. Therefore, the ELBO is futher given by

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}} &= \mathbb{E}_{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} \left[\log \frac{p(g, \boldsymbol{\mu}, \boldsymbol{\Lambda}, z)}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})}\right] \\
&= \mathbb{E}_{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} \log p(g \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, z) + \mathbb{E}_{q(\boldsymbol{\Lambda})} \int_{\boldsymbol{\mu}} \log \frac{p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})}{q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})} q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) d\boldsymbol{\mu} \\
&+ \int_{\boldsymbol{\Lambda}} \log \frac{p(\boldsymbol{\Lambda})}{q(\boldsymbol{\Lambda})} q(\boldsymbol{\Lambda}) d\boldsymbol{\Lambda} + \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})} \int_z \log \frac{p(z)}{q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda})} q(z, \boldsymbol{\mu}, \boldsymbol{\Lambda}) d\boldsymbol{\mu} d\boldsymbol{\Lambda} \\
&= \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) q(z)} \log p(g \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, z) - \mathbb{E}_{q(\boldsymbol{\Lambda})} D_{KL}\left(q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) \| p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})\right) \\
&- D_{KL}\left(q(\boldsymbol{\Lambda}) \| p(\boldsymbol{\Lambda})\right) - \mathbb{E}_{q(\boldsymbol{\mu}, \boldsymbol{\Lambda})} D_{KL}\left(q(z \mid g, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \| p(z)\right).
\end{aligned} \tag{17}$$

One benefit of using a conjugate prior is that it yields a closed-form solution to expectations in calculating the ELBO loss function. For the first term in equation 17, we can leverage the following property of multivariate normal distribution

$$x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \Rightarrow \mathbb{E}\left[x^\intercal \boldsymbol{A} x\right] = \mathrm{trace}(\boldsymbol{A}\boldsymbol{\Lambda}^{-1}) + \boldsymbol{\mu}^\intercal \boldsymbol{A} \boldsymbol{\mu}, \tag{18}$$

which allows us to transform the first term in the form

$$\begin{aligned}\mathbb{E}_{q(z,\boldsymbol{\mu},\boldsymbol{\Lambda})} \log p(g \mid \boldsymbol{\mu}, \boldsymbol{\Lambda}, z) &= \frac{1}{2}\mathbb{E}_{q(z,\boldsymbol{\mu},\boldsymbol{\Lambda})}\left[-(g-\boldsymbol{\mu})^\intercal \boldsymbol{\Lambda}(g-\boldsymbol{\mu}) - D\log(2\pi) + \log\det(\boldsymbol{\Lambda})\right] z \\ &\propto -\frac{1}{2}\sum_{c=1}^{C} q(z)\mathbb{E}_{q(\boldsymbol{\Lambda}_c)}\left[(g-\boldsymbol{\mu}_c)^\intercal \Lambda_c (g-\boldsymbol{\mu}_c) + \frac{D}{\beta_c} - \log\det(\Lambda)\right] \\ &= -\frac{1}{2}\sum_{c=1}^{C} q(z)\left[\nu_c (g-\boldsymbol{\mu}_c)^\intercal V_c (g-\boldsymbol{\mu}_c) + \frac{D}{\beta} - \log\det(V) - \psi_D(\frac{\nu_c}{2}) + D\log\pi\right],\end{aligned} \tag{19}$$

where $\psi_D(\cdot)$ is the *multivariate digamma function*. Meanwhile, the second and the third term in the equation 17 also have closed-form solutions (Bishop and Nasrabadi, 2006) in the form

$$\mathbb{E}_{q(\boldsymbol{\Lambda})} D_{KL}\left(q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) \| p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})\right) = \frac{1}{2}\sum_{c=1}^{C} \beta_0 \nu_c (\eta_c - \eta_0)^\intercal V_c (\eta_c - \eta_0) + \frac{K}{2}(\frac{\beta_0}{\beta_c} - \log\frac{\beta_0}{\beta_c} - 1); \tag{20}$$

$$D_{KL}\left(q(\boldsymbol{\Lambda}) \| p(\boldsymbol{\Lambda})\right) = \frac{1}{2}\sum_{c=1}^{C} \nu_c \left[\mathrm{trace}(V_0^{-1} V_c) - D\right] - \nu_0 \log\det(V_0^{-1} V_c) + 2\log\frac{\Gamma_D(\frac{\nu_0}{2})}{\Gamma_D(\frac{\nu_c}{2})} + (\nu_c - \nu_0)\psi_D(\frac{\nu_c}{2}). \tag{21}$$

A.3 Variational z-posterior

According to Bishop and Nasrabadi (2006), the optimal factor of $\log q(z)$ is derived from

$$\log q^*(z) = \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}} \log p(g|\boldsymbol{\mu}, \boldsymbol{\Lambda}, z) + \text{const.} \tag{22}$$

Therefore, we can make use of the results in equation 19 and estimate the variational z-posterior by

$$\begin{aligned}\log q(z=c) &= \log\pi_c + \frac{1}{2}\mathbb{E}_{q(\boldsymbol{\Lambda}_c)}[\log\det(\boldsymbol{\Lambda}_c)] - \frac{D}{2}\log(2\pi) - \frac{1}{2}\mathbb{E}_{q(\boldsymbol{\mu},\boldsymbol{\Lambda})}\left[(g-\boldsymbol{\mu}_c)^\intercal \boldsymbol{\Lambda}_c (g-\boldsymbol{\mu}_c)\right] \\ &= \log\pi_c + \log\det(V_c) + \psi_2(\frac{\nu_c}{2}) - \frac{D}{2\beta_c} - \frac{\nu_c}{2}(g-\boldsymbol{\mu}_c)^\intercal V_c (g-\boldsymbol{\mu}_c) + \text{const}\end{aligned} \tag{23}$$

B Implementation Details

B.1 Model

We implement our model in PyTorch. All the Multi-layer Perceptrons (MLPs) inside the GNeVA model consist of a hidden layer and an output layer, with layer normalization and ReLU activation. The size of the hidden feature is set to 128. For parameterization of the goal spatial distribution, we have two attention-based modules that fuse features of the map and surrounding participants with the target vehicle, including a Context Attention Module and an Interaction Attention Module. The Context Attention module parameterizes the posterior distribution of means $q(\boldsymbol{\mu})$. As shown in Figure 5a, the module contains a cascade of L_c self-attention layers. The map feature map $\mathbf{s}$, target feature map $\mathbf{x}_{\leq H}^{\text{target}}$, and the surrounding participants feature map $\mathbf{x}_{\leq H}^{\text{surr}}$ concatenate to form the context feature map $\mathbf{X}_c$ and pass an multi-head attention (MHA) layer (Vaswani et al., 2017). The operator is given as

$$\begin{aligned}\mathrm{MHA}(\mathbf{X}) &= \frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_k}}\mathbf{V}, \\ \text{where} \quad \mathbf{Q} = \mathbf{X}\mathbf{W}_q,\ \mathbf{K} &= \mathbf{X}\mathbf{W}_k,\ \mathbf{V} = \mathbf{X}\mathbf{W}_v.\end{aligned} \tag{24}$$

The $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are weights of linear layers. Following the MHA layer, there exists a ReLU activation, a residual connection layer, and a layer-normalization layer. The operation in a self-attention layer is given as

$$\mathrm{SelfAttention}(\mathbf{X}) = \mathrm{LayerNorm}(\mathbf{X} + \mathrm{ReLU}(\mathrm{MHA}(\mathbf{X}))). \tag{25}$$

In our implementation, we build our Context Attention module with $L_c = 3$ self-attention layers. To get the target feature map after context attention, we extract the corresponding row in the output feature map that

matches the locations of the target feature map in $\mathbf{X}_c$. It passes a consecutive MLP to derive the mean posterior parameters $\boldsymbol{\eta}^q$ and $\boldsymbol{\kappa}^q$.

As illustrated in Figure 5b, the Interaction Attention module shares a structure similar to the Context Attention module but differs in two ways. First, the input features are the feature maps of the target and surrounding participants. We drop the map features since they are temporally static; hence, we consider they have no effects on future uncertainty, as we mentioned in the paper. Meanwhile, we apply a mask to the surrounding participant feature map and remove those without potential future interactions with the target vehicle. In our implementation, we simplify the discussion by considering only agents with observations of their motion states at the observation horizon (i.e., $\mathbf{x}_H$) and remove the rest. We build our Interaction Attention module with $L_i = 1$ self-attention layers and get the target feature map after interaction attention the same way as in the Context Attention module.

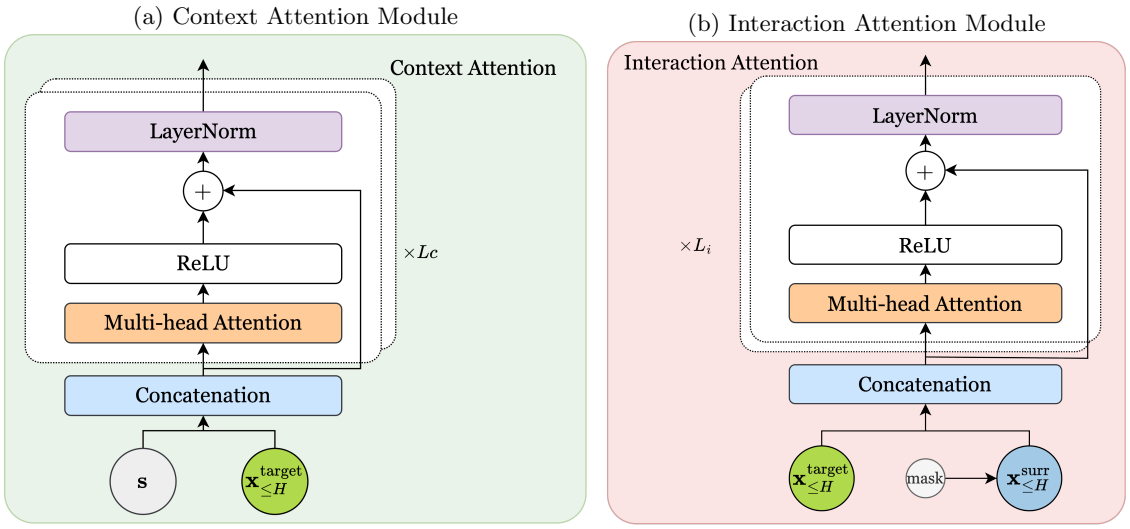


Figure 5: Illustrations of the Attention Modules.

B.2 Training

We train our GNeVA model with a batch size of 64 for 36 epochs on two NVIDIA A100 GPUs. We use AdamW (Loshchilov and Hutter, 2019) with a weight decay of 0.001. For scheduling, the learning rate increases linearly to 0.001 for the first 1000 steps, then cosine anneals to $3e-7$. In addition, we project raw coordinates in the source data into a target-centric reference frame by translating and then rotating the coordinates with respect to the location and heading angle of the target vehicle at the observation horizon.

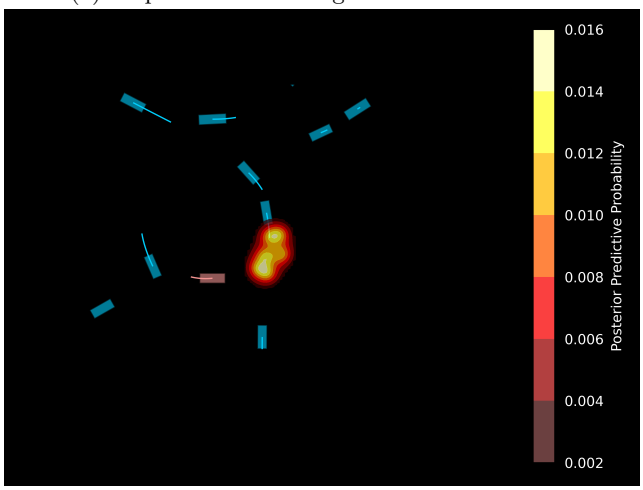
C Additional Experiment Results

C.1 Sensitivity Analysis: Road Geometry

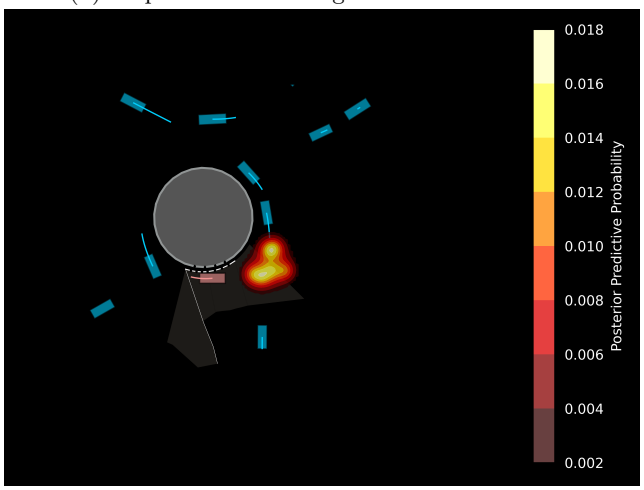
Road geometry plays a significant role in determining the spatial location of goal distributions. We expect the model to respond to changing road geometry to guarantee its generalization ability. In this experiment, we investigate how the proposed GNeVA model responds to the changing road geometry by adjusting the observation radius. A low observation radius leads to a reduced sensible range of surrounding lanes. We achieve this by removing all the lane polygons from the source data that have no intersection with the circle centered at the target vehicle's location at the observation horizon, with a radius of the observation radius. As a control factor, we assume complete observation of all the surrounding traffic under the same scenario.

Results in Figure 6 show that with increasing observation radius, the observation of the downstream lane structure becomes more accessible. Meanwhile, the shape of the posterior predictive goal distribution changes accordingly.

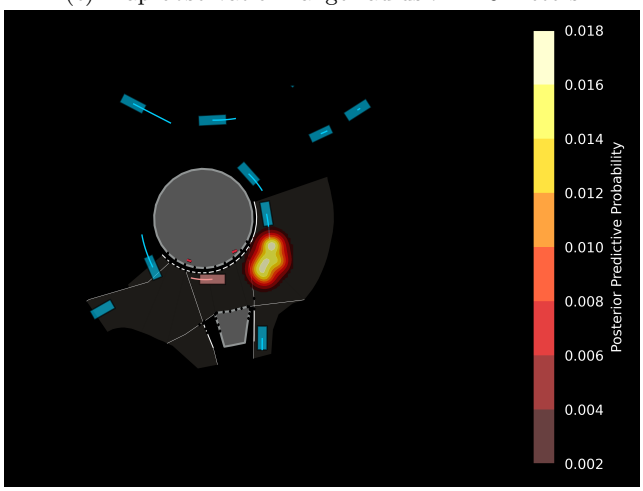
(a) Map observation range radius $r = 0$ meters



(b) Map observation range radius $r = 5$ meters



(c) Map observation range radius $r = 10$ meters



(d) Map observation range radius $r = 20$ meters

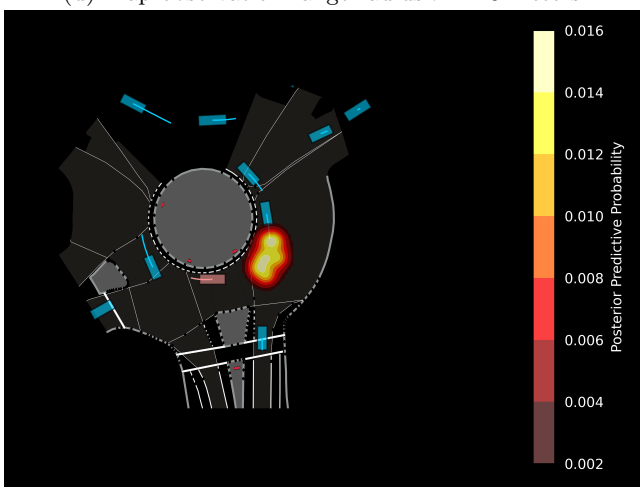


Figure 6: Posterior predictive goal distribution under different map observation range radius settings. The case is sampled from the `DR_USA_Roundabout_FT` scenario samples in the INTERACTION test dataset.

At first, when the radius is $r = 0$ meters, the GNeVA model has no access to the map information, which means the model predicts the future movement of the target agent based solely on its trajectories and all surrounding participants. As a result, the problem resembles a car-following problem, where the agent follows the path of its leading vehicle. In the visualization, the predictive distribution is concentrated around the history trajectory of the nearest potential leading vehicle. The result indicates that the GNeVA model can learn the car-following maneuver unsupervised.

Then, when the radius gradually increases from $r = 0$ to $r = 10$ meters, the GNeVA model has more information about the shape of downstream lanes. We can observe the mixture components first spreading and then concentrating back around the history trajectory of the leading vehicle. This indicates that GNeVA can pay attention to the shape of the known lane. When $r = 5$ meters, the model can only observe the current and the predecessor lane, where the current lane has a left boundary curved slightly towards the left and a right boundary slightly toward the right. Such a discrepancy potentially leads to the spreading of mixture components, indicating the target vehicle can go left or right. However, when $r = 10$ meters, the model can fully observe the shape of the downstream lane, reducing uncertainty about the possible future.

Finally, when we increase the observation radius to 20 and 50 meters, the shape of the distribution does change

quite significantly, which means the model has potentially learned only to pay attention to necessary downstream lanes. The visualization suggests that, instead of overfitting agents' history, the GNeVA model can effectively learn the road geometry constraints and assign proper attention to important lanes, which can be essential to guarantee its generalization ability (Lu et al., 2022).

C.2 Ablation Study: NMS Sampling

In this section, we investigate the effects of different NMS sampling settings on the final performance. Two key adjustable parameters of the NMS sampling algorithm are the **sampling radius** and the **sampling threshold**. The combined effect of these parameters determines the density of goal candidates sampled from the spatial distribution.

NMS Sampling Radius. Table 5 shows the prediction performance evaluated under different sampling radius settings. Ranging from 0.5 meters to 3.0 meters with a fixed IoU threshold of 0%, the optimal setting regarding mADE_6, mFDE_6, and MR_6 are observed at a radius of 2.5 meters, 2 meters, and 3 meters, respectively. Generally, with an increasing sampling radius, the spread of goal candidates increases, and within a proper range, the probability of hitting the ground-truth goal increases.

Table 5: Performance under different sampling radius. The results are evaluated on an IoU threshold of 0%.

Radius (meters)	mADE	mFDE	MR
0.5	0.3238	0.7700	0.1058
1.0	0.3111	0.7317	0.0953
1.5	0.2994	0.6930	0.0850
2.0	0.2952	**0.6387**	0.0780
2.5	**0.2946**	0.6730	0.0716
3.0	0.2970	0.6764	**0.0664**

NMS Sampling IoU Threshold. Results in Table 6 indicate that with a fixed sampling radius, the prediction performance varies with different IoU thresholds. In our experiment, we find that with a sampling radius of 2 meters, the optimal setting regarding mADE_6, mFDE_6, and MR_6 are observed at the threshold of 50%, 0%, and 25%. Generally, with an increasing IoU threshold, the spread of goal candidates decreases, leading to a denser candidate pool.

Table 6: Performance under different IoU thresholds. The results are evaluated on a sampling radius of 2 meters.

IoU (%)	mADE	mFDE	MR
0	0.2952	**0.6387**	0.0780
25	0.2965	0.6707	**0.0614**
50	**0.2949**	0.6706	0.0651

In our paper, we balance the three metrics and use a radius of 2 meters with 0% IoU threshold.