# Probabilistic Symmetry for Improved Trajectory Forecasting

**Anonymous Authors**[1]

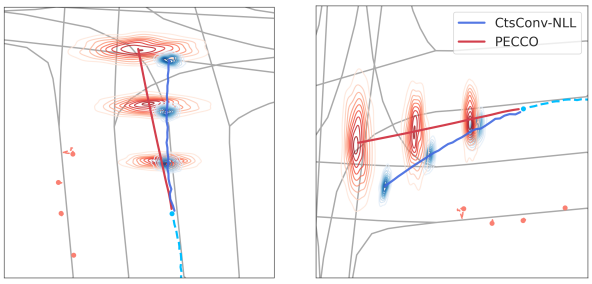
## Abstract

Trajectory prediction is a core AI problem with broad applications in robotics and autonomous driving. While most existing works focus on deterministic prediction, producing probabilistic forecasts to quantify prediction uncertainty is critical for downstream decision-making tasks such as risk assessment, motion planning, and safety guarantees. We introduce a new metric, mean regional score (MRS), to evaluate the quality of probabilistic trajectory forecasts. We propose a novel probabilistic trajectory prediction model, Probabilistic Equivariant Continuous COnvolution (PECCO) and show that leveraging symmetry, specifically rotation equivariance, can improve the predictions' accuracy as well as coverage. On both vehicle and pedestrian datasets, PECCO shows state-of-the-art prediction performance and improved calibration compared to baselines.

## 1. Introduction

Predicting the future trajectory of multiple agents is a critical task with applications in autonomous driving (Chang et al., 2019), single-cell RNA sequencing (Tong et al., 2020), and hurricane tracking (Hall and Jewson, 2007). The problem is difficult due to the strong dependency among the agents where the number of interactions grows quadratically with the number of agents; moreover, agent movements are influenced by the environmental features such as road boundaries on the map. In practice, it is impossible to exactly measure all dependencies among the agents and the influence from the environment. Such a partially observed setting introduces a significant amount of uncertainty.

Probabilistic modeling of trajectory dynamics provides a principled framework to represent uncertainty. However, it remains elusive how to quantify and evaluate the probabilistic forecasts. Consider the example of autonomous driving. Although along a single-lane road with little traffic it may be relatively certain drivers will continue at their current speed, it is difficult to know with certainty which way a vehicle approaching an intersection will turn purely from the past trajectories. By looking at map information such as turn lanes, historical traffic flow data, and inferred preferences, it is possible to place a wider or narrower probability distribution over the potential future trajectories. To succeed, such a probabilistic model must have *good coverage* and be *well-calibrated*; that is, it must cover all the likely future scenarios without being so broad and uncertain as to be useless.




(a) Original scene (b) Rotated scene

*Figure 1.* Prediction on the same scene rotated by 90 degrees. PECCO is consistent in trajectory and uncertainty prediction, whereas the non-equivariant model fails.

Trajectory prediction has been investigated extensively in deep learning (Liang et al., 2020b; Gao et al., 2020; Wang et al., 2018), mostly focusing on deterministic point estimates. There have been some efforts in making probabilistic forecasts for trajectories. For example, Tang and Salakhutdinov (2019) introduce latent variables to the encoder-decoder framework, Lee et al. (2017); Salzmann et al. (2020) propose conditional variational autoencoders (CVAE), but little analysis of the quality of such probabilistic predictions has been done.

Probabilistic symmetry plays an important role in the presence of data scarcity (Bloem-Reddy and Teh, 2020). Our key insight is to exploit *symmetry* to estimate high-dimensional conditional distributions with limited data. We assume the predicted probability distribution is rotation and translation equivariant. That is, if the input data is transformed, the probability distribution will also be likewise transformed. In Figure 1, we see the same car approaching an intersection from either the south or east. The scenes are related by a $\pi/2$ rotation. As the absolute compass direction is not particularly meaningful for local trajectory prediction, the model should thus output the same probability distributions over future trajectories for the car coming from the east as

that coming from the north, but rotated by $\pi/2$.

In this paper, we propose a **P**robabilistic **E**quivariant **C**ontinuous **CO**nvolutional model, PECCO, an equivariant probabilistic trajectory prediction model. PECCO uses continuous convolution (Wang et al., 2018) and enforces rotational equivariance as in (Walters et al., 2021), achieving more accurate and calibrated probablistic forecasts. Continuous convolution generalizes discrete convolution and is useful for modeling interactions in many particle systems. Rotational equivariance not only allows our model to produce physically consistent predictions, the multiplicative nature of equivariance allows us to model a density space with a smaller sample size. For each sample which an equivariant model is trained on, an equivariant model learns as if it were trained on all transformations of that sample by the symmetry group (Wang et al., 2021).

Other methods for enforcing equivariance include data augmentation and normalization. Data augmentation adds rotated versions of the samples to the training dataset and allows the model to learn rotational equivariance. However, this slows training drastically, requires greater model capacity, and rarely achieves the level of equivariance or accuracy as equivariant neural networks (Salzmann et al., 2020). To apply data normalization, we could rotate the scene so the agent is pointing up in each case as in (Gao et al., 2020). However, it is impossible to rotate the scene for multiple agents without a canonical reference frame. PECCO allows the weights to be relative to the local orientation of each agent without the need to rotate the scene repeatedly.

Our main contributions are as follows:

- We incorporate symmetry into probabilistic modeling of high-dimensional sequence data.
- We design an equivariant neural network, PECCO, to forecast multi-agent trajectories with uncertainty.
- We introduce a new metric, mean regional score (MRS), to evaluate the quality of probabilistic forecasts.
- We demonstrate that PECCO improves both accuracy and calibration of probabilistic forecasts for two real-world benchmark datasets.

## 2. Related Work

**Trajectory prediction.** Multi-agent trajectory forecasting has been extensively studied, approaches ranged from Kalman filters (Kalman, 1960) to non-linear Gaussian Process Regression models (Jordan, 1998). However, these methods either rely on strong assumptions of the dynamics, or do not explicitly model multi-agent interactions. We refer readers to (Rudenko et al., 2020) for a comprehensive survey of such methods. Advancements in deep learning have allowed flexible modeling of trajectory dynamics (Alahi et al., 2016; Lee et al., 2017; Wang et al., 2018; Deo and Trivedi, 2018; Sadeghian et al., 2019; Liang et al., 2020a; Salzmann et al., 2020; Liang et al., 2020b; Walters et al., 2021; Gao et al., 2020; Roddenberry et al., 2021), but they mostly focused on point estimation without uncertainty.

Recent methods have shifted to predicting distributions of future trajectories, capturing the inherent uncertainty. There are two main classes of models: (1) *explicitly* via exact likelihood (Tang and Salakhutdinov, 2019; Chai et al., 2019; Gu et al., 2021) and via variational inference (Sohn et al., 2015; Ivanovic and Pavone, 2019; Salzmann et al., 2020; Lee et al., 2017), or (2) *implicitly* with Generative Adversarial Networks (GANs) (Gupta et al., 2018; Liu et al., 2019). Our work falls into the first category where we model the distributions parametrically. A parametric model allows us to evaluate the likelihood of any future trajectories, which are useful for downstream planning tasks (Chai et al., 2019; Schwarting et al., 2018).

Despite the development in probabilistic modeling, there is no standard metric for quantifying the prediction uncertainty. Negative log likelihood often overfits the distribution (Guo et al., 2017), and best-of-n-sample results do not evaluate the full distribution (Ivanovic and Pavone, 2021). We argue that probabilistic forecasts should accurately reflect the uncertainty in the model predictions. We propose a non-parametric metric extending mean interval score (Gneiting and Raftery, 2007). Furthermore, by leveraging the internal symmetries of trajectories, our model produces more accurate and calibrated probabilistic forecasts.

**Equivariant Deep Learning.** Geometric deep learning that leverages invariance and symmetries has found wide applications in areas ranging from image recognition (Bao and Song, 2019; Worrall et al., 2017; Weiler and Cesa, 2019) to reinforcement learning (van der Pol et al., 2020). Equivariant neural networks are studied for modeling dynamics as well - Fuchs et al. (2020) use $\mathrm{SE}(3)$-equivariant transformers to predict trajectories for a small number of particles as a regression task, and (Walters et al., 2021) proposed a $\mathrm{S0}(2)$ equivariant continuous convolution for traffic trajectory prediction. All methods mentioned above are deterministic. (Köhler et al., 2020) and (Satorras et al., 2021) studies equivariant normalizing flows for modeling symmetric densities, however their domains focus on generative modeling rather than forecasting, and therefore differs from our work significantly. To our knowledge, no previous work has studied equivariant neural networks for probabilistic forecasting.

**Uncertainty Quantification (UQ).** Uncertainty quantification is critical for risk assessment in safety-critical do-

mains. Properly quantified uncertainties can be used to generate probabilistic constraints and whereby generate more robust planning and control strategies (Ostafew et al., 2016; Bujarbaruah et al., 2019). With the increasing use of deep learning in forecasting tasks, many works have UQ for neural networks (Luo et al., 2021; Wu et al., 2021; Guo et al., 2017). (Stankevičiūtė et al., 2021) proposes a conformal prediction algorithm for 1D RNN forecasters. (Luo et al., 2021) explores using conformal methods to produce collision warnings with safety guarantees. However, these works focus only on classification or 1-dimensional forecasts. To our knowledge, PECCO is the first paper to study quality of UQ for multivariate spatiotemporal forecasts.

In our experiments, we show that the uncertainty distributions produced by deep generative models for trajectory forecasting are often not calibrated. Hence, better UQ methods are needed. In this work, we expand metrics from quantile and conformal prediction to the multivariate setting, and introduce a method to evaluate forecasted uncertainties. We also present a model design that produces uncertainty distributions with better calibration.

## 3. Background

We give a short background on symmetry and equivariance and their probabilistic extension.

**Symmetry and Equivariance.** A *symmetry group* $G$ is a set together with a composition operation $\circ\colon G \times G \to G$ which is associative and has an identity and inverses.

The group $G$ can transform a vector space $V$ by specifying a *representation* which is a mapping $\rho\colon G \to \mathrm{GL}_n(V)$ sending each element of the group $G$ to an invertible $n \times n$ matrix such that $\rho(g_1 \circ g_2) = \rho(g_1)\rho(g_2)$.

Given a function $f\colon X \to Y$ such that $G$ has representations $\rho_X$ and $\rho_Y$ acting on $X$ and $Y$ respectively, we say $f$ is *$G$-equivariant* if for all $x \in X$ and $g \in G$, we have $\rho_Y(g)f(x) = f(\rho_X(g)x)$. That is, a transformation of the input of $x$ induces a corresponding transformation of the output. *Invariance* for the function $f$ is a special case in which $\rho_Y(g)y = y$.

**$\mathbf{SO(2)}$ Rotational Symmetry.** In this paper, we consider equivariance to 2D planar rotations, $\mathrm{SO}(2) = \{\mathrm{Rot}_\theta : 0 \le \theta < 2\pi\}$ which has composition $\mathrm{Rot}_{\theta_1} \circ \mathrm{Rot}_{\theta_2} = \mathrm{Rot}_{(\theta_1+\theta_2) \bmod 2\pi}$. As a Lie group, $\mathrm{SO}(2)$ also has an underlying topological structure given by the manifold $S^1 \cong \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\| = 1\}$. Here, we consider two specific representations of $\mathrm{SO}(2)$. The *standard representation* $\rho_1$ on $\mathbb{R}^2$ is by $2\times2$ rotation matrices. The *regular representation* $\rho_{\mathrm{reg}}$ on $L^2(\mathrm{SO}(2)) = \{\varphi : \mathrm{SO}(2) \to \mathbb{R} : |\varphi|^2 \text{ is integrable}\}$ is $\rho_{\mathrm{reg}}(\mathrm{Rot}_\phi)(\varphi) = \varphi \circ \mathrm{Rot}_{-\phi}$. Here $L^2$ is a Hilbert space.

**$\mathbf{SO(2)}$ Equivariant Continuous Convolution** Continuous convolution is the generalization of the discrete convolution for point clouds. The feature vector $\mathbf{f}^{(i)} \in \mathbb{R}^{c_{\mathrm{in}}}$ of particle $i$ forms a vector field $\mathbf{f}$, and the kernel of the convolution $K : \mathbb{R}^2 \mapsto \mathbb{R}^{c_{\mathrm{out}} \times c_{\mathrm{in}}}$ forms a matrix field: for each point $\mathbf{x} \in \mathbb{R}^2$, $K(\mathbf{x})$ is a $c_{\mathrm{out}} \times c_{\mathrm{in}}$ matrix. The continuous convolution is then be defined by

$$\mathbf{g}^{(i)} = \sum_j a(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|)K(\mathbf{x}^{(j)} - \mathbf{x}^{(i)}) \cdot \mathbf{f}^{(j)}.$$

By (Weiler and Cesa, 2019), this is $\mathrm{SO}(2)$-equivariant if

$$K(gv) = \rho_{\mathrm{out}}(g)K(v)\rho_{\mathrm{in}}(g^{-1}).$$

ECCO (Walters et al., 2021) defines the convolution kernel $K$ in polar coordinates $K(\theta, r)$. Let $\mathbb{R}^{c_{\mathrm{in}}}$ and $\mathbb{R}^{c_{\mathrm{out}}}$ be $\mathrm{SO}(2)$-representations $\rho_{\mathrm{in}}$ and $\rho_{\mathrm{out}}$ respectively, then the convolution kernel satisfies the equivariance condition as

$$K(\theta + \phi, r) = \rho_{\mathrm{out}}(\mathrm{Rot}_\theta)K(\phi, r)\rho_{\mathrm{in}}(\mathrm{Rot}_\theta^{-1}),$$

making the continuous convolution $\mathrm{SO}(2)$-equivariant.

**Conditional Coverage.** Given an input variable $X$ and output variable $Y$, a prediction interval with significance (error rate) $\alpha$ is *valid* if it satisfies

$$P(Y \in \hat{C}^\alpha | X = x) \ge 1 - \alpha \tag{1}$$

which means the ground truth output $Y$ falls into the prediction interval $\hat{C}^\alpha$ with at least $1 - \alpha$ probability, conditioned on the input. The left hand side of the inequality is known as *conditional coverage*, which we will refer to as simply *coverage* in this paper. Theoretically, it is difficult to ensure Equation (1) with statistical methods without further assumptions on data (Barber et al., 2020). Nevertheless, we show that our method satisfies Equation 1 in experiments.

## 4. Score Function for Probabilistic Forecast

We consider a probabilistic approach to trajectory forecasting. Ideally, we want the predicted distribution to be both sharp and valid under the definition in Equation 1. To quantify the uncertainty, we review the classic score function Mean Interval Score (MIS) and derive a 2 dimensional extension, Mean Regional Score (MRS).

**Mean Interval Score.** Mean Interval Score (MIS) is a proper scoring rule (Gneiting and Raftery, 2007) for interval forecasts that rewards narrower confidence intervals and encourages coverage. Specifically, let $Y \sim P_Y$ be a one-dimensional random variable, and an upper bound $u$, a lower bound $l$ be its estimated $\frac{\alpha}{2}$ and $(1-\frac{\alpha}{2})$ quantiles respectively,

MIS is defined using samples $y_i \sim P_Y$.

$$\text{MIS}(u, l; Y) = \frac{1}{N}\sum_{i=1}^{N}[(u-l) + \frac{2}{\alpha}(y_i - u)\mathbb{1}\{y_i > u\} + \frac{2}{\alpha}(l - y_i)\mathbb{1}\{y_i < l\}] \quad (2)$$

Where the three terms in Equation 2 accounts for the width of the interval, and how much the sample point exceeds the upper bound and the lower bound, respectively. MIS has the advantage of being easy to compute and does not require the model to be parametric (Gneiting and Raftery, 2007). It can also be used as an objective function for probabilistic forecasting (Wu et al., 2021).

**Mean Regional Score.** For n-D distributions, we introduce a new metric called the Mean Region Score (MRS), which generalizes MIS to higher dimensions. Let $Z \sim P_Z$ be an n-dimensional continuous random variable. First, we need to generalize $\alpha$-quantile and define the confidence region. Let the region bordering $z$ as $R(z)$, the confidence region of level $\alpha$

$$R_P(\alpha) = \inf\{R(z) | \int_{R(z)} P(z)dz \geq \alpha\} \quad (3)$$

is the smallest region whose probability exceeds level $\alpha$.

When the distribution is Gaussian $P_Z = \mathcal{N}(\mu, \Sigma)$ with mean $\mu \in \mathbb{R}^n$ and the covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ that is positive definite. The $(1-\alpha)$ confidence region of an n-D multivariate normal distribution $P = \mathcal{N}(\mu, \Sigma)$ is an ellipsoid that can be written as (Slotani, 1964):

$$R(1-\alpha) = \{z | (z-\mu)^T \Sigma^{-1}(z-\mu) \leq \chi_n^2(1-\alpha)\}.$$

where $\chi_n^2$ is the chi-squared distribution with n degrees of freedom. Then, for a given sample $z_i \in \mathbb{R}^n$, we can draw an ellipsoid whose edge coincides $z_i$ defined as

$$R(z_i) = \{z | (z-\mu)^T \Sigma^{-1}(z-\mu) \leq c'\}$$

where $c' = (z_i - \mu)^T \Sigma^{-1}(z_i - \mu)$.

The MRS score of the $(1-\alpha)$ prediction interval thus can be evaluated as:

$$\text{MRS}(R; Z) = \frac{1}{N}\sum_{i=1}^{N}[|R(1-\alpha)| + \frac{1}{\alpha}|R(z_i) \backslash R(1-\alpha)|\mathbb{1}\{z_i \in R^C(1-\alpha)\}] \quad (4)$$

where the first term corresponds to the area of the confidence region and the second term measures how much each data $z_i$ deviates from the region. $|\cdot|$ measures the size of the set. It is also easy to prove (see Appendix A.4) that MRS is a proper scoring rule (Gneiting and Raftery, 2007), which means MRS is optimized if and only if the distribution $P$ recovers that of $Z$.

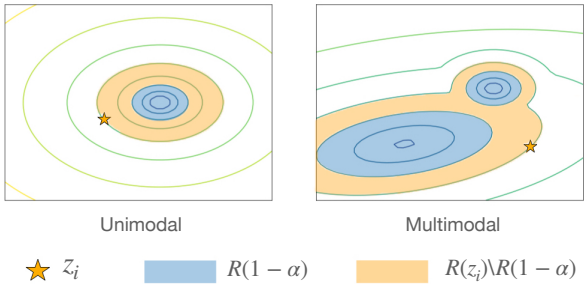



*Figure 2.* MRS illustration for unimodal and multimodal distributions. Given a sample $z_i$, MRS calculates the area of blue region plus that of the the orange region scaled by $1/\alpha$.

Figure 2 illustrates MRS for both unimodal and multimodal distributions. Note that when the distribution is single-modal, the confidence interval in Equation 4 is both *equally tailed* and the *shortest* confidence interval. For multi-modal distributions, we solve for the *smallest* confidence region to cover probable events. In appendix B.1 we present a method to estimate MRS for arbitrary 2-D distributions, where we divide the domain space into small grids to estimate density, and obtain $R(1-\alpha)$ and $R(z_i)$ by assembling high density regions.

## 5. PECCO

### 5.1. Problem Setup

Given past trajectories of $n$ agents over $t$ timesteps $x_j^{(i)} \in \mathbb{R}^2$ where $1 \leq i \leq n$ and $1 \leq j \leq t$, and the environmental context information $\mathbf{e}$ including marker positions of map lane boundaries, we model the probability distribution of agents' positions over $k$ future time steps as $p_\theta(x_{t+1:t+k} | x_{1:t}, \mathbf{e})$, with $x_j = (x_j^{(1)}, \cdots, x_j^{(n)})$ being the positions of all agents. We introduce PECCO, a deep learning model that leverages rotational equivariance to produce probabilistic forecasts.

The high-level architecture of our model is illustrated in Figure 3. PECCO takes as input the positions of all agents $x_{1:t}$ in the past, and the covariance matrix $\Sigma_{x,j}$ at time $j$. It outputs the probability distribution of each agents' velocity as a 2-D Gaussian $\mathcal{N}(\mu_{v,j+1}^{(i)}, \Sigma_{v,j+1}^{(i)})$ for the next time step. The velocity distribution is then integrated into a position distribution $\mathcal{N}(\mu_{x,j+1}^{(i)}, \Sigma_{x,j+1}^{(i)})$ through dynamics integration. PECCO predicts the future $k$ timesteps autoregressively. The output distributions are guaranteed to be equivariant under rotation.

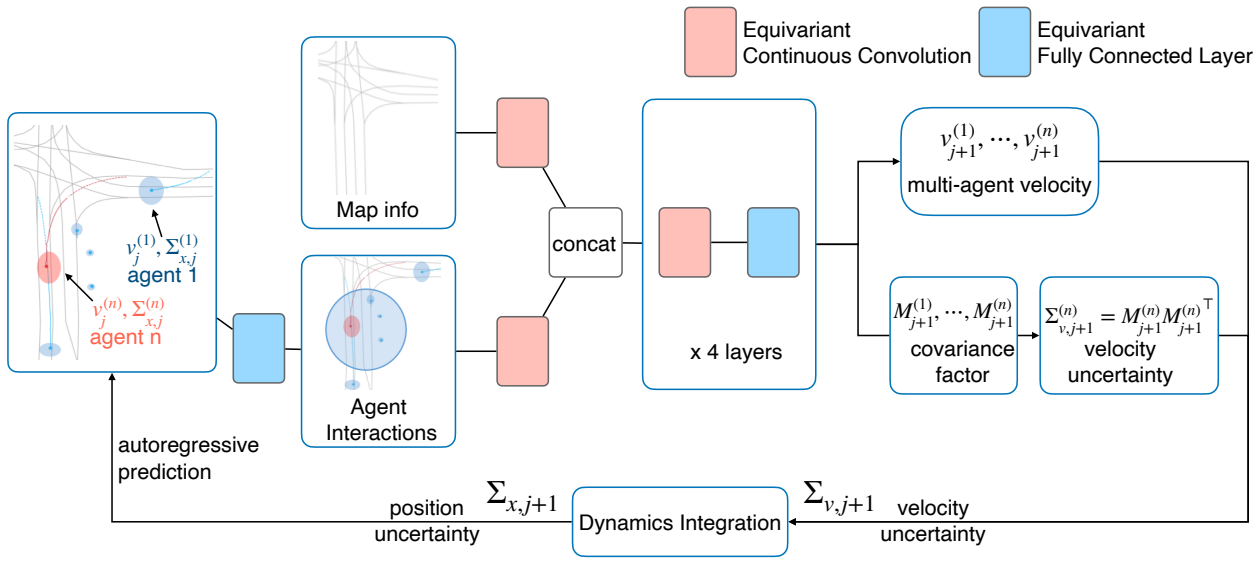



*Figure 3.* Overview of PECCO's model architecture. Agent trajectories consisting of velocities and position uncertainties are encoded along with map information by equivariant continuous convolution and fully connected layers. The model outputs $v_{j+1}$ and $\Sigma_{v,j+1}$ for all agents, which we use to calculate position uncertainty $\Sigma_{x,j+1}$ via dynamics. The model takes in the forecast and predicts autoregressively.

### 5.2. Probabilistic Symmetry through Equivariant Neural Networks

Rotational equivariance effectively reduces the dimension of the data space by placing different samples in the same equivalence class. This improves data coverage for better probabilistic modeling.

Intuitively, real-world trajectory dynamics have intrinsic symmetry. That is, if the past trajectories and environmental data such as the lane and map are rotated, then the probability of a rotated future trajectory will be equally likely. We can model the probability density function $p_\theta$ as an invariant function of its inputs,

$$p_\theta(x_{t+1:t+k}|x_{1:t}, \mathbf{e}) = p_\theta(gx_{t+1:t+k}|gx_{1:t}, g\mathbf{e}). \quad (5)$$

Here, past and future positions $x_j^{(i)} \in \mathbb{R}^2$ are transformed according the standard representation $\rho_1$.

In order to implement Equation 5, we assume future positions follow a multivariate normal distribution $x_j^{(i)} \sim N(\mu_j^{(i)}, \Sigma_j^{(i)})$. This is a common assumption in trajectory forecasting literature (Rudenko et al., 2020) and provides a parametric closed form for computing confidence regions and the score function.

We aim to construct an equivariant neural network $f_\theta$ which outputs the parameters $\mu_j^{(i)}$ and $\Sigma_j^{(i)}$ autoregressively, taking as input probability distributions over positions of all agents in the past $k$ timesteps

$$\mu_{j+1}, \Sigma_{j+1} = f_\theta(\mu_{j-t+1:j}, \Sigma_{j-t+1:j}, \mathbf{e}).$$

where $\mu_j = (\mu_j^{(1)}, \ldots, \mu_j^{(n)})$ and $\Sigma_j = (\Sigma_j^{(1)}, \ldots, \Sigma_j^{(n)})$ and $\mathbf{e}$ denotes environmental information.

In this case, the equivariance of $f_\theta$ leads to the desired invariance of $p_\theta$. This may be seen as a partial evaluation or currying of the conditional probability density function which has the effect of transforming invariance to equivariance. The following proposition relates equivariant networks with probabilistic symmetry. See Appendix A for a proof.

**Proposition 1.** *If the one-step probabilistic forecasting model $f_\theta$ is $G$-equivariant, then the probability distribution $p_\theta(x_{t+1:t+k}|x_{1:t}, \mathbf{e})$ is invariant as in Equation 5.*

**Equivariant Conditional Normal Distributions.** In order to enforce $\mathrm{SO}(2)$-equivariance for $f_\theta$, the following proposition describes how the mean and covariance matrix for a 2-D Gaussian transforms under a rotation of $\mathbb{R}^2$.

**Proposition 2.** *Given multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ over $\mathbb{R}^2$ with probability density function $p_{\mu,\Sigma}$ and $g \in \mathrm{SO}(2)$, then $\mathcal{N}(g\mu, g\Sigma g^T)$ is also a multivariate normal distribution and $p_{g\mu,g\Sigma g^T}(v) = p_{\mu,\Sigma}(g^{-1}v)$ for all $v \in \mathbb{R}^2$.*

To guarantee the covariance matrix of $f_\theta$ is positive-definite and symmetric, i.e. $\Sigma \in \mathrm{PosDefSym}_2(\mathbb{R})$, we make use of the following fact.

**Proposition 3.** *The map*

$$\varphi\colon \mathrm{GL}_2(\mathbb{R}) \to \mathrm{PosDefSym}_2(\mathbb{R})$$
$$M \mapsto MM^T$$

*is surjective and equivariant. For* $g \in \mathrm{SO}(2)$,

$$\varphi(gM) = g\varphi(M)g^T. \tag{6}$$

*Moreover,* $\varphi$ *admits a one-sided inverse which is also equivariant,*

$$\psi\colon \mathrm{PosDefSym}_2(\mathbb{R}) \to \mathrm{GL}_2(\mathbb{R}) \tag{7}$$

$$\Sigma \mapsto Q\Lambda^{\frac{1}{2}} \tag{8}$$

*where* $Q\Lambda Q^T$ *is the eigendecomposition of* $\Sigma$ *and* $Q$ *is orthogonal. That is,* $\varphi(\psi(\Sigma)) = \Sigma$.

As a consequence of Proposition 2 and Proposition 3, we can ensure that $f_\theta$ outputs a normal distribution which transforms correctly under rotations by setting

$$\begin{aligned} \mu_{j+1}, M_{j+1} &= \tilde{f}_\theta(\mu_{j-k:j}, M_{j-k:j}, \mathbf{e}) \\ \Sigma_{j+1} &= M_{j+1}M_{j+1}^T \end{aligned} \tag{9}$$

and by constraining $\tilde{f}_\theta$ to be equivariant with respect to the action in Equation 6. In this case, $\mathrm{SO}(2)$ acts by transforming the the columns of $M$ independently as vectors in $\mathbb{R}^2$. Thus the data $(\mu_j^{(i)}, \Sigma_j^{(i)})$ for each agent and time step is comprised of 3 copies of the standard representation $\rho_1$ as defined in Section 3. Given this $\mathrm{SO}(2)$-action, we can enforce $\mathrm{SO}(2)$-equivariance in the neural network $\tilde{f}_\theta$ using existing techniques from (Walters et al., 2021).

### 5.3. Dynamics Integration (dyna)

Instead of predicting the position directly, PECCO outputs a Gaussian distribution over the *velocity* as $\mathcal{N}(\mu_{v,j}, \Sigma_{v,j})$. More specifically, it predicts $(\mu_{v,j}, M_{v,j})$ at each time step and the covariance matrix $\Sigma_{v,j}$ is calculated as in Proposition 3. However, we want to obtain the uncertainty over *position* as $\Sigma_{x,j}$ and perform autoregressive forecasting. We leverage dynamics integration to propagate the uncertainty from velocity to position.

Assuming that all agents in the system can be approximated as linear discrete time dynamics $x_{j+1} = x_j + \Delta t \cdot v_j$, we can obtain the uncertainty of predicted position $\Sigma_{x,j+1}$ by

$$\Sigma_{x,j+1} = \Sigma_{x,j} + (\Delta t)^2 \Sigma_{v,j} + 2\Delta t \cdot \mathrm{cov}(x_j, v_j).$$

As in previous works (Salzmann et al., 2020), we make the assumption that the cross covariance matrix $\mathrm{cov}(x_j, v_j)$ is zero for simplicity of implementation. A consequence of such setup is that it enforces that the uncertainty grows monotonically over time. During training, the gradients are calculated after the entire trajectory is predicted.

## 6. Experiments

We show that our model achieves state of the art prediction accuracy and predicts a more calibrated confidence region compared to baseline models in two trajectory prediction scenarios - autonomous vehicle and pedestrian.

**Baselines.** We compare our model PECCO against the following baselines:

- `LSTM`: An encoder-decoder LSTM model for deterministic prediction optimizing mean square error. We also trained a version with cannonicalized data `LSTM-r`.
- `CtsConv` (Ummenhofer et al., 2019): Continuous convolution over point cloud data for trajectory prediction, a non-equivariant counterpart of ECCO. `CtsConv-aug` is trained with an additional data augmentation step where we randomly rotate the scenes.
- `ECCO` (Walters et al., 2021): State-of-the-art equivariant model for deterministic trajectory prediction.
- `Multiple Futures Prediction (MFP)` (Tang and Salakhutdinov, 2019): A encoder-decoder model for multimodal probabilistic forecasts.
- `Trajectron++` (Salzmann et al., 2020): State-of-the-art probabilistic trajectory prediction model with conditional VAE architecture.

For the deterministic models (`LSTM`, `CtsConv`), we modify them by training with Negative Log-Likelihood (NLL) and Mean Regional Score (MRS) as loss functions, the results for the deterministic baselines are in Appendix C.2.

**Evaluation Metrics.** We include the following metrics to evaluate both point estimate performance, as well as the quality of the predicted distributions.

- *Minimum Average Displacement Error* ($\mathrm{minADE}_6$): average $l_2$ displacement error over $k$ steps between the prediction and ground truth. We report ADE for deterministic models and the minimum ADE over 6 samples for probabilistic models.
- *Minimum Final Displacement Error* ($\mathrm{minFDE}_6$): $l_2$ displacement error between the predicted and the ground truth final position. We report FDE for deterministic models and the minFDE over 6 samples for probabilistic models.
- *Negative Log Likelihood* (NLL): Mean NLL of the ground truth trajectory under the output distribution.
- *Mean Regional Score* (MRS): Score function for confidence region prediction introduced in Equation 4. We report MRS for $\alpha = 0.1$, the 90% cofidence region.
- *Coverage*: The empirical estimate of probability of the true value lying in the predicted interval, defined in

| Model | $\text{minADE}_6 \downarrow$ | $\text{minFDE}_6 \downarrow$ | NLL↓ | MRS ↓ | Cov@1s(%) | Cov@2s | Cov@3s |
|---|---|---|---|---|---|---|---|
| | | | Argoverse | | | | |
| LSTM-NLL | 1.64 ± .05 | 4.17 ± .10 | 3.07 ± .08 | 19.1 ± 1.5 | 8.8 ± 0.7 | 8.5 ± 0.7 | 7.0 ± 0.8 |
| LSTM-MRS | 1.86 ± .04 | 4.71 ± .08 | 12096. ± 305. | 7.4 ± 0.3 | 99.4 ± 0.2 | 99.8 ± 0.1 | 99.8 ± 0.1 |
| LSTM-NLL-r | 1.61 ± .02 | 4.15 ± .08 | **2.48** ± .03 | 20.3 ± 2.9 | 10.1 ± 1.5 | 10.5 ± 1.0 | 9.8 ± 1.9 |
| LSTM-MRS-r | 1.66 ± .04 | 4.22 ± .08 | 37.26 ± 2.31 | **7.3** ± 0.2 | **86.4** ± 5.1 | **88.5** ± 4.7 | **91.6** ± 5.4 |
| CtsConv-NLL | 1.74 ± .03 | 4.43 ± .06 | 49.13 ± 2.16 | 1409. ± 277. | 6.3 ± 2.2 | 0.02 ± .01 | 0.01 ± .01 |
| CtsConv-MRS | 1.76 ± .02 | 4.45 ± .05 | 16.80 ± .51 | 114. ± 10. | 65.4 ± 3.5 | 11.4 ± 1.2 | 0.21 ± .10 |
| CtsConv-NLL-aug | 1.66 ± .02 | 4.23 ± .06 | 11.81 ± .01 | 857. ± 126. | 22.3 ± 2.1 | 1.7 ± 0.5 | 0.02 ± .01 |
| Trajectron++ | 1.83 ± .02 | 3.85 ± .07 | 3.15 ± .17 | 55.2 ± 2.5 | 45.5 ± 5.3 | 37.6 ± 3.2 | 34.9 ± 2.5 |
| MFP | 1.53 ± .04 | 3.77 ± .06 | 3.56 ± .02 | 12.5 ± 1.2 | 81.2 ± 5.1 | 53.0 ± 4.9 | 21.3 ± 6.8 |
| PECCO | **1.39** ± .02 | **3.41** ± .03 | 4.26 ± 0.1 | 8.36 ± 0.16 | 74.9 ± 0.2 | 93.6 ± 0.8 | 94.5 ± 0.9 |
| | | | TrajNet++ | | | | |
| LSTM-NLL-r | 0.85 ± .02 | 1.64 ± .03 | 2.78 ± .02 | 4.5 ± 0.6 | 21.9 ± 4.3 | 23.2 ± 4.2 | 23.7± 3.9 |
| LSTM-MRS-r | 0.94 ± .01 | 1.84 ± .02 | 123.7 ± 5.0 | **3.3** ± 0.1 | 95.7 ± 5.2 | **94.3** ± 5.6 | 96.5 ± 3.4 |
| CtsCov-NLL | 1.08 ± .02 | 2.36 ± .09 | 5.33 ± .08 | 33.1 ± 3.2 | 73.7 ± 20.6 | 20.7 ± 5.2 | 15.2 ± 6.7 |
| CtsConv-MRS | 0.96 ± .02 | 2.28 ± .07 | 5.97 ± .20 | 8.2 ± 0.9 | **93.4** ± 5.2 | 64.5 ± 5.2 | 46.1 ± 4.8 |
| CtsCov-NLL-aug | 0.92 ± .01 | 1.76 ± .03 | 6.74 ± .11 | 20.0 ± 1.0 | 62.1 ± 3.3 | 36.3 ± 4.9 | 34.1 ± 5.8 |
| Trajectron++ | 1.14 ± .03 | 2.31 ± .05 | 2.83 ± .12 | 45.1± 6.4 | 50.2 ± 2.2 | 45.8 ± 3.5 | 32.9 ± 3.5 |
| MFP | 0.85 ± .02 | 1.70 ± .04 | **2.20** ± .04 | 15.5 ± 0.6 | 79.1 ± 4.3 | 32.5 ± 3.1 | 22.8 ± 3.2 |
| PECCO | **0.72** ± .02 | **1.41** ± .05 | 2.37 ± .04 | 6.1 ± 0.1 | 80.8 ± 4.5 | 81.9 ± 2.3 | **84.5** ± 3.0 |

*Table 1.* Performance comparison on the benchmark Argoverse and TrajNet++ dataset. Bold-faced numbers indicate the best performance. Cov@1s(%) is the percentile coverage at the 1 second prediction; prediction is more *calibrated* if closer to 90%. We can see PECCO achieves superior regression accuracy while maintaining good coverage.

equation 1. We report the coverage of 90% quantile of the predicted Gaussian, i.e. $\alpha = 0.1$. The prediction is more *calibrated* if the coverage is closer to 90%.

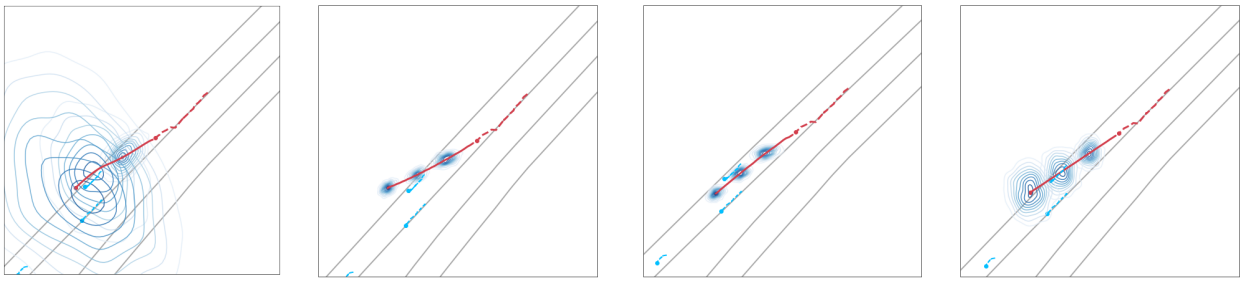
**Datasets.** The Argoverse autonomous vehicle motion forecasting (Chang et al., 2019) is a widely used vehicle trajectory prediction benchmark. The task is to predict 3 second trajectories based on all vehicle history in the past 2 seconds recorded at 10Hz. TrajNet++ (Sadeghian et al., 2018) is a popular pedestrian trajectory benchmark with a focus on agent-agent scenarios. The task is to predict 12 time steps for all agents given 9. For data and training details see C.1.

We report the performance on the official validation set for Argoverse, and on a 10% test split for TrajNet++. PECCO predicts all agents' trajectories jointly. Following literature, we report on metrics for the agent vehicle (Argoverse) and primary pedestrian (TrajNet++).

**Prediction Accuracy.** Table 1 show the prediction performance on two benchmark datasets. PECCO achieves better regression accuracy compared to non-equivariant baseline in terms of minADE and minFDE, with a notable 9% improvement in minADE over the the best performing baseline, MFP. PECCO's improved probability coverage allows for more diverse sampling and hence can produce trajectories closer to ground truth. For TrajNet++, PECCO outperforms baseline models in both prediction accuracy and calibration.

**Uncertainty Evaluation.** PECCO's uncertainty estimates are able to achieve consistently good coverage compared to other methods whose coverage deteriorates over the prediction horizon. By comparing LSTM-NLL and LSTM-MIS with their augmented counterparts LSTM-NLL-aug and LSTM-MRS-aug, we can see that data augmentation through rotation improves both accuracy and calibration. Models trained with MRS loss (LSTM-MRS, LSTM-MRS-aug, CtsConv-MRS) can consistently achieve high coverage, but their predictions suffer from the wide distribution as reflected by their worse NLL and minADE/FDE numbers. On the other hand, even though trained using NLL loss, PECCO is able to maintain good calibration while achieving improved accuracy by leveraging equivariance.

Figure 4 visualizes a typical situation to illustrate this difference. We plot the predicted distribution at 10, 20, and 30 timesteps of prediction (1 timestep is 0.1 seconds). We can see the probable region predicted by LSTM-MRS-r explodes at timesteps 20 and 30, whereas CtsConv-NLL and CtsConv-MIS tend to be overconfident in their predictions. PECCO is able to predict a Gaussian that covers the cases for both staying in the lane and changing to the right lane.



(a) LSTM-MRS-r (b) CtsConv-NLL (c) CtsConv-MRS (d) PECCO

*Figure 4.* Comparison of uncertainty predicted at a lane change. Red trajectories are the agent of interest in the same scenario. Note how the LSTM predicted uncertainty explodes after a few time steps, and PECCO is able to model possibility of both staying and lane change.

| **Model** | $\text{minADE}_6 \downarrow$ | $\text{minFDE}_6 \downarrow$ | MRS $\downarrow$ | Cov@1s(%) | Cov@2s | Cov@3s |
|---|---|---|---|---|---|---|
| Conformal LSTM | 2.45 $\pm$ 0.09 | 4.68 $\pm$ 0.15 | 198.1 $\pm$ 12.0 | 90.1 $\pm$ 0.1 | 92.7 $\pm$ 0.1 | 92.8 $\pm$ 0.1 |
| Conformal ECCO | 1.96 $\pm$ 0.06 | 4.32 $\pm$ 0.10 | 220.9 $\pm$ 8.1 | **90.0** $\pm$ 0.1 | **90.1** $\pm$ 0.1 | **90.0** $\pm$ 0.1 |
| PECCO | **1.39** $\pm$ .02 | **3.41** $\pm$ 0.03 | **8.52** $\pm$ 0.16 | 74.9 $\pm$ 0.2 | 93.6 $\pm$ 0.8 | 92.5 $\pm$ 0.9 |

*Table 2.* Comparison with conformal prediction methods on Argoverse dataset. PECCO produces a parametric distribution with a tighter confidence region (small MRS), whereby achieving better regression accuracy while maintaining competitive coverage.

### 6.1. Model Analysis

**Comparison with data augmentation and cannonicalization.** Data augmentation and cannonicalization are popular methods to implicitly exploit symmetry in trajectory data. In Table 3 we compare PECCO to its non-equivariant counterparts with the addition of data augmentation and cannonicalization. Equivariance dramatically improves the prediction performance in all aspects, especially calibration.

| | $\text{minA/FDE}_6$ | NLL | MRS | Cov(%) |
|---|---|---|---|---|
| CtsConv | 1.74 / 4.43 | 49.13 | 1498.4 | 2.2 |
| +cannon | 1.66 / 4.28 | 47.46 | 1246.9 | 4.5 |
| +aug | 1.67 / 4.23 | 11.81 | 857.0 | 17.1 |
| equivariant | **1.39 / 3.41** | **4.26** | **8.52** | **87.5** |

*Table 3.* Data Augmentation Comparison on the Argoverse Dataset

**Dynamics Integration Ablation.** Dynamic integration (dyna) introduced in Section 5.3 enforces the uncertainty to grow monotonically over time. As an ablative study, Table 4 show that PECCO with dynamic integration has much better calibration compared to if without; for the Argoverse dataset, the improved calibration also informs better performance on minADE/FDE.

**Comparison with Conformal Prediction.** Conformal prediction is an alternative uncertainty quantification method with theoretical guarantees. It does not impose any assumptions about the underlying distribution. We implemented two conformal prediction baselines based on (Stankevičiūtė et al., 2021) (Appendix B.2). We use half of the dataset for training and calibration respectively, and show performance in Table 2. Conformal methods achieve guaranteed $\geq 90\%$ coverage, but suffer in prediction accuracy due to less training data. The predicted regions are also much larger (higher MRS), which are less desirable in downstream tasks.

| **Argoverse** | $\text{minA/FDE}_6$ | NLL | MRS | Cov(%) |
|---|---|---|---|---|
| no-dyna | 1.52 / 3.76 | 9.72 | 39.1 | 38.6 |
| dyna | **1.39 / 3.41** | **4.26** | **8.36** | **87.5** |
| **Pedestrian** | $\text{minA/FDE}_6$ | NLL | MRS | Cov(%) |
| no-dyna | **0.72** / 2.12 | 4.71 | 24.6 | 39.6 |
| dyna | 0.73 / **1.98** | **2.37** | **6.12** | **83.7** |

*Table 4.* Dynamics Integration (dyna) Ablation: using dyna encourages the uncertainty to grow over time and improves coverage.

## 7. Conclusion

We propose Probabilistic Equivariant Continuous Convolution (PECCO), a novel method for improving uncertainty quantification for trajectory forecasting. We design an equivariant neural network under which the predicted distributions transform correspondingly as inputs are transformed. We introduce the MRS metric to bring attention to the calibration of probabilistic forecasts. By leveraging equivariance, PECCO produces more accurate and calibrated probabilistic forecasts compared to existing methods. A future direction of this work is to expand PECCO to produce multimodal forecasts. In this paper, we made the simplifying assumption that the uncertainties are 2-D normal distributions. By modifying PECCO to output a mixture of Gaussians, we can describe more complex trajectory distributions.

## References


Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Erkao Bao and Linqi Song. Equivariant neural networks and equivarification. *arXiv preprint arXiv:1906.07172*, 2019.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference, 2020.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90), 2020.

Monimoy Bujarbaruah, Xiaojing Zhang, Marko Tanaskovic, and Francesco Borrelli. Adaptive mpc under time varying uncertainty: Robust and stochastic. *arXiv preprint arXiv:1909.13473*, 2019.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477), 2007.

Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*. PMLR, 2017.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Timothy M Hall and Stephen Jewson. Statistical modelling of north atlantic tropical cyclone tracks. *Tellus A: Dynamic Meteorology and Oceanography*, 59(4), 2007.

Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Boris Ivanovic and Marco Pavone. Rethinking trajectory forecasting evaluation. *arXiv preprint arXiv:2107.10297*, 2021.

Michael I. Jordan, editor. *Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond*. The MIT Press, 1998.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 1960.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*. PMLR, 2020.

Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*. Springer, 2020a.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*. Springer, 2020b.

Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in Neural Information Processing Systems*, 2019.

Rachel Luo, Shengjia Zhao, Jonathan Kuck, Boris Ivanovic, Silvio Savarese, Edward Schmerling, and Marco Pavone. Sample-efficient safety assurances using conformal prediction. *arXiv preprint arXiv:2109.14082*, 2021.

Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1), 2016.

T Mitchell Roddenberry, Nicholas Glaze, and Santiago Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*. PMLR, 2021.

Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: a survey. *The International Journal of Robotics Research*, 39(8), 2020.

Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018.

Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*. Springer, 2020.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Bernd Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1, 2018.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.

Mlnoru Slotani. Tolerance regions for a multivariate normal population. *Annals of the Institute of Statistical Mathematics*, 16(1), 1964.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Kamilė Stankevičiūtė, Ahmed Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. *Advances in Neural Information Processing Systems*, 2019.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International Conference on Machine Learning*. PMLR, 2020.

Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2019.

Elise van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Robin Walters, Jinxi Li, and Rose Yu. Trajectory prediction using equivariant continuous convolution. *International Conference on Learning Representations*, 2021.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. *International Conference on Learning Representations*, 2021.

Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Dongxia Wu, Liyao Gao, Xinyue Xiong, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Quantifying uncertainty in deep spatiotemporal forecasting. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2021.

# A. Proofs

## A.1. Proofs for Equivariance of Gaussians [Proposition 2]

In this section, we prove that the Gaussian probability density function is $\mathrm{SO}(2)$-equivariant.

**Proposition.** *Given multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ over $\mathbb{R}^2$ with probability density function $p_{\mu,\Sigma}$ and $g \in \mathrm{SO}(2)$, then $p_{g\mu,g\Sigma g^T}(v) = p_{\mu,\Sigma}(g^{-1}v)$ for all $v \in \mathbb{R}^2$.*

*Proof.*

$$\begin{aligned} P_{g\mu,g\Sigma g^T}(v) &= \frac{1}{2\pi \det(g\Sigma g^T)} \exp\left(-\frac{1}{2}(v-g\mu)^T(g\Sigma g^T)^{-1}(v-g\mu)\right) \\ &= \frac{1}{2\pi \det(\Sigma)} \exp\left(-\frac{1}{2}(g^{-1}v-\mu)^T g^T g^{-T}\Sigma^{-1}g^{-1}g(g^{-1}v-\mu)\right) \\ &= \frac{1}{2\pi \det(\Sigma)} \exp\left(-\frac{1}{2}(g^{-1}v-\mu)^T\Sigma^{-1}(g^{-1}v-\mu)\right) \\ &= P_{\mu,\Sigma}(g^{-1}v) \end{aligned}$$

□

**Proposition.** *Given a normal distribution $\mathcal{N}(\mu, \Sigma)$, its group-actioned $\mathcal{N}(g\mu, g\Sigma g^T)$ is also a valid normal distribution.*

*Proof.* We prove the proposition by proving that $g\Sigma g^T$ is a symmetric positive definite matrix.

1) *(Symmetry)* $(g\Sigma g^T)^T = g^{TT}\Sigma^T g^T = g\Sigma g^T$.

2) *(Positive Definite)* Let $v \in \mathbb{R}^2$ and $v \neq 0$. Let $w = g^T v$, note that $w \neq 0$. Then $v^T g\Sigma g^T v = w^T \Sigma w > 0$, which implies $\Sigma$ is a positive definite matrix. □

## A.2. Equivariance of construct for positive definite and symmetric $\Sigma$ [Proposition 3]

Note that every positive definite symmetric matrix can be written as $\Sigma = MM^T$ for some $M \in GL_2(\mathbb{R})$, so we can represent $\Sigma$ by a map $\varphi$ where

$$\begin{aligned} \varphi\colon \mathrm{GL}_2(\mathbb{R}) &\to \mathrm{PosDefSym}_2(\mathbb{R}) \\ M &\mapsto MM^T \end{aligned}$$

**Proposition.** *$\varphi$ is $\mathrm{SO}(2)$-equivariant.*

*Proof.* $\varphi(gMg^T) = gMg^T(gMg^T)^T = gMg^T g^{TT} M^T g^T = gMM^T g^t = g\varphi(M)g^T$ □

We defined the one-sided inverse of $\varphi$ as $\psi$:

$$\begin{aligned} \psi\colon \mathrm{PosDefSym}_2(\mathbb{R}) &\to \mathrm{GL}_2(\mathbb{R}) \\ \Sigma &\mapsto Q\lambda^{\frac{1}{2}} \end{aligned}$$

where $Q\lambda Q^T$ is the eigendecomposition of $\Sigma$ and $Q$ is orthogonal. That is, $\varphi(\psi(\Sigma)) = \Sigma$.

**Proposition.** *$\psi$ is $\mathrm{SO}(2)$-equivariant.*

*Proof.* For symmetric positive definite matrix $\Sigma$, its eigenvector $v_i$ and eigenvalue $\lambda_i$ follows the eigenequation

$$\Sigma v_i = \lambda_i v_i.$$

For $\forall g \in \mathrm{SO}(2)$,

$$g\Sigma v_i = \lambda_i g v_i \implies g\Sigma g^{-1} g v_i = \lambda_i g v_i \implies (g\Sigma g^{-1})(g v_i) = \lambda_i (g v_i),$$

so $gv_i$ is an eigenvector to $g\Sigma g^{-1}$, with corresponding eigenvalue $\lambda_i$.

Therefore,

$$\psi(g\Sigma) = gQ\Lambda^{\frac{1}{2}} = g\psi(\Sigma).$$

□

### A.3. Proof of Proposition 1: The Equivariant Model Gives an Invariant Conditional Distribution.

We now prove Proposition 1.

**Proposition** (Proposition 1). *If the one-step probabilistic forecasting model $f_\theta$ is $G$-equivariant, then the conditional probability distribution $p_\theta(\mathbf{x}_{t+1:t+k}|\mathbf{x}_{1:t}, \mathbf{e})$ is invariant as in Equation 5.*

*Proof.* Since $\mathbf{x}_{1:t}$ are known, we define $\Sigma_{1:t} = \epsilon \mathrm{Id}$ for $\epsilon$ small approximating a delta distribution at $\mu_{1:t} = \mathbf{x}_{1:t}$. We then evaluate $\mu_{j+1}, \Sigma_{j+1} = f_\theta(\mu_{j-t+1:j}, \Sigma_{j-t+1:j}, \mathbf{e})$ autoregressively for $j = 1, \ldots, k$. Finally define

$$\begin{aligned} p_\theta(\mathbf{x}_{t+1:t+k}|\mathbf{x}_{1:t}, \mathbf{e}) &= \prod_{j=1}^{k} p_\theta(\mathbf{x}_{t+j}|\mathbf{x}_{j:t+j-1}, \mathbf{e}) \\ &= \prod_{j=1}^{k} p_{N(\mu_{t+j}, \Sigma_{t+j})}(\mathbf{x}_{t+j}). \end{aligned} \tag{10}$$

For $g \in \mathrm{SO}(2)$, we want to show

$$p_\theta(g\mathbf{x}_{t+1:t+k}|g\mathbf{x}_{1:t}, g\mathbf{e}) = p_\theta(\mathbf{x}_{t+1:t+k}|\mathbf{x}_{1:t}, \mathbf{e}).$$

Thus for $g\mathbf{x}_{1:t}, g\mathbf{e}$ we initialize $g\mu_{1:t}, g\Sigma_{1:t}g^T$ since $g\Sigma_{1:t}g^T = gg^T\Sigma_{1:t} = \Sigma_{1:t}$. Applying $f_\theta$ repeatedly and invoking equivariance, we obtain

$$g\mu_{t+j}, g\Sigma_{t+j}g^T = f_\theta(g\mu_{j:t+j-1}, g\Sigma_{j:t+j-1}g^T, g\mathbf{e}). \tag{11}$$

for $j = 1, \ldots, k$. Then by Equation 10,

$$p_\theta(g\mathbf{x}_{t+1:t+k}|g\mathbf{x}_{1:t}, g\mathbf{e}) = \prod_{j=1}^{k} p_{N(g\mu_{t+j}, g\Sigma_{t+j}g^T)}(g\mathbf{x}_{t+j}).$$

Then by Proposition 2 this is

$$\prod_{j=1}^{k} p_{N(\mu_{t+j}, \Sigma_{t+j})}(g^{-1}g\mathbf{x}_{t+j}),$$

which cancels $g^{-1}g = 1$, and applying Equation 10 again gives $p_\theta(\mathbf{x}_{t+1:t+k}|\mathbf{x}_{1:t}\mathbf{e})$. □

### A.4. Proof that MRS is a strictly proper scoring rule

Scoring rules are summary measures to evaluate probabilistic forecasts; they take in the forecasted distribution and the event or value that materializes and assign a numerical score. As defined in (Gneiting and Raftery, 2007), a scoring rule is proper if it is maximized when the forecaster recovers the ground truth distribution. It is strictly proper if the maximum is unique.

**Proposition.** *MRS is a strictly proper score.*

*Proof.* For arbitrary function $h$, we first define the scoring rule

$$S(R; z) = -|R| + \frac{1}{\alpha}(|R| - |R(z)|)\mathbb{1}\{z \in R^C\} + h(z),$$

which means if the forecaster quotes region $R$ at the level $\alpha \in (0,1)$ and $z$ materializes, then the score $S(R;z)$ will be rewarded, where $R$ and $R(z)$ are defined as ellipsoid parametric by $\mu$ and $\Sigma$. Then the expected score under the probability measure $P \in \mathcal{P}$ which $z$ follows is defined as

$$S(R;P) = \int S(R;z)\, dP(z).$$

For $P \in \mathcal{P}$, let $R_*$ denote the unique true $P$-region at level $\alpha$. We say that scoring rule $S$ is strictly proper if

$$S(R_*;P) \geq S(R;P),$$

for all region $R$ and for all probability measures $P \in \mathcal{P}$, where equality holds if and only if $R = R_*$.

We identify $P$ with the associated distribution function so that $P(R_*^C) = \alpha$. If $R_* \subset R$, then

$$\begin{aligned}
S(R_*;P) - S(R;P) &= \int -|R_*|\, dP(z) + \frac{1}{\alpha}\int_{R_*^C} (|R_*| - |R(z)|)\, dP(z) \\
&\quad + \int |R|\, dP(z) - \frac{1}{\alpha}\int_{R^C} (|R| - |R(z)|)\, dP(z) \\
&= -|R_*| + |R| + \frac{1}{\alpha}|R_*|P(R_*^C) - \frac{1}{\alpha}|R|P(R^C) - \frac{1}{\alpha}\int_{R_*^C \setminus R^C} |R(z)|\, dP(z) \\
&\geq |R| - \frac{1}{\alpha}|R|P(R^C) - \frac{1}{\alpha}|R|(P(R_*^C) - P(R^C)) \\
&= 0,
\end{aligned}$$

as it is supposed to be. If $R \subset R_*$, then an analogous argument applies.

Putting $h(z) = 0$ and reversing the sign of the scoring rule, yields the negatively oriented regional score. Finally, the mean score of this oriented regional score is our MRS, which is then proved to be proper.

□

## B. Algorithms

### B.1. Empirical method for estimating MRS for general 2D distribution

We present an empirical method to estimate MRS given any 2D distribution. We first discretize the 2D space that is the domain of the probability distribution into $n$ grid cells, indexing them as $g_i$ where $i \in \{1, \ldots, j\}$. Then we can numerically estimate the density of each grid cell $p(g_i)$ for $i = 1, \ldots, n$. The confidence region can be estimated by selecting the grid cells with the highest likelihoods until they sum to $1 - \alpha$.

$$R(1-\alpha) = \bigcup_{i \in K} g_i, \quad K = \text{argmin}_K : \Sigma_{i \in K} p(g_i) \geq 1 - \alpha$$

where $K$ is the superlevel set of grid cells $g_i$ whose empirical probability density is at least $1 - \alpha$.

Similarly, we define the estimated region

$$R(z_i) = \bigcup g_i \quad \text{for } p(g_i) \geq p(z_i)$$

as the superlevel set of data point $z_i$. This way, we can calculate MRS for any 2D using the estimated $R(1-\alpha)$ and $R(z_i)$ in the MRS formula.

$$\text{MRS}(R;Z) = \frac{1}{N}\sum_{i=1}^{N} [|R(1-\alpha)| + \frac{1}{\alpha}|R(z_i) \backslash R(1-\alpha)| \mathbb{1}\{z_i \in R^C(1-\alpha)\}]$$

Figure 5 illustrates the idea, this method is used to estimate the MRS scores for multimodal methods `Trajectron++` and `MFP` in our experiments.

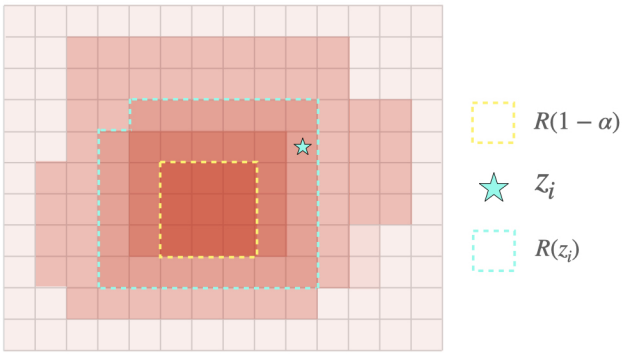



*Figure 5.* Illustration for empirical MRS estimation, where we calculate the probability density at each grid, and select the highest density grids to construct $R(z_i)$ and $R(1-\alpha)$. It is

### B.2. Conformal algorithm for 2-D Time Series

We extend the inductive conformal prediction (ICP) algorithm (Shafer and Vovk, 2008) to conformally estimate a confidence region for 2D forecasts. Given an exchangeable set of trajectory observations $\mathcal{D} = \{(\mathbf{x}_{1:t}^{(i)}, \mathbf{e}^{(i)}), \mathbf{x}_{t+1:t+k}^{(i)}\}_{i=1}^{l}$ and a new sample $(\mathbf{x}_{1:t}^{(l+1)}, \mathbf{e}^{(l+1)})$, the ICP algorithm returns $k$ prediction intervals,$[\Gamma_1^\alpha, \ldots, \Gamma_k^\alpha]$, one for each timestep, such that:

$$\mathbb{P}[\,\forall j \in \{1, \ldots, k\},\ x_{t+j} \in \Gamma_j^\alpha(\mathbf{x}^{(l+1)})\,] \geq 1 - \alpha$$

for any underlying predictive model. This inequality is the *validity* condition. (Stankevičiūtė et al., 2021) proves the 1-D case for time-series forecasting validity; we leave proving it for the 2-D case for future work.

ICP requires an underlying forecasting model and a choice of nonconformity score. In our case, LSTM and ECCO are both models that take as input $(\mathbf{x}_{1:t}, \mathbf{e})$ and outputs $\mathbf{x}_{t+1:t+k}$. We select root mean square error (RMSE) as our nonconformity score as it is commonly used in the 1-d setting and naturally extends to high dimensions. We split the training set into the proper training set and a calibration set of equal size: $\mathcal{D} = \mathcal{D}_{train} \bigcup \mathcal{D}_{cal}$. We train our model $M$ on $\mathcal{D}_{train}$ and obtain the critical nonconformity scores $\hat{\gamma}_1, \ldots, \hat{\gamma}_k$ as in algorithm 1 in (Stankevičiūtė et al., 2021). Then, for input $(\mathbf{x}_{1:t}, \mathbf{e})$ and $\hat{x}_{t+1:t+k} = M(\mathbf{x}_{1:t}, \mathbf{e})$ we can construct a set of confidence intervals

$$\Gamma_j^\alpha(\mathbf{x}_{(1:t)}^{(l+1)}) = \{x \in \mathbb{R}^2 \mid RMSE(x, \hat{x}_j) \leq \hat{\gamma}_j\} \quad \text{for } j \in \{1, \ldots, k\}$$

We illustrate some example scenes with confidence regions provided by conformal-ECCO in figure 6. Note that the regions for timestep $j$ are given by $\hat{\gamma}_j$, hence are the same for every scenario. With our choice of RMSE as the nonconformity score, the region is a circle; they appear elliptical due to scale compression.

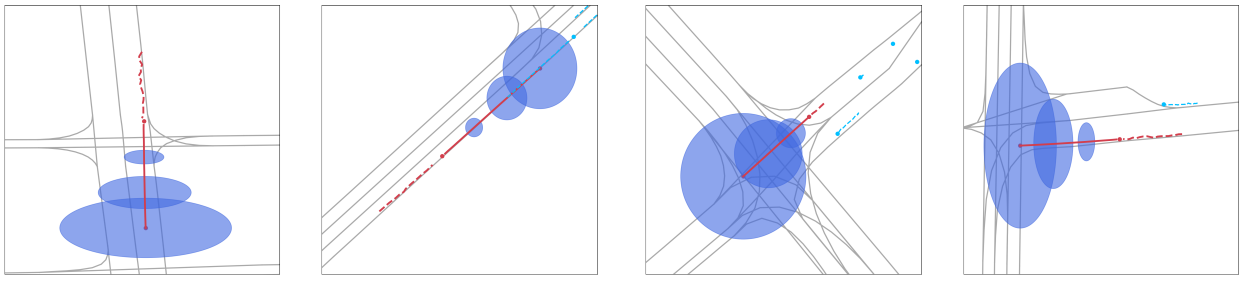


*Figure 6.* Scene illustrations for confidence regions given by conformal-ECCO. Note that in order to achieve 90% coverage, the region is larger than needed, especially in straight-lane cases like left two scenarios. Due to the lack of underlying assumption about the distribution, the model isn't able to adjust the shape of uncertainty according to data either.

# C. Experiment Details

## C.1. Implementation Details

**Dataset and Preprocessing** The Argoverse autonomous vehicle dataset contains 205,942 samples, consisting of diverse driving scenarios from Miami and Pittsburgh. We split 90/10 into a training set and validation set of size 185,348 and 20,594 respectively. The official validation set of size 39,472 is used for testing and reporting performance. We preprocess the scenes to filter out incomplete trajectories and cap the number of vehicles modeled to 60. If there are less than 60 cars in the scenario, we insert dummy cars into them to achieve consistent car numbers. For map information, we only include center lanes with lane directions as features. Similar to vehicles, we introduce dummy lane nodes into each scene to make lane numbers consistently equal to 650.

We use the latest release of the TrajNet++ dataset (Update 4.0) for our pedestrian experiments. TrajNet++ is a compiled set of pedestrian trajectories captured in both indoor and outdoor locations such as in universities, hotels, Zara, and train stations. The sample in this dataset is 21 timestamps long, and the goal is to predict the 2D spatial positions for each pedestrian in the future 12 timestamps given the first 9. The pedestrian dataset contains 240,896 samples, which we split 80/10/10 into train, validation, and test sets. Similar to Argoverse, we filter out incomplete trajectories in processing and either cap or insert dummy pedestrians such that each scene has 60 agents. No map information was used in the pedestrian dataset.

We include our code for preprocessing, model implementation, and training in the supplementary materials.

**Hyperparameters and Training Details** We trained the PECCO model with 4 equivariant continuous convolution layers of hidden size of (8, 16, 16, 16) respectively for Argoverse, and (4, 8, 16, 16) for the Trajnet++ pedestrian dataset. Our models are all trained with Adam optimizer with a base learning rate $r = 0.001$, and linear learning rate scheduler set to $\gamma = 0.95$. For Argoverse task, we set the CtsConv radius to be 40, and for the pedestrian task we set it to be 6. PECCO's Arogoverse model has 129k parameters. For comparison, Trajectron++ has 127k and MFP has 67K.

All our models without map information are trained for 10K iterations with batch size 32 with learning rate updating every 150 iterations. Most of our experiments are performed on a server with 4 RTX 2080 Ti GPUs, and it takes around 9-12 hours to finish training. We run each experiment 3 times with different random initialization and data order. The numbers reported in tables 1 2, and 5 are the mean and standard deviation of those 3 runs.

## C.2. Deterministic Baseline Results

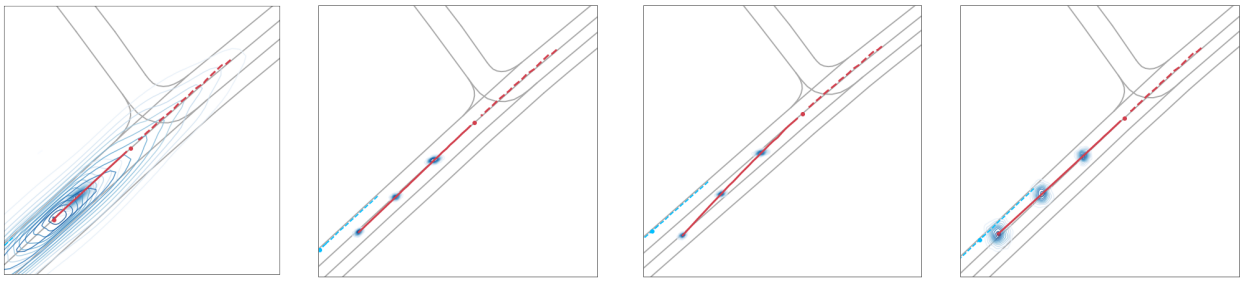
We present numbers for deterministic baseline models for Argoverse and Trajnet++. By sampling, we achieve better minADE/minFDE performance with probabilistic models, but they serve as valuable baselines to illustrate the difficulty of the task.

| | Argoverse | | Trajnet++ | |
|---|---|---|---|---|
| **Model** | ADE↓ | FDE↓ | ADE↓ | FDE↓ |
| Constant Velocity | 2.77 | 6.16 | 1.21 | 2.37 |
| Nearest Neighbor | 3.52 | 7.85 | 1.25 | 2.61 |
| LSTM | 1.97 ± .05 | 4.98 ± .31 | 1.01 ± .02 | 1.98 ± .08 |
| CtsConv | 1.87 ± .06 | 4.43 ± .28 | 1.35 ± .05 | 2.97 ± .16 |
| ECCO | 1.68 ± .04 | 3.98 ± .19 | 0.94 ± .01 | 2.05 ± .03 |

*Table 5.* Deterministic baselines on the Argoverse and TrajNet++ dataset

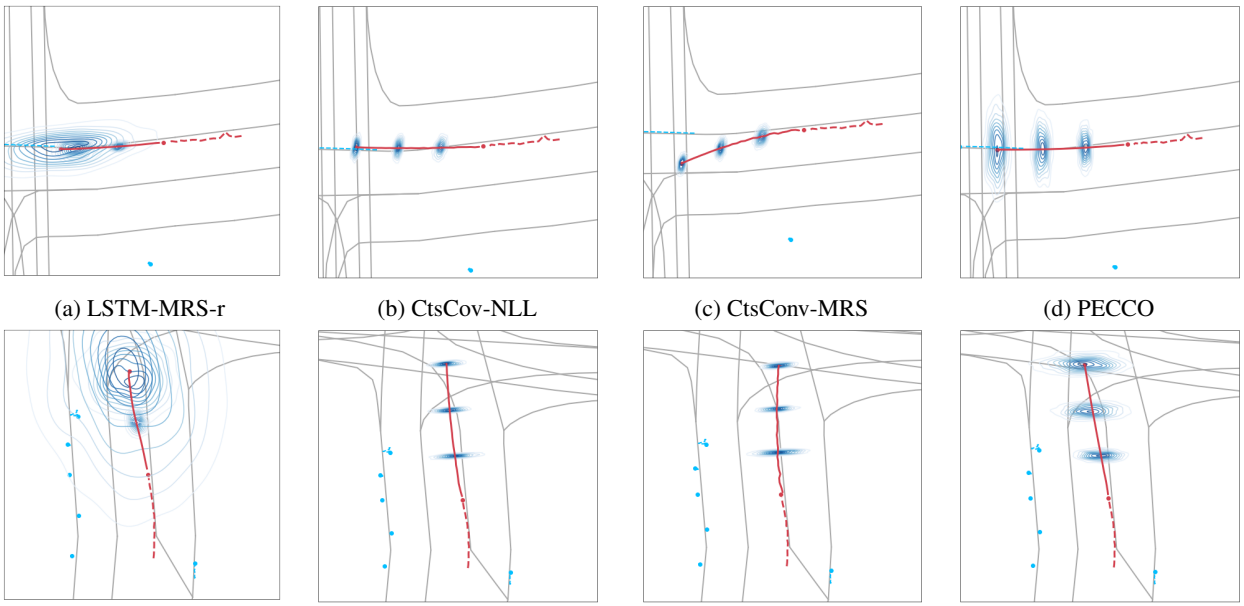
## C.3. Visualizations

We include 3 more scenarios for qualitatively comparison between models. They provide insights consistent with Figure 4: LSTM-MRS-r tends to overestimate uncertainty while CtsConv models tend to underestimate collapsing into one mode. PECCO, on the other hand, forecasts a tighter probability that covers the possibilities well.



(a) LSTM-MRS-r (b) CtsConv-NLL (c) CtsConv-MRS (d) PECCO

*Figure 7.* Another lane change example. Red trajectories is the agent of interest in the same scenario. CtsConv-NLL and CtsConv-MRS captured two different modes of futures, whereas PECCO is able to cover both.



(a) LSTM-MRS-r (b) CtsCov-NLL (c) CtsConv-MRS (d) PECCO

(e) LSTM-MRS-r (f) CtsConv-NLL (g) CtsConv-MRS (h) PECCO

*Figure 8.* Comparison of uncertainty predicted at 2 crossroads. Red trajectories is the agent of interest in the same scenario. Note how the LSTM predicted uncertainty explodes at the final time step, and PECCO is able to model the possibility of turns.