BiTraP: Bi-directional Pedestrian Trajectory Prediction with Multi-modal Goal Estimation

Yu Yao[1], Ella Atkins[2], Matthew Johnson-Roberson[3], Ram Vasudevan[4], and Xiaoxiao Du[3]

***Abstract*—Pedestrian trajectory prediction is an essential task in robotic applications such as autonomous driving and robot navigation. State-of-the-art trajectory predictors use a conditional variational autoencoder (CVAE) with recurrent neural networks (RNNs) to encode observed trajectories and decode multi-modal future trajectories. This process can suffer from accumulated errors over long prediction horizons (≥ 2 seconds). This paper presents *BiTraP*, a goal-conditioned bi-directional multi-modal trajectory prediction method based on the CVAE. BiTraP estimates the goal (end-point) of trajectories and introduces a novel bi-directional decoder to improve longer-term trajectory prediction accuracy. Extensive experiments show that BiTraP generalizes to both first-person view (FPV) and bird's-eye view (BEV) scenarios and outperforms state-of-the-art results by $\sim 10-50\%$. We also show that different choices of non-parametric versus parametric target models in the CVAE directly influence the predicted multi-modal trajectory distributions. These results provide guidance on trajectory predictor design for robotic applications such as collision avoidance and navigation systems. *Our code is available at: https://github.com/umautobots/bidireaction-trajectory-prediction*.**

***Index Terms*—Computer Vision for Automation, Human and Humanoid Motion Analysis and Synthesis, Deep Learning Methods, Multi-modal Trajectory Prediction, Goal-conditioned Prediction**

I. Introduction

UNDERSTANDING and predicting pedestrian movement behaviors is crucial for autonomous systems to safely navigate interactive environments. By correctly forecasting pedestrian trajectories, a robot can plan safe and socially-aware paths in traffic [1], [2], [3], [4] and produce alarms about anomalous motions (e.g., crashes or near collisions) [5], [6], [7], [8], [9]. Early work often assumed a deterministic future, where only one trajectory is predicted for each person given past observations [10], [11], [12]. However, pedestrians move with a high degree of stochasticity so multiple plausible and distinct future behaviors can exist [13], [14]. Recent studies [15], [16], [17], [18], [19], [20] have shown predicting a distribution of multiple potential future trajectories (i.e., multi-modal prediction) rather than a single best trajectory can more accurately model future motions of pedestrians.

Recurrent neural networks (RNNs), notably long short-term memory networks (LSTMs) and gated recurrent units (GRUs), have demonstrated success in trajectory prediction [2], [21], [22], [23]. However, existing models recurrently predict future trajectories based on previous output thus their performance tends to deteriorate rapidly over time ($>$ 560 ms) [14], [24]. We propose to address this problem with a novel goal-conditioned bi-directional trajectory predictor, named *BiTraP*. BiTraP first estimates future goals (end-points of the future trajectories) of pedestrians and then predicts trajectories by combining forward passing from current position and backward passing from estimated goals. Predicting goals can improve long-term trajectory predictions, as pedestrians in real world often have desired goals and plan paths to reach these goals [25]. Compared to existing goal-conditioned methods [25], [26], [27] where goals were used as an input to a forward decoder, BiTraP takes goals as starting positions of a backward decoder and predicts future trajectories from two directions, thus mitigating accumulated error over longer prediction horizons.

Recently, generative models such as the generative adversarial network (GAN) [13] and conditional variational autoencoder (CVAE) [28], [16], were developed to predict multi-modal distributions of future trajectories. Our BiTraP model predicts multi-modal trajectories based on CVAE which learns target future trajectory distributions conditioned on the observed past trajectories through a stochastic latent variable. The two most common forms of the latent variable follow either a Gaussian distribution or a categorical distribution, resulting in either a non-parametric target distribution [16], [25] or a parametric target distribution model such as a Gaussian Mixture Model (GMM) [19], [20]. There has been limited research on how latent variable distributions impact predicted multi-modal trajectories. To fill this gap, we conducted extensive comparison studies using two variations of our BiTraP method: a non-parametric model using Gaussian latent variables (BiTraP-NP) and a GMM model using categorical latent variables (BiTraP-GMM). We implemented two types of loss functions, best-of-many (BoM) L2 loss [29] and negative log-likelihood (NLL) loss [20] to evaluate different predicted trajectory behaviors (e.g., spread and diversity). We show that latent variable distribution choices are closely related to the diversity of predicted distributions, which provides guidance for selecting trajectory predictors for robot navigation and

Manuscript received: October 15, 2020; Revised December 28, 2020; Accepted January 25, 2021.

This paper was recommended for publication by Editor Cesar Cadena upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by a grant from Ford Motor Company via the Ford-UM Alliance under award N028603. This material is based upon work supported by the Federal Highway Administration under contract number 693JJ319000009. Any options, findings, and conclusions or recommendations expressed in the this publication are those of the author(s) and do not necessarily reflect the views of the Federal Highway Administration.

[1]Y. Yao is with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109 USA brianyao@umich.edu

[2]E. Atkins is with the Aerospace Engineering Department, University of Michigan, Ann Arbor, MI 48109 USA ematkins@umich.edu

[3]M. Johnson-Roberson and X. Du are with Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 USA mattjr@umich.edu; xiaodu@umich.edu

[4]R. Vasudevan is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA ramv@umich.edu

collision avoidance systems.

The contributions of this work are summarized as follows. First, we developed a novel bi-directional trajectory predictor, *BiTraP*, with a multi-modal goal estimation module and a bi-directional decoder network, and show it offers significant improvements on trajectory prediction performance especially for longer (≥ 2 seconds) prediction horizons. Second, we studied parametric versus non-parametric target modeling methods by presenting two variations of our model, BiTraP-NP and BiTraP-GMM, and compare their influence on the diversity of predicted distribution. Extensive experiments with both first person and bird's eye view datasets show the effectiveness of BiTraP models in different domains.

II. Related Work

Our BiTraP model consists of two parts: a multi-modal goal estimator and a goal-conditioned bi-directional trajectory predictor. This section describes related work in multi-modal trajectory prediction and goal-conditioned prediction.

CVAE-based Approaches for Multi-modal Trajectory Prediction. Probabilistic approaches, particularly conditional variational autoencoder (CVAE) based models, have been developed for multi-modal trajectory prediction. Different from GANs [13], [30], CVAEs can explicitly learn the form of a target distribution conditioned on past observations by learning the latent distribution from which it samples. Some CVAE methods assume the target trajectory follows a non-parametric (NP) distribution and produces multi-modal predictions by sampling from a Gaussian latent space. Lee *et al*. [16] first used CVAE for multi-modal trajectory prediction by incorporating Gaussian latent space sampling to an long short-term memory encoder-decoder (LSTM-ED) model. However, reference [16] samples from a zero-mean-unit-variance Gaussian distribution during inference, which does not describe input-target correlation. CVAE with LSTM components has since been used in many applications [31], [32], [33]. Other CVAE-based methods assume parametric trajectory distributions. Ivanovic *et al*. [19] assumed the target trajectory follows a Gaussian Mixture Model (GMM) and designed a Trajectron network to predict GMM parameters using a spatio-temporal graph. Trajectron++ [20] extended Trajectron to account for dynamics and heterogeneous input data. Our work extends existing CVAE models with goal estimation and develops a novel bi-directional decoder network to mitigate long-term accumulated errors. Our work also provides novel insights in comparisons between CVAE target distributions (NP and GMM).

Trajectory Conditioned on Goals. Incorporating goals has been shown to improve trajectory prediction. Rehder *et al*. [26] and Huang *et al*. [34] proposed a particle-filter based method to estimate a goal as a prior for trajectory prediction. The predicted goal was used in [34] to create nominal trajectories, and the trajectory offset at each way point was predicted with a single-directional LSTM. Xue *et al*. [35] used a bi-directional LSTM encoder to classify observed pedestrian trajectories as several goal regions, and then predicted trajectory with a single-directional LSTM. Wu *et al*. [36] proposed a similar idea with neighbor features incorporated. Rhinehart *et al*. [27] estimated multi-modal semantic action as goals and planned conditioned trajectories using imitative models. Deo *et al*. [37] used IRL to estimate goal states and fused results with past trajectory encodings to generate predictions. We drew inspiration from [38], which computed forward and backward rewards based on current position and goal; the path was planned using Inverse Reinforcement Learning (IRL). Our method is distinct due to its novel bi-directional decoding and integration combined with a CVAE to achieve multi-modal prediction. Most recently, Mangalam *et al*. [25] designed a PECNet which showed state-of-the-art results on BEV trajectory prediction datasets. However, PECNet only concatenated past trajectory encodings and end-point encodings, which we believe did not fully take advantage of goal information. We have designed a bi-directional trajectory decoder in which current trajectory information is passed forward to the end-points (goals) and goals are recurrently propagated back to the current position. Experiment results show that our goal estimation can help generate more accurate trajectories.

III. BiTraP: Bi-directional Trajectory Prediction with Goal Estimation

Our BiTraP model performs goal-conditioned multi-modal bi-directional trajectory prediction in either first-person view (FPV) or bird's eye view (BEV). Let $\mathbf{X}_t = [X_{t-\tau+1}, X_{t-\tau+2}, ..., X_t]$ denote observed past trajectory at time t, where X_t is bounding box location and size (x, y, w, h) in pixels for FPV [22], [23] and (x, y) position in meters for BEV [20]. Given $\mathbf{X}_t$, we first estimate goal G_t of the person then predict future trajectory $\mathbf{Y}_t = [Y_{t+1}, Y_{t+2}, ..., Y_{t+\delta}]$, where τ and δ are observation and prediction horizons, respectively. Define goal $G_t = Y_{t+\delta}$ as the future trajectory endpoint, which is given in training and unknown in testing. We adopt a CVAE model to realize multi-modal goal and trajectory prediction. BiTraP contains four sub-modules: conditional prior network $p_\theta(Z|\mathbf{X}_t)$ to model latent variable Z from observations, recognition network $q_\phi(Z|\mathbf{X}_t, \mathbf{Y}_t)$ to capture dependencies between Z and $\mathbf{Y}_t$, goal generation network $p_\omega(G_t|\mathbf{X}_t, Z)$, and trajectory generation network $p_\psi(\mathbf{Y}_t|\mathbf{X}_t, G_t, Z)$ where ϕ, θ, ω and ψ represent network parameters. Either parametric or non-parametric models can be used to design networks p_ψ and p_ω for CVAE. Non-parametric models do not assume the distribution format of target $\mathbf{Y}_t$ but learn it implicitly by learning the distribution of Z. Parametric models assume a known distribution format for $\mathbf{Y}_t$ and predict distribution parameters. We design non-parametric and parametric models in Sections III-A and III-B, and explain different loss functions to train these models in Sections III-C and III-D.

A. BiTraP with Non-parametric (NP) Distribution

BiTraP-NP is built on a standard recurrent neural network encoder-decoder (RNN-ED) based CVAE trajectory predictor as in [16], [25], [29], [32], except it predicts goal first and then predict trajectories leveraging goals. Following previous work, we assume Gaussian latent variable $Z \sim \mathcal{N}(\mu_Z, \sigma_Z)$

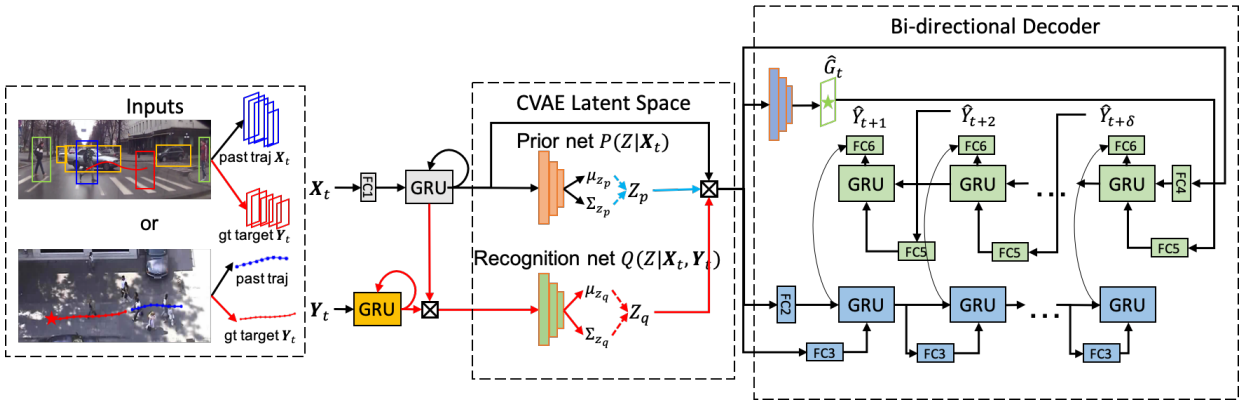


Fig. 1: Overview of our BiTraP-NP network. Red, blue and black arrows show processes that appear in training only, inference only, and both training and inference, respectively. BiTraP-NP is distinct from previous methods in its combination of goal estimator and bi-directional decoder.

and a non-parametric target distribution format. Fig. 1 shows the network architecture of BiTraP-NP.

Encoder and goal estimation. First, observed trajectory $\mathbf{X}_t$ is processed by a gated-recurrent unit (GRU) encoder network to obtain encoded feature vector h_t. In training, ground truth target $\mathbf{Y}_t$ is encoded by another GRU yielding h_{Y_t}. Recognition network $q_\phi(Z|\mathbf{X}_t, \mathbf{Y}_t)$ takes h_t and h_{Y_t} to predict distribution mean μ_{Z_q} and covariance Σ_{Z_q} which capture dependencies between observation and ground truth target. Prior network $p_\theta(Z|\mathbf{X}_t)$ assumes no knowledge about target and predicts μ_{Z_p} and Σ_{Z_p} using h_t only. Kullback–Leibler divergence (KLD) loss between $\mathcal{N}(\mu_{Z_p}, \Sigma_{Z_p})$ and $\mathcal{N}(\mu_{Z_q}, \Sigma_{Z_q})$ is optimized so that dependency between $\mathbf{Y}_t$ and $\mathbf{X}_t$ is implicitly learned by the prior network. Latent variable Z is sampled from $\mathcal{N}(\mu_{Z_q}, \Sigma_{Z_q})$ and concatenated with h_t to predict multi-modal goals $\hat{G}_t$ with goal generation network $p_\omega(G_t|\mathbf{X}_t, Z)$. In testing, we directly draw multiple samples from $\mathcal{N}(\mu_{Z_p}, \Sigma_{Z_p})$ and concatenate h_t to predict estimated goals $\hat{G}_t$. We use 3-layer multi-layer perceptrons (MLPs) for prior, recognition and goal generation networks.

Trajectory Decoder. Predicted goals $\hat{G}_t$ are used as inputs to a bi-directional trajectory generation network $p_\psi(\mathbf{Y}_t|\mathbf{X}_t, \hat{G}_t, Z)$, the trajectory decoder, to predict multi-modal trajectories. BiTraP's decoder contains forward and backward RNNs. The forward RNN is similar to a regular RNN decoder (Eq. (1)) except its output is not transformed to trajectory space. The backward RNN is initialized from encoder hidden state h_t. It takes estimated goal $\hat{Y}_{t+\delta} = \hat{G}_t$ as the initial input (Eq. (2)) and propagates from time $t+\delta$ to $t+1$ so backward hidden state is updated from the goal to the current location. Forward and backward hidden states for the same time step are concatenated to predict the final trajectory way-point at that time (Eq. (3)). These steps can be formulated as

$$h^f_{t+1} = GRU_f(h^f_t, W^i_f h^f_t + b^i_f), \tag{1}$$

$$h^b_{t+\delta-1} = GRU_b(h^b_{t+\delta}, W^i_b \hat{Y}_{t+\delta} + b^i_b), \tag{2}$$

$$\hat{Y}_{t+\delta-1} = W^o_f h^f_{t+\delta-1} + W^o_b h^b_{t+\delta-1} + b^o, \tag{3}$$

where, f, b, i and o indicate "forward", "backward", "input" and "output" respectively, and h^f_t and $h^b_{t+\delta}$ are initialized by passing h_t through two different fully-connected networks.

B. BiTraP with GMM Distribution

Parametric models predict trajectory distribution parameters instead of trajectory coordinates. BiTraP-GMM is our parametric variation of BiTraP assuming a GMM for the trajectory goal and at each way-point [19], [20]. Let $p(Y_{t+\delta})$ denote a K-component GMM at time step $t+\delta$. We assume $p(Y_{t+\delta}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(Y_{t+\delta}|\mu^i_{t+\delta}, \Sigma^i_{t+\delta})$, where each Gaussian component can be considered the distribution of one trajectory modality. Mixture component weights π_i sum to one, thus forming a categorical distribution. Each π_i indicates the probability (confidence) that a person's motion belongs to that modality. We design latent vector Z as a categorical (Cat) variable $Z \sim Cat(K, \pi_{1:K})$ parameterized by GMM component weights $\pi_{1:K}$ rather than separately-computed parameters. Similar to BiTraP-NP, we use three 3-layer MLPs for the prior, recognition and goal generation networks, and a bi-directional RNN decoder for the trajectory generation network. Instead of directly predicting trajectory coordinates, generation networks of BiTraP-GMM estimate the $\mu^i_{t+\delta}$ and $\Sigma^i_{t+\delta}$ of the ith Gaussian components at time $t+\delta$. In training, we sample one Z from each category to ensure all trajectory modalities are trained. In testing, we sample Z from $Cat(K, \pi_{1:K})$ so it is more probable to sample from high-confidence trajectory modalities.

C. Residual Prediction and BoM Loss for BiTraP-NP

Instead of directly predicting future location [23] or integrating from predicted future velocity [20], BiTraP-NP predicts change with respect to the current location based on residuals $\hat{Y}_{t+\delta} = Y_{t+\delta} - X_t$. There are two advantages of residual prediction. First, it assures the model will predict the trajectory starting from the current location, providing smaller initial loss than predicting location from scratch. Second, the residual target can be less noisy than the velocity target due to the fact that trajectory annotation is not always accurate. Standard CVAE loss includes NLL loss of the predicted distribution which is not applicable to NP methods due to their unknown

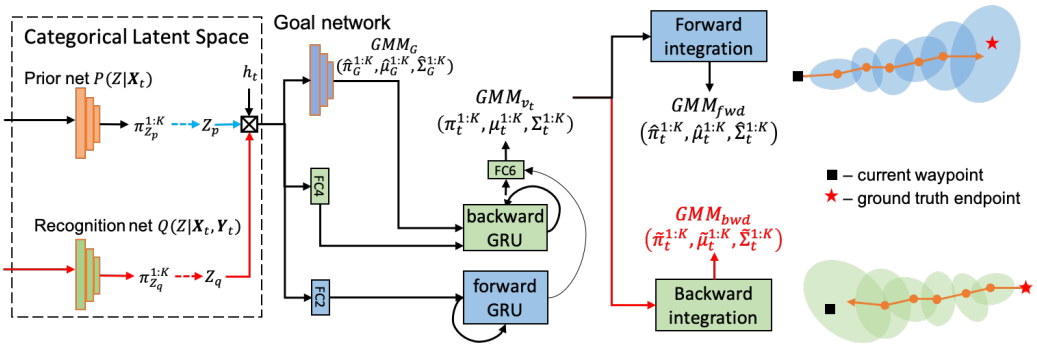


Fig. 2: Latent space sampling and decoder modules of BiTraP-GMM. The ellipse shows one of K GMM components at each timestep. The rest of the network is the same as BiTraP-NP in Fig. 1. BiTraP-GMM is distinct from previous methods with its goal estimator, bi-directional decoder and bi-directional integration process.

distribution format. L2 loss between predictions and targets can be used as a substitution [16]. To further encourage diversity in multi-modal prediction, we use best-of-many (BoM) L2 loss as in [29]. The final loss function for BiTraP-NP is a combination of the goal L2 loss, the trajectory L2 loss and the KL-divergence loss between prior and recognition networks, written as

$$L_{NP} = \min_{i \in N} \left\| G_t - X_t - \hat{G}_t^i \right\| + \min_{i \in N} \sum_{\tau=t+1}^{t+\delta} \left\| Y_\tau - X_t - \hat{Y}_\tau^i \right\| + KLD, \tag{4}$$

where $\hat{G}_t$ and $\hat{Y}_\tau$ are the predicted goal and trajectory waypoints with respect to current position X_t.

D. Bi-directional NLL Loss for BiTraP-GMM

Similar to [20], our BiTraP-GMM models the pedestrian velocity distribution as a GMM at each time step. The velocity GMM is then integrated forward to obtain the GMM distribution of trajectory waypoints $Y_{t+\delta}$ as shown by blue blocks in Fig. 2. We assume linear dynamics for pedestrian and use a single integrator as in Eq. (5). The loss function is then the summation of negative log-likelihood (NLL) of the ground truth future waypoints over the prediction horizon, formulated as

$$GMM_{Y_{t+\delta}}(\hat{\pi}_{t+\delta}^{1:K}, \hat{\mu}_{t+\delta}^{1:K}, \hat{\Sigma}_{t+\delta}^{1:K}) = X_t + \int_t^{t+\delta} GMM_{v_\tau}(\pi_\tau^{1:K}, \mu_\tau^{1:K}, \Sigma_\tau^{1:K}) d\tau, \tag{5}$$

$$NLL_{fwd} = \sum_{\tau=t}^{t+\delta} -\log p(Y_\tau | \hat{\pi}_\tau^{1:K}, \hat{\mu}_\tau^{1:K}, \hat{\Sigma}_\tau^{1:K}), \tag{6}$$

where $\pi_\tau^{1:K}$, $\mu_\tau^{1:K}$, $\Sigma_\tau^{1:K}$ are velocity GMM parameters at time $\tau \in [t+1, t+\delta]$, and the $\hat{\cdot}$ symbol indicates location GMM parameters obtained from integration. $p(\cdot)$ is the GMM probability density function. Such an NLL emphasizes earlier waypoints along the prediction horizon because a waypoint at time $t+1$ is used in integration results over $t+2$, $t+3$, ..., while these later waypoints are not used when computing $t+1$. This goes against our proposed idea which is to leverage a bi-directional temporal model. Therefore, we compute bi-directional NLL loss with reverse integration from the goal, formulated as

$$GMM'_{Y_t}(\tilde{\pi}_t^{1:K}, \tilde{\mu}_t^{1:K}, \tilde{\Sigma}_t^{1:K}) = G_t - \int_{t+\delta}^{t} GMM_{v_\tau}(\pi_\tau^{1:K}, \mu_\tau^{1:K}, \Sigma_\tau^{1:K}) d\tau, \tag{7}$$

$$NLL_{bwd} = \sum_{\tau=t+\delta} -\log p'(Y_\tau | \tilde{\pi}_\tau^{1:K}, \tilde{\mu}_\tau^{1:K}, \tilde{\Sigma}_\tau^{1:K}). \tag{8}$$

where $p(\cdot)'$ is the backward GMM probability density function, the $\tilde{\cdot}$ symbol indicates backward location GMM parameters. The final loss function for BiTraP-GMM can be written as

$$L_{GMM} = -\log p_G(G_t | \hat{\pi}_G^{1:K}, \hat{\mu}_G^{1:K}, \hat{\Sigma}_G^{1:K}) + NLL_{fwd} + NLL_{bwd} + KLD, \tag{9}$$

where the first term is NLL loss of the goal estimation, NLL_{fwd} and NLL_{bwd} are computed from forward and backward integration, the KLD term is the KL-divergence similar to Eq. (4).

IV. Experiments and Results

In this section, we empirically evaluate BiTraP-NP and BiTraP-GMM models on both first-person view (FPV) and bird's eye view (BEV) trajectory prediction datasets. We also provide a comparative study and discussion on the effects of model and loss selection.

Datasets. Two FPV datasets, Joint Attention for Autonomous Driving (JAAD) [39] and Pedestrian Intention Estimation (PIE) [23], and two benchmark BEV datasets, ETH [40] and UCY [41], were used in our experiments. JAAD contains 2,800 pedestrian trajectories captured from dash cameras annotated at 30Hz. PIE contains 1,800 pedestrian trajectories also annotated at 30Hz, with longer trajectories and more comprehensive annotations such as semantic intention, ego-motion and neighbor objects. ETH-UCY datasets contain five sub-datasets captured from down-facing surveillance cameras in four different scenes with 1,536 pedestrian trajectories annotated at 2.5Hz.

Implementation Details. For JAAD and PIE, we used the original training/testing splits with 0.5/1.5 second (15/45

frame) observation/prediction horizons as in [23]. For ETH-UCY, leave-one-out cross validation was applied, and the observation/prediction horizon was set to 3.2/4.8 seconds (*i.e.*, 8/12 frames) per [13], [20]. We follow these conventions to make fair comparisons between our results and those reported in previous papers. According to [14], [42], both 1.5 and 4.8 seconds can be considered long-term horizons. We used hidden size 256 for all encoders and decoders in BiTraP across all datasets. All models were trained with batch size 128, learning rate 0.001, and an exponential LR scheduler [20] on a single NVIDIA TITAN XP GPU.

A. Experiments on JAAD and PIE Datasets

Baselines. We compare our results against the following baseline models: 1) Linear Kalman filter, 2) Vanilla LSTM model, 3) Bayesian-LSTM model (B-LSTM) [43], 4) PIE$_{traj}$, an attentive RNN encoder-decoder model, 5) PIE$_{full}$, a multi-stream attentive RNN model, by injecting ego-motion and semantic intention stream to PIE$_{traj}$, and 6) FOL-X [22], a multi-stream RNN encoder-decoder model using residual prediction. We also conducted an ablation study for a deterministic variation of our model (BiTraP-D), where the multi-modal CVAE module was removed.

Evaluation Metrics. Following [22], [23], [43], our BiTraP model was evaluated using: 1) bounding box Average Displacement Error (ADE), 2) box center ADE (C_{ADE}) and 3) box center Final Displacement Error (C_{FDE}) in squared pixels. For our multi-modal BiTraP-NP and BiTraP-GMM, we compute the best-of-20 results (the minimum ADE and FDE from 20 randomly-sampled trajectories), following [13], [20], [44]. We also report the Kernel Density Estimation-based Negative Log Likelihood (KDE-NLL) metric for BiTraP-NP and BiTraP-GMM to evaluate the overall predicted distribution. KDE-NLL evaluates the NLL of the ground truth under a distribution fitted by a KDE on 2000 trajectory samples from each prediction model [20], [45]. For all metrics, lower values are better.

Results. Table I presents trajectory prediction results with JAAD and PIE datasets. Our deterministic BiTraP-D model shows consistently lower displacement errors across various prediction horizons than baseline methods such as PIE$_{traj}$ and FOL-X indicating our goal estimation and bi-directional prediction modules are effective. Our BiTraP-D model, based only on past trajectory information, also outperforms the state-of-the-art PIE$_{full}$, which requires additional ego-motion and semantic intention annotations. Table I also shows that non-parametric multi-modal method BiTraP-NP performs better on displacement metrics while parametric method BiTraP-GMM performs better on the NLL metric. This difference illustrates the objectives of these methods: BiTraP-NP generates diverse trajectories, and one trajectory was optimized to have minimum displacement error, while BiTraP-GMM generates trajectory distributions with more similarity to the ground truth trajectory.

Fig. 3 shows trajectory prediction results on sample frames from the PIE dataset. We observed that when a pedestrian intends to cross the street or change directions, the multi-modal BiTraP methods yield higher accuracy and more reasonable predictions than the deterministic variation. For example, as shown in Fig. 3(b), the deterministic BiTraP-D model (top row) can fail to predict the trajectory and the end-goal, where a pedestrian intends to cross the street in the future; the multi-modal BiTraP-NP model (bottom row) can successfully predict multiple possible future trajectories, including one where the pedestrian is crossing the street matching ground truth intention. Similar observations can be made in other frames. This result indicates multi-modal BiTraP-NP can predict multiple possible futures, which could help a mobile robot or a self-driving car safely yield to pedestrians. Although BiTraP-NP samples diverse trajectories, it still predicts distribution with high likelihood around ground truth targets and low likelihood in other locations as shown in Fig. 3(b)-3(d).

B. Experiments on ETH-UCY Datasets

Baselines. We compare our methods with five multi-modal baseline methods: S-GAN [13], SoPhie [44], S-BiGAT [30], PECNet [25] and Trajectron++ [20]. PECNet and Trajectron++ are most recent. PECNet is a goal-conditioned method using non-parametric distribution (thus directly comparable to our BiTraP-NP) while Trajectron++ uses a GMM trajectory distribution directly comparable to our BiTraP-GMM. Note that the baselines incorporated social information while our method focuses on investigating goal-based trajectory modeling and do not require extra input such as social or scene information.

Evaluation Metrics. Following [13], [25], [44], we used best-of-20 trajectory ADE and FDE in meters as evaluation metrics. We also report Average and Final KDE-NLL (ANLL and FNLL) metrics on 2000 sampled trajectories [20], [45] to evaluate the predicted trajectory and goal distribution.

Results. Table II shows the best-of-20 ADE/FDE results across all methods. We observed that BiTraP-NP outperforms the state-of-the-art goal based method (PECNet) by a large margin ($\sim 12\% - 51\%$), demonstrating the effectiveness of our bi-directional decoder module. BiTraP-NP also obtains lower ADE/FDE on most scenes ($\sim 12\%$-24% improvement) compared with Trajectron++. Our BiTraP-GMM model was trained using NLL loss, so it shows higher ADE/FDE results compared with BiTraP-NP. This is consistent with our FPV dataset observations in Section IV-A. Nevertheless, BiTraP-GMM still achieves similar or better results than PECNet and Trajectron++.

To further evaluate predicted trajectory distributions, we report KDE-NLL results in Table III. As shown, BiTraP-GMM outperforms Trajectron++ with lower ANLL and FNLL on *ETH*, *Univ*, *Zara1* and *Zara2* datasets. On *Hotel*, Trajectron++ achieves lower NLL values which may be due to the possible higher levels of inter-personal interactions than in other scenes. We observed improved ANLL/FNLL on *Hotel* (-1.88/0.27) when combining the BiTraP-GMM decoder with the interaction encoder in [20], consistent with our hypothesis.

Fig. 4 shows qualitative examples of our predicted trajectories using the BiTraP-NP and BiTraP-GMM models.

TABLE I: Results on JAAD and PIE datasets. The center row shows deterministic baselines including our ablation model BiTraP-D; the bottom row shows our proposed multi-modal methods. NLL is not available for deterministic methods since they predict single trajectories. Lower values are better.

Methods	JAAD				PIE			
	ADE (0.5/1.0/1.5s)	C_{ADE} (1.5s)	C_{FDE} (1.5s)	NLL	ADE (0.5/1.0/1.5s)	C_{ADE} (1.5s)	C_{FDE} (1.5s)	NLL
Linear [23]	233/857/2303	1565	6111	-	123/477/1365	950	3983	-
LSTM [23]	289/569/1558	1473	5766	-	172/330/911	837	3352	-
B-LSTM [43]	159/539/1535	1447	5615	-	101/296/855	811	3259	-
FOL-X [22]	147/484/1374	1290	4924	-	47/183/584	546	2303	-
PIE$_{traj}$ [23]	110/399/1280	1183	4780	-	58/200/636	596	2477	-
PIE$_{full}$ [23]	-	-	-	-	-/-/556	520	2162	-
BiTraP-D	**93/378/1206**	**1105**	**4565**	-	**41/161/511**	**481**	**1949**	-
BiTraP-NP (20)	**38/94/222**	**177**	**565**	18.9	**23/48/102**	**81**	**261**	16.5
BiTraP-GMM (20)	153/250/585	501	998	**16.0**	38/90/209	171	368	**13.8**

(a) (b) (c) (d)

Fig. 3: Qualitative results of deterministic (top row) vs multi-modal (bottom row) bi-directional prediction. Past (dark blue), ground truth future (red) and predicted future (green) trajectories and final bounding box locations are plotted. In the bottom row, each BiTraP-NP likelihood heatmap fits a KDE over samples. The orange color indicates higher probability.

TABLE II: Trajectory prediction results (ADE/FDE) on BEV ETH-UCY datasets. Lower is better.

Datasets	S-GAN [13]	SoPhie [44]	S-BiGAT [30]	PECNet [25]	Trajectron++ [20]	BiTraP-NP	BiTraP-GMM
ETH	0.81/1.52	0.70/1.43	0.69/1.29	0.54/0.87	0.43/0.86	**0.37/0.69**	0.40/0.74
Hotel	0.72/1.61	0.76/1.67	0.49/1.01	0.18/0.24	**0.12/0.19**	**0.12**/0.21	0.13/0.22
Univ	0.60/1.26	0.54/1.24	0.55/1.32	0.35/0.60	0.22/0.43	**0.17/0.37**	0.19/0.40
Zara1	0.34/0.69	0.30/0.63	0.30/0.62	0.22/0.39	0.17/0.32	**0.13/0.29**	0.14/**0.28**
Zara2	0.42/0.84	0.38/0.78	0.36/0.75	0.17/0.30	0.12/0.25	**0.10/0.21**	0.11/0.22
Average	0.58/1.18	0.54/1.15	0.48/1.00	0.29/0.48	0.21/0.39	**0.18/0.35**	0.19/0.37

TABLE III: Average-NLL/Final-NLL (ANLL/FNLL) results on ETH-UCY datasets. Lower is better.

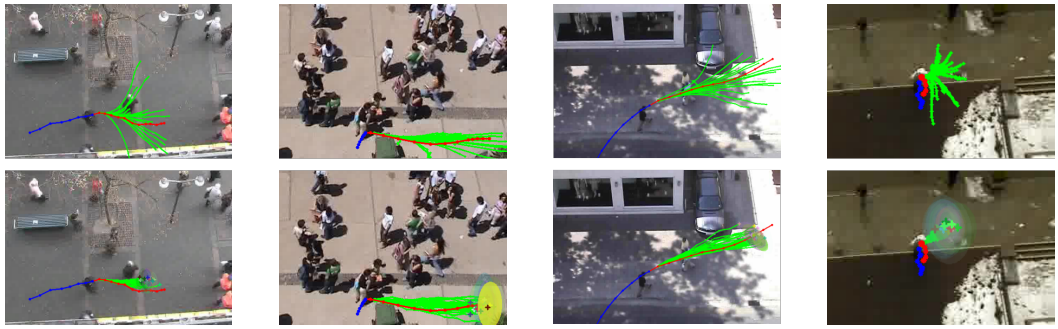
Datasets	S-GAN [13]	Trajectron++ [19]	BiTraP-NP	BiTraP-GMM
ETH	15.70/-	1.31/4.28	3.80/3.79	**0.96/3.55**
Hotel	8.10/-	**-1.94/0.25**	-0.41/1.26	-1.60/0.51
Univ	2.88/-	-1.13/2.13	-0.84/2.15	**-1.19/2.03**
Zara1	1.36/-	-1.41/1.83	-0.81/1.85	**-1.51/1.56**
Zara2	0.96/-	-2.53/0.50	-1.89/1.31	**-2.54/0.38**

As shown, BiTraP-NP (top row) generates future possible trajectories with a wider spread (more diverse), while BiTraP-GMM generates more compact distributions. This is consistent with our quantitative evaluations as reported in Table III, where the lower NLL results of BiTraP-GMM correspond to more compact trajectory distributions. Fig. 4(d) illustrates a challenging case where a pedestrian walks forward and then turns around. Predicting such a sudden "turn around" action would be difficult, resulting in a higher ADE of BiTraP-NP (0.72) and BiTraP-GMM (1.21) compared to average ADE (0.37 and 0.40) on ETH. Such challenging cases are related to uncertainty in pedestrian intention and behaviors and will be investigated in future work given more comprehensive behavior annotations.

We also computed KDE-NLL results for both Trajectron++ and BiTraP-GMM methods at each time step to analyze how BiTraP affects both short-term and longer-term (up to 4.8 seconds) prediction results. Per Fig. 5, BiTraP-GMM outperforms Trajectron++ with longer prediction horizons (after 1.2 seconds on *ETH*, *Univ*, *Zara1*, and *Zara2*). This shows the backward passing from the goal helps reduce error with longer prediction horizon.

C. Additional Experiments

Ablation study. We conducted two ablation experiments. To show bi-directional decoder effectiveness, we removed



(a) Hotel (b) Univ (c) Zara2 (d) ETH

Fig. 4: Visualizations of BiTraP-NP (first row) and BiTraP-GMM (second row). Twenty sampled future trajectories are plotted. For BiTraP-GMM, we also plot end-point GMM distributions as colored ellipses. Size indicates component Σ_k and transparency indicates component weight π_k.

TABLE IV: Ablation study results (ADE/FDE and ANLL/FNLL). Lower is better.

Method	BiTraP-NP				BiTraP-GMM			
	w/o backward (TraP-NP)		w/ backward		w/o bi-loss		w/ bi-loss	
	ADE/FDE	ANLL/FNLL	ADE/FDE	ANLL/FNLL	ADE/FDE	ANLL/FNLL	ADE/FDE	ANLL/FNLL
ETH	0.44/0.96	4.20/4.45	**0.37/0.69**	**3.80/3.79**	0.43/0.80	1.11/3.81	**0.40/0.74**	**0.96/3.55**
Hotel	0.13/0.23	-0.17/1.64	**0.12/0.21**	**-0.41/1.26**	0.16/0.25	-1.32/0.80	**0.13/0.22**	**-1.60/0.51**
Univ	0.21/0.43	-0.21/2.78	**0.17/0.37**	**-0.84/2.15**	0.20/0.41	-1.16/2.06	**0.19/0.40**	**-1.19/2.03**
Zara1	0.15/0.31	-0.37/2.27	**0.13/0.29**	**-0.81/1.85**	0.19/0.35	-0.90/2.12	**0.14/0.28**	**-1.51/1.56**
Zara2	0.12/0.23	-1.70/1.54	**0.10/0.21**	**-1.89/1.31**	0.13/0.25	-2.38/0.64	**0.11/0.22**	**-2.54/0.38**

TABLE V: Computational times with 20/2000 samples.

Method	Scene Graph	Model inference	Total
S-GAN[13]	N/A	103/10445 ms	103/10300 ms
Trajectron++[20]	11ms	55/58 ms	66/69 ms
TraP-NP	N/A	5.3/5.9 ms	5.3/5.9 ms
BiTraP-NP	N/A	8.3/9.1 ms	8.3/9.1 ms
BiTraP-GMM	N/A	69/72ms	69/72ms

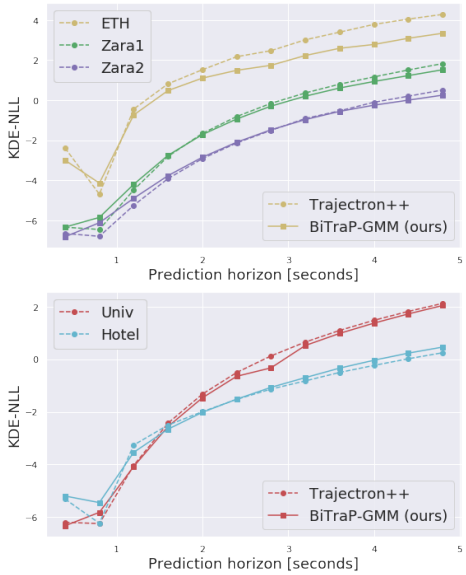


Fig. 5: KDE-NLL results on the ETH-UCY dataset per timestep up to 4.8 seconds.

the backward decoder from BiTraP-NP and compared its performance with the original BiTraP-NP model (w/o backward (TraP-NP) vs w/ backward). To show bi-directional loss effectiveness in BiTraP-GMM, we compared two BiTraP-GMM models trained with forward loss and bi-directional loss (w/o bi-loss vs w/ bi-loss). A comparison of ADE/FDE and ANLL/FNLL results is presented in Table IV. Using a bi-directional decoder (BiTraP-NP) improves ADE/FDE by 10%-28% (ANLL/FNLL by ∼0.4) from the model without backward decoder. By using bi-directional loss (bi-loss), the ADE/FDE of BiTraP-GMM model improves by 5-18% on ETH, and ANLL/FNLL improves by ∼0.25.

Computational time. We provide model inference time of Social GAN [13], Trajectron++ [20] and our BiTraP-NP and BiTraP-GMM models in Table V. Trajectron++ generates scene graphs before running the model so computation time is summed over scene graph generation and model inference. For Social GAN and our method, total time consists of model inference time only. We show computational times for number of samples 20 and 2000. Time differences of BiTraP models between the two numbers are ∼ 3ms, while the difference of S-GAN is extremely large as it generates samples one-by-one. BiTraP-GMM is ∼ 3ms slower than Trajectron++, not significant since both methods run at ∼ 70ms per frame (∼ 14 FPS) on average. BiTraP-NP is about 8x faster than Trajectron++ and BiTraP-GMM since it does not fit a GMM model or perform dynamic integration. Adding the bi-directional decoder slows inference by ∼ 3ms (TraP-NP vs BiTraP-NP). All experiments are conducted on the same machine used for training.

V. Conclusion

We presented *BiTraP*, a bi-directional multi-modal trajectory prediction method conditioned on goal estimation. We demonstrated that our proposed model can achieve state-of-the-art results for pedestrian trajectory prediction on both first-person view and bird's eye view datasets. The current BiTraP models, with only observed trajectories as inputs, already surpass previous methods which required additional ego-motion, semantic intention, and/or social information. By conducting a comparative study between non-parametric (BiTraP-NP) and parametric (BiTraP-GMM) models, we observed that the different latent variable choice affects the diversity of target distributions of future trajectories. We hypothesized that such difference in predicted distribution directly influences the collision rate in robot path planning and showed that collision metrics can be used to guide predictor selection in real world applications. For future work, we plan to incorporate scene semantics and social components to further boost the performance of each module. We are also interested in using predicted goals and trajectories to infer and interpret pedestrian intention and actions.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *CVPR*, 2016.

[2] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in *CVPR*, 2019.

[3] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2063–2073, 2014.

[4] N. Li, Y. Yao, I. Kolmanovsky, E. Atkins, and A. Girard, "Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections," *arXiv preprint arXiv:1904.05423*, 2019.

[5] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, "Learning regularity in skeleton trajectories for anomaly detection in videos," in *CVPR*, 2019.

[6] Y. Yao, M. Xu, Y. Wang, D. J. Crandall, and E. M. Atkins, "Unsupervised traffic accident detection in first-person videos," in *IROS*, 2019.

[7] Y. Yao, X. Wang, M. Xu, Z. Pu, E. Atkins, and D. Crandall, "When, where, and what? a new dataset for anomaly detection in driving videos," *arXiv preprint arXiv:2004.03044*, 2020.

[8] Y. Yao and E. Atkins, "The smart black box: A value-driven automotive event data recorder," in *ITSC*, 2018, pp. 973–978.

[9] ——, "The smart black box: A value-driven high-bandwidth automotive event data recorder," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[11] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[12] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[13] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018.

[14] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *ICCV*, 2015.

[15] H. O. Jacobs, O. K. Hughes, M. Johnson-Roberson, and R. Vasudevan, "Real-time certified probabilistic pedestrian forecasting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2064–2071, 2017.

[16] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *CVPR*, 2017.

[17] C. Anderson, X. Du, R. Vasudevan, and M. Johnson-Roberson, "Stochastic sampling simulation for pedestrian trajectory prediction," in *IROS*, 2019, pp. 4236–4243.

[18] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[19] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *ICCV*, 2019.

[20] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *arXiv preprint arXiv:2001.03093*, 2020.

[21] X. Du, R. Vasudevan, and M. Johnson-Roberson, "Bio-LSTM: a biomechanically inspired recurrent neural network for 3-D pedestrian pose and gait prediction," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1501–1508, 2019.

[22] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush, "Egocentric vision-based future vehicle localization for intelligent driving assistance systems," in *ICRA*, 2019, pp. 9711–9717.

[23] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *ICCV*, 2019.

[24] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *ICRA*, 2018, pp. 4563–4570.

[25] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," *arXiv preprint arXiv:2004.02025*, 2020.

[26] E. Rehder and H. Kloeden, "Goal-directed pedestrian prediction," in *ICCVW*, 2015.

[27] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *ICCV*, 2019.

[28] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NIPS*, 2015.

[29] A. Bhattacharyya, B. Schiele, and M. Fritz, "Accurate and diverse sampling of sequences based on a "best of many" sample objective," in *CVPR*, 2018.

[30] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *NIPS*, 2019.

[31] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *IV*, 2018.

[32] B. Ivanovic, E. Schmerling, K. Leung, and M. Pavone, "Generative modeling of multimodal multi-human behavior," in *IROS*, 2018.

[33] C. Choi, A. Patil, and S. Malla, "Drogon: A causal reasoning framework for future trajectory forecast," *arXiv preprint arXiv:1908.00024*, 2019.

[34] Z. Huang, A. Hasan, and K. Driggs-Campbell, "Intention-aware residual bidirectional lstm for long-term pedestrian trajectory prediction," *arXiv preprint arXiv:2007.00113*, 2020.

[35] H. Xue, D. Q. Huynh, and M. Reynolds, "Bi-prediction: pedestrian trajectory prediction based on bidirectional lstm classification," in *DICTA*, 2017.

[36] J. Wu, H. Woo, Y. Tamura, A. Moro, S. Massaroli, A. Yamashita, and H. Asama, "Pedestrian trajectory prediction using birnn encoder–decoder framework," *Advanced Robotics*, vol. 33, no. 18, pp. 956–969, 2019.

[37] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," *arXiv preprint arXiv:2001.00735*, 2020.

[38] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *ICRA*, 2018, pp. 5903–5908.

[39] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," *arXiv preprint arXiv:1609.04741*, 2016.

[40] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009.

[41] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *CVPR*, 2014.

[42] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2891–2900.

[43] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *CVPR*, 2018.

[44] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *CVPR*, 2019.

[45] L. A. Thiede and P. P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *ICCV*, 2019.

Supplementary File:
BiTraP: Bi-directional Pedestrian Trajectory Prediction with Multi-modal Goal Estimation

Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du

I. CVAE Preliminaries

A Conditional Variation Autoencoder (CVAE) is a conditional generative model designed to output target data Y based on latent variable Z and observation X [1]. A CVAE consists of three modules: a **conditional prior network** $p_\theta(Z|X)$ to model latent variable Z conditioned on observation X, a **recognition network** $q_\phi(Z|X,Y)$ to capture dependencies between Z and target Y, and a **generation network** $p_\psi(Y|X,Z)$ to generate the target Y, where ϕ, θ, and ψ represent network parameters. Stochastic latent variable $Z \in \mathbb{R}^d$ is sampled from a pre-defined distribution format such as a Gaussian distribution. The CVAE samples Z and generates target Y conditioned on observation X. The objective of a typical CVAE model is to maximize its variational lower bound

$$\begin{aligned}\max_{\theta,\phi,\psi} & \mathbb{E}_{q_\phi(Z|X,Y)}\Big[\log p_\psi(Y|X,Z)\Big] \\ & - KL\Big(q_\phi(Z|X,Y)||p_\theta(Z|X)\Big),\end{aligned} \tag{1}$$

where the first term maximizes the expectation of the log-likelihood of the target in the predicted distribution; the K-L (Kullback–Leibler) divergence term minimizes the difference between the recognition network and the conditional prior network. In this paper, we designed a modified CVAE with two generation networks and optimize both networks end-to-end.

II. ETH-UCY Dataset Variation Study

We present an analysis of the ETH-UCY dataset to show variation of the dataset as follows. We calculated statistics about pedestrian density, trajectory length, and velocity in the five subscenes, as shown in Table I below. This table is an extension of Table I in [2]. Note that *Zara1* and *Zara2* were collected from the same scene, and *Hotel* was collected from a scene visually similar to *Zara1/Zara2*. During cross-validation experiments, we observed that such scene similarity made the prediction results on Hotel, Zara1 and Zara2 datasets better because the training set contains similar scenes to the testing set. For example, a model trained with *ETH*, *Hotel*, *Univ* and *Zara1* can perform well when being tested with *Zara2*. An ETH scene contains longer trajectories and larger velocities compare to the other four scenes, making our prediction error higher on this dataset. *Univ* is a densely populated subscene collected on a university campus (on average $\sim$ 26.77 pedestrians per frame, as shown in Table I), but it has shorter trajectories and lower velocities so our methods still perform reasonably well on this dataset. Another observation from this table is that the pedestrian velocities are usually slower in scenes with larger numbers of pedestrians. Note in the second column to the right of Table I that the goal points are farther in ETH and Zara1 datasets than average (3.54 and 4.22 meters, respectively). The "future σ_l" values show the standard deviation of the distances between the goal and the current position, and a larger value indicates that some goal points are farther from the current position while others are close. This means the datasets contain diverse pedestrian motions; some are walking fast and others are wandering around or loitering. We have observed that goal prediction on pedestrians with wandering or loitering behaviors are more diverse because their motion intent is unclear, making it more difficult to predict where the pedestrian might go since they are currently not moving. On the other hand, goal prediction for pedestrians walking fast is clearer, since their motion intent can be more easily observed.

TABLE I: Mean and std (μ_n/σ_n) of pedestrian count per frame of each dataset. We also show the mean and std distance (μ_l/σ_l) and velocity (μ_v/σ_v) for observed trajectories (past 3.2 seconds) and future trajectories (future 4.8 seconds). The units for distance and velocity are m and m/s.

Datasets	μ_n/σ_n	Obs. μ_l/σ_l	Obs. μ_v/σ_v	Future μ_l/σ_l	Future μ_v/σ_v
ETH	6.15/4.46	2.85/2.44	1.04/0.88	3.54/3.19	0.87/0.80
Hotel	5.60/3.41	1.23/1.57	0.46/0.56	1.87/2.50	0.46/0.57
Univ	26.77/20.31	1.50/1.08	0.54/0.40	2.27/1.63	0.53/0.40
Zara1	5.91/3.21	2.70/1.09	0.97/0.40	4.22/1.64	0.97/0.38
Zara2	9.24/3.97	1.53/1.59	0.55/0.57	2.32/2.43	0.54/0.57

As for the First Person View (FPV) dataset, our method takes the detected bounding box sequence as input and relies on the accuracy of the object (in this case, human) detector. We followed previous work using the provided object bounding box sequences in the JAAD and PIE datasets for training and testing. Thus, impact from potential object occlusion was minimized in this work. We also trained our method with dropout functions added to the encoder network, adding robustness to potentially missed objects.

III. Robot Navigation Simulation Experiment Using BiTraP

To quantitatively analyze application of the BiTraP-GMM and BiTraP-NP models to robot navigation tasks, we designed a simulated robot navigation experiment based on the ETH-UCY bird's-eye view dataset. In this experiment, given predicted pedestrian trajectory distributions in a scene using

*{brianyao, ematkins, mattjr, ramv, xiaodu}@umich.edu

our BiTraP models and pre-planned paths for a robot, we show that we are able to compute the collision likelihood for each path, and thus are able to predict collision rate and select the safest path for the robot. Assuming a mobile robot navigates among pedestrians, we present results on two tasks: 1) Select the safest path for the robot and 2) Predict whether a path will collide with any other pedestrians in the scene. In this section, we first introduce our experiment setup. Then, we present evaluation results of our BiTraP models on path selection and collision prediction tasks.

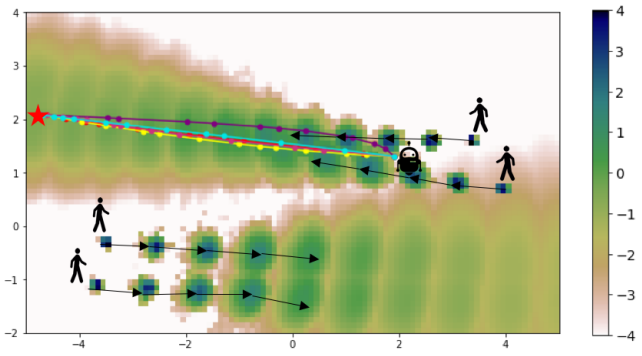


Fig. 1: Generation of Monte Carlo (MC) robot trajectories for collision detection experiments using Bezier curves. We illustrate five MC trajectory samples including start (robot icon) and end (red star) waypoints. Predicted trajectory distributions of neighbor pedestrians are plotted as a heat map; their walking directions are indicated by black arrows.

Experimental Setup. We selected all samples with more than one pedestrian in the test split [3] from ETH-UCY. Each sample has a node pedestrian (the pedestrian used for testing in previous work) and several neighbor pedestrians (the pedestrians used for social modeling in previous work) as in [3], [4]. We regard the node pedestrian as a "robot" navigating among other neighbor pedestrians. The starting and goal points of the "robot" are the same as the current position and goal point of the node pedestrian. A sample scene with one "robot" navigating among four other pedestrians is shown in Fig. 1. For the robot, 100 Monte Carlo (MC) paths were generated from start state to end point following quadratic and cubic Bezier curves [5]. Other more complex path planners could be used to generate additional experimental datasets. We assume the robot must reach the designated goal in 12 time steps, matching the prediction horizon for the pedestrian node in each scene. We uniformly generate waypoints along the path and randomly shift each by up to $\pm 50\%$ of the step length, resulting in a trajectory sequence containing 12 random waypoints. Other pedestrians follow their original (ground truth) trajectories in the scene. For each neighbor pedestrian, we run BiTraP-NP and BiTraP-GMM separately. Each method samples 2000 future trajectories to fit one Gaussian Kernel Density Estimation (KDE) model for each pedestrian as the predicted future distribution. Then, we compute the maximum KDE log-likelihood of all the waypoints on all robot MC paths and treat this log-likelihood value as a collision score. The higher the collision score, the more likely a collision will happen along this path. Given these collision scores, we compute the safest path collision rate (SPCR) as reported in *Task 1* below. Receiver operating characteristic (ROC) and precision-recall (P-R) curve results are reported in *Task 2*.

Task 1: Predict the Safest Path. We mark the robot MC path in each scene with minimum collision score as the "safest" (lowest collision likelihood) path. Then, we compute Euclidean distances between each safest path waypoint and other pedestrians' ground truth future trajectories. A collision is tallied if the minimum distance between a path and any pedestrians in the scene is less than 0.2 meters. Collision rate is computed as the number of paths with collision divided by the total number of safest paths. Due to the randomness in MC path generation, we conducted the simulation experiment five times with BiTraP-NP and BiTraP-GMM predictors separately and report collision rate mean (μ) and standard deviation (σ) values in Table II. As a comparison, we also present the collision rate of a randomly selected path among the 100 MC paths. The randomly selected paths do not have very high collision rates since the paths are planned based on pedestrian ground truth start and goal positions which are less likely to be involved in a collision. Compare to randomly selected paths, paths selected by our methods reduce the SPCR by a large margin. This shows that our predictors are effective for safest path selection. Both of our BiTraP methods achieve collision rate lower than 1% on *ETH*, *Hotel* and *Zara1* datasets. The *Univ* dataset is more difficult due to its high pedestrian density, and *Zara2* is most difficult because many pedestrian trajectories are quite close to each other. BiTraP-GMM shows lower SPCRs than BiTraP-NP on four datasets, indicating that it predicted more accurate (compared to ground truth) distributions. On *Zara1*, BiTraP-NP outperforms BiTraP-GMM by a small margin. BiTraP-NP ANLL and FNLL metric values as reported in the main paper are still higher than BiTraP-GMM values. A possible explanation is that BiTraP-NP predicts more diverse distributions thus detects some collisions not identified by BiTraP-GMM.

TABLE II: SPCR($\mu \pm \sigma$), AUC and AP results of our methods on ETH-UCY data group.

	Random from 100 (SPCR)	BiTraP-NP (SPCR/AUC/AP)	BiTraP-GMM (SPCR/AUC/AP)
ETH	$0.6 \pm 0.4\%$	$0.3 \pm 0.1\%$/ 92.3/ 24.2	$\mathbf{0.1 \pm 0.1\%}$/ **95.5**/ **26.0**
HOTEL	$0.4 \pm 0.3\%$	$0.1 \pm 0.1\%$/ 86.4/ 22.4	$\mathbf{0.0 \pm 0.0\%}$/ **91.6**/ **29.1**
Univ	$8.5 \pm 1.4\%$	$5.8 \pm 0.5\%$/ 81.0/ 33.4	$\mathbf{3.6 \pm 0.2\%}$/ **87.6**/ **43.4**
Zara1	$2.4 \pm 0.5\%$	$\mathbf{0.6 \pm 0.2\%}$/ 88.9/ 38.6	$0.8 \pm 0.3\%$/ **90.4**/ **41.6**
Zara2	$6.1 \pm 0.6\%$	$3.2 \pm 0.1\%$/ 81.0/ 44.0	$\mathbf{2.5 \pm 0.3\%}$/ **87.5**/ **52.6**

Task 2: Predict Collision for Any Path. The collision rate metric above only evaluates the safest path as selected by a trajectory predictor thus neglects all other paths. In the real-world, a trajectory predictor must be sufficiently accurate for the robot to accurately predict future collisions with high precision with a low missing rate (high true positive rate, TPR) and a low false alarm rate (low false positive rate,

Method	KDE NLL				FDE ML			
	@1s	@2s	@3s	@4s	@1s	@2s	@3s	@4s
Trajectron++ base [4]	-2.69	-2.46	-1.76	-1.09	0.03	0.17	0.37	0.60
Trajectron++ $\int$, map [4]	-5.58	-3.96	-2.77	-1.89	**0.01**	0.17	0.37	0.62
BiTraP-GMM (ours)	**-6.08**	**-4.21**	**-2.98**	**-2.05**	0.02	**0.15**	**0.35**	**0.58**

TABLE III: Pedestrian-only trajectory prediction results on nuScenes dataset.

FPR). To show the performance of BiTraP-NP and BiTraP-GMM predictors in terms of these metrics, we plotted the collision prediction ROC curve and P-R curve as follows. First, we collected all MC paths for the robot and tallied their collision scores. By setting a threshold γ, we can classify a path as collided (positive) or not collided (negative) and compute the TPR (i.e., recall), FPR and precision values. The ground truth label of each path is computed in the same way as before. By decreasing γ from a maximum value to minimum value (6 and -10 in this work), we plot the ROC and P-R curves shown in Fig. 2. The corresponding area under curve (AUC) and average precision (AP) are presented in Table II. In this work, AP is computed by equally spaced recall levels $\{1/40, 2/40,...,1\}$ following [6].

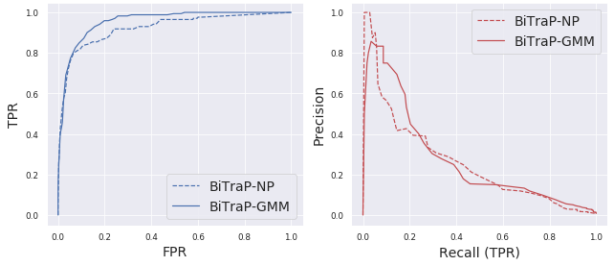
As shown in Fig. 2 and Table II, both BiTraP-NP and BiTraP-GMM methods achieve high AUCs (e.g., > 90 on *ETH*). Generally, BiTraP-GMM outperforms BiTraP-NP by a small margin in terms of both AUC and AP (e.g., 95.5 vs 92.3 AUC, and 26.0 vs 24.2 AP on *ETH*). Note that in real-world mobile robot applications missed collision detection (false negative) is unacceptable due to safety. That is to say, a high TPR (recall) is required. As can be observed in the higher TPR regions (x-axis) of the P-R curves, BiTraP-GMM outperforms BiTraP-NP on *ETH* (Fig. 2(a)) and *Hotel* (Fig. 2(b)), and both methods perform similarly on *Zara1* (Fig. 2(d)). On *Univ* (Fig. 2(c)) and *Zara2* (Fig. 2(e)), when the TPR is greater than a relatively high value (say 0.8), the FPR are higher (> 0.2) than in the other datasets, indicating increased chance of false alarms on these two datasets.

Compared to the ROC curve, the P-R curve is more suitable for imbalanced datasets due to the fact that it evaluates the fraction of true positives among positive predictions. This fits our case where the ratio of with-collision to no-collision paths is around 1:140, a large imbalance. On *Univ* and *Zara2* (Fig. 2(c) and 2(e)), BiTraP-GMM has higher precision than BiTraP-NP across almost all recall values. On the other hand, on *ETH*, *Hotel* and *Zara1* (Fig. 2(a) 2(b) and 2(d)), the two methods achieve similar precision at higher recall regions (e.g., when recall> 0.6). This is because when the threshold γ is too low, many paths are predicted as collided by both methods.

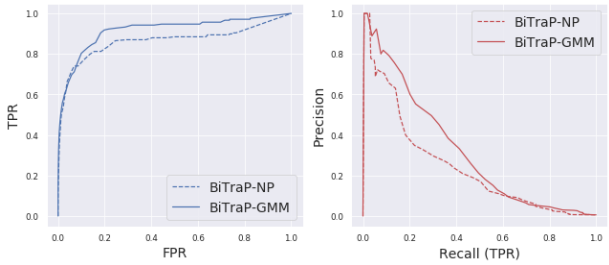
The ROC and P-R curves also verified our observation regarding the diversity of the predicted trajectory distribution as described in the main paper. At a fixed TPR on the ROC curves, we observe that BiTraP-NP always has a greater FPR than BiTraP-GMM, consistent with our hypothesis that BiTraP-NP predicts more diverse distributions, thus predicts more false alarms. Similarly, with fixed recall in P-R curves, BiTraP-NP has lower precision due the greater number of false alarms.

Discussion on BEV data. This section motivates our decision to design this experiment using Birds' Eye View (BEV) data, such as the ETH-UCY dataset. Many existing works on dynamics modeling and path planning are actually in Bird's-Eye View (BEV) scenarios [7], [8], [9], [10], [11], [12], [13], [14] because the physical distance between agents, objects, and goals can be more easily measured. In these papers, the robots either assumed accessibility to BEV sensors or created BEV maps by detecting the positions of surrounding objects using onboard sensors. The navigation problem was then solved in BEV. Although the robot sometimes only observes FPV image/point clouds, many methods still rely on creating a BEV world to perform navigation. For example, [9], [13] designed their navigation simulation in BEV, and [14] used the ETH-UCY dataset to test their navigator, the same dataset we used. Related BEV applications also include traffic monitoring, surveillance and security, and path planning for navigational purposes at cafes, shopping malls, and airport service robots [12], [14]. To this end, we concluded it was reasonable to conduct our experiments using BEV data.

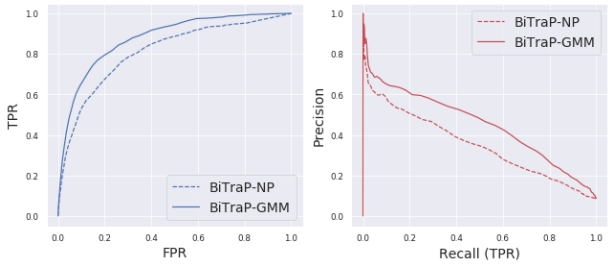
The purpose of this simulation experiment was to provide an alternative metric to evaluate and compare the performance of our two proposed predictor variations (*i.e.*, BiTraP-NP vs BiTraP-GMM). We also aimed to show that predictor performance can be used to compute collision rates and guide applications such as path selection. Conducting this simulation experiment in BEV was more feasible and reasonable for this purpose because our predictors can predict future trajectories of surrounding pedestrians and generate a cost map for the robot. It is also easier to visualize the paths in BEV, which is helpful in verifying consistent observations between main manuscript and supplementary results regarding the diversity/compactness of predicted trajectory distributions, *i.e.*, BiTraP-NP predicts more diverse distributions while BiTraP-GMM predicts more compact distributions. It is certainly possible to design a similar experiment using FPV dataset. In that case BEV data must be generated from FPV data if camera parameters are available to serve our experiment purpose anyway, so we decided to show results on BEV datasets directly. The goal of providing this supplementary experiment was to take a step at closing the gap between trajectory prediction research and navigation research by placing a robot equipped with our predictors into the environment and showing it is possible to compute and evaluate how much the collision rate is



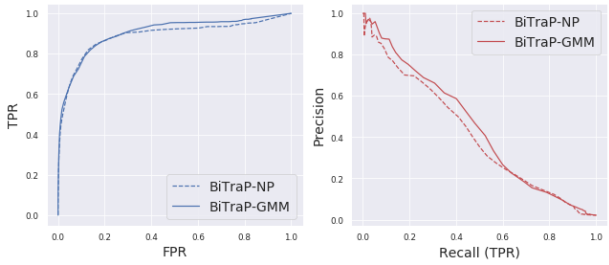
(a) ETH



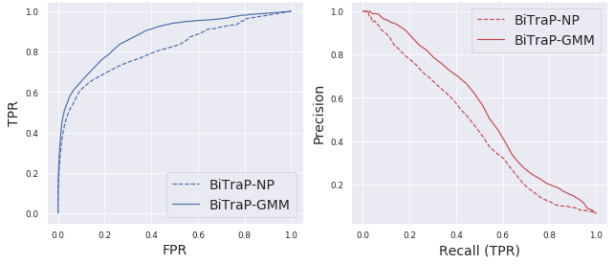
(b) Hotel



(c) Univ



(d) Zara1



(e) Zara2

Fig. 2: ROC (left) and P-R (right) curves of BiTraP-NP and BiTraP-GMM on ETH dataset.

reduced given different predictors. We hope this experiment will inspire future researchers to evaluate their trajectory predictor using not only the prediction accuracy/error but also robot navigation metrics such as reduction of collision rate.

In summary, this simulated robot collision experiment demonstrated our proposed BiTraP trajectory predictor can be used in future robotic applications, such as predicting collisions and selecting safest paths in robot navigation tasks. Results from this supplementary experiment are consistent with our main paper's observations and further verify our hypothesis regarding the diversity/compactness of predicted trajectory distributions, i.e., BiTraP-NP predicts more diverse distributions while BiTraP-GMM predicts more compact distributions. The SPCR, ROC (AUC) and P-R (AP) metrics used in this experiment act as a supplement to the currently reported and widely used ADE/FDE and KDE-NLL metrics in the main paper. We believe these additional metrics and experiments offer an intuitive and complementary performance evaluation of the two proposed BiTraP models (NP and GMM) and their applications for tasks such as collision prediction and path selection.

IV. Experiment and Result on nuScenes Dataset

Among the datasets we have evaluated on, JAAD and PIE are first-person view only while ETH and UCY are focusing on campus or sidewalks only. To further present the performance of BiTraP in bird's eye view autonomous driving scenarios, we evaluate on the nuScenes dataset [15]. The nuScenes dataset contains trajectories collected from 850 scenes, 700 for training and 150 for testing [15]. We followed [4] to extract training and testing trajectories and trained our model using the same configurations as in ETH-UCY experiment. Note that we treat the pedestrian position at 4 seconds in the future as the target of our goal or endpoint during training.

Evaluation metrics. To be comparable with [4], the most-likely (ML) prediction is used to compute the final displacement error (FDE). We also use the kernel density estimation negative log-likelihood (KDE NLL) as in our other experiments.

Method	FDE ML			
	@1s	@2s	@3s	@4s
Trajectron++ base [4]	0.18	0.57	2.25	2.24
Trajectron++ $\int$, map [4]	**0.07**	0.45	1.14	2.20
BiTraP-GMM (ours)	0.08	**0.43**	**1.06**	**1.99**

TABLE IV: Vehicle-only trajectory prediction results on nuScenes dataset.

Results. As can be seen in Table III, adding dynamic integration and map encoding to the base Trajectron++ improved the distribution accuracy by a large margin but does not affect the FDE ML, indicating similar modes but smaller variances of the predicted distributions. Trajectron++ based methods used interactions and/or encoded map as inputs while our BiTraP-GMM only takes target pedestrians past trajectory. As in Table III, BiTraP-GMM improves the KDE-NLL at all evaluated time steps and also improves FDE

after 2 seconds, showing how does the bi-directional strategy improves prediction accuracy. Note that the Trajectron++ benchmark lacks a ablation with integration but not map encoding (e.g. Trajectron++ $\int$) to show the necessity of map. However, our experiment shows that map may not be a very important information when predicting pedestrian trajectories on nuScenes dataset since BiTraP-GMM outperforms "Trajectron++ $\int$, map".

REFERENCES

[1] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NIPS*, 2015.

[2] C. Anderson, X. Du, R. Vasudevan, and M. Johnson-Roberson, "Stochastic sampling simulation for pedestrian trajectory prediction," in *IROS*, 2019, pp. 4236–4243.

[3] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018.

[4] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control," *arXiv preprint arXiv:2001.03093*, 2020.

[5] J. Gallier and J. H. Gallier, *Curves and surfaces in geometric modeling: theory and algorithms*. Morgan Kaufmann, 2000.

[6] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *CVPR*, 2019.

[7] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009.

[8] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an image-based motion context for multiple people tracking," in *CVPR*, 2014.

[9] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *ICRA*, 2010.

[10] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[11] M. Kuderer, H. Kretzschmar, and W. Burgard, "Teaching mobile robots to cooperatively navigate in populated environments," in *IROS*, 2013.

[12] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *IROS*, 2017.

[13] C. I. Mavrogiannis, V. Blukis, and R. A. Knepper, "Socially competent navigation planning by deep learning of multi-agent path topologies," in *IROS*, 2017.

[14] C. Cao, P. Trautman, and S. Iba, "Dynamic channel: A planning framework for crowd navigation," in *ICRA*, 2019.

[15] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.