arXiv:2104.03086v1 [cs.LG] 7 Apr 2021

Trajectory Prediction with Latent Belief Energy-Based Model

Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu
Department of Statistics, University of California, Los Angeles (UCLA)
{bopang, tyzhao, xiexu}@ucla.edu, ywu@stat.ucla.edu

Abstract

Human trajectory prediction is critical for autonomous platforms like self-driving cars or social robots. We present a latent belief energy-based model (LB-EBM) for diverse human trajectory forecast. LB-EBM is a probabilistic model with cost function defined in the latent space to account for the movement history and social context. The low-dimensionality of the latent space and the high expressivity of the EBM make it easy for the model to capture the multi-modality of pedestrian trajectory distributions. LB-EBM is learned from expert demonstrations (i.e., human trajectories) projected into the latent space. Sampling from or optimizing the learned LB-EBM yields a belief vector which is used to make a path plan, which then in turn helps to predict a long-range trajectory. The effectiveness of LB-EBM and the two-step approach are supported by strong empirical results. Our model is able to make accurate, multi-modal, and social compliant trajectory predictions and improves over prior state-of-the-arts performance on the Stanford Drone trajectory prediction benchmark by 10.9% *and on the ETH-UCY benchmark by* 27.6%.

1. Introduction

Forecasting the future trajectories of pedestrians is critical for autonomous moving platforms like self-driving cars or social robots with which humans are interacting. It has recently attracted interest from many researchers [15, 64, 25, 50, 3, 7, 28, 32]. See [49] for an overview. Trajectory forecast is a challenging problem since human future trajectories depend on a multitude of factors such as past movement history, goals, behavior of surrounding pedestrians. Also, future paths are inherently multimodal. Given the past trajectories, there are multiple possible future paths. We propose a latent belief energy-based model (LB-EBM) which captures pedestrian behavior patterns and subtle social interaction norms in the latent space and make multimodal trajectory predictions. LB-EBM is learned from expert demonstrations (i.e., human trajectories) following the principle of inverse reinforcement learning (IRL) [36, 11, 12, 17].

Traditional IRL approaches [36] first learn a cost function from expert demonstrations in an outer loop and then use reinforcement learning to extract the policy from the learned cost function in an inner loop. These approaches are often highly computationally expensive. To avoid such an issue, GAIL (Generative Adversarial Imitation Learning) [21, 7] optimizes a policy network directly. GAIL can generate multimodal action predictions given an expressive policy generator. The multimodality is however modeled implicitly and completely relies on the policy generator. Our approach strikes a middle ground between traditional IRL and GAIL. We learn an energy-based model (EBM) as the cost function in a low dimensional latent space and map the EBM distribution to actions with a policy generator. Similar to traditional IRL, we learn a cost function but our cost function is defined in a low dimensional space so that our cost function is easier to model and learn. Resembling GAIL, we also learn a policy generator which allows for directly mapping a latent vector to the action trajectory, while we explicitly learn a multimodal cost function instead of learning it implicitly and completely relying on the policy generator.

An EBM [59, 38, 40] in the form of Boltzmann or Gibbs distribution maps a latent vector to its probability. It has no restrictions in its form and can be instantiated by any function approximators such as neural networks. Thus, this model is highly expressive and learning from human trajectories allows it to capture the multimodality of the trajectory distribution. Our proposed LB-EBM is defined in a latent space. An encoder is jointly learned to project human trajectories into the latent space and hence provides expert demonstrations to the latent cost function.

Furthermore, this cost function accounts for trajectory history and motion behavior of surrounding pedestrians. Thus sampling from or optimizing the cost function yields a latent belief, regarding future trajectory, which considers the centric agent's behavior pattern and social context surrounding this agent. A future trajectory is then forecasted in two steps. We first use the social-aware latent belief vector to make a rough plan for future path. It is intuitive that human do not plan every single future step in advance but we often have a rough idea about how to navigate through our future

path, which is based on one's belief after observing other agents' motion. The belief is inherently related to the agent's behavior pattern. This forms the intuitive motivation of our modeling approach. Conditioned on the plan, the trajectory is then predicted with the assistance of individual motion history and social cues. Several recent works take two steps to make trajectory forecast. They either first estimate the final goal [32] or make a plan on a coarse grid map [29]. We take a similar approach. The plan in our approach is defined to be positions of some well-separated steps in the future trajectory, which can be easily extracted from the data.

The proposed LB-EBM and other modules are learned end-to-end. We test our model on the Stanford Drone (SDD) trajectory prediction benchmark and the ETH-UCY benchmark and improves the prior state-of-the-art performance by 10.9% on SDD and 27.6% on ETH-UCY.

Our work has the following contributions.

- We propose a latent belief energy-based model (LB-EBM), following the principle of IRL, which naturally captures the multimodal human trajectory distribution.
- Our approach predicts multimodal and social compliant future trajectories.
- Our model achieves the state-of-the-art on widely-used human trajectory forecasting benchmarks.

2. Related Work

Agents' motions depend on their histories, goals, social interactions with other agents, constraints from the scene context, and are inherently stochastic and multimodal. Conventional methods of human trajectory forecasting model contextual constraints by hand-crafted features or cost functions [4, 20, 61]. With the recent success of deep networks, RNN-based approaches have become prevalent. These works propose to model interactions among multiple agents by applying aggregation functions on their RNN hidden states [1, 15, 19], running convolutional layers on agents' spatial feature maps [5, 10, 64, 58], or leveraging attention mechanisms or relational reasoning on constructed graphs of agents [27, 50, 51, 63, 57]. Some recent studies are, however, rethinking the use of RNN and social information in modeling temporal dependencies and borrowing the idea of transformers into the area [13]. We apply these social interaction modeling approaches with a few modifications in our work.

Modeling Goals. Recent progress has suggested that directly modeling goals could significantly decrease the error for trajectory forecasting. [47] introduces a prediction method conditioning on agent goals. [32] proposes to first predict the goal based on agents' individual histories and then to forecast future trajectories conditioning on the predicted goal. [29] introduces a two-step planning scheme, first in a coarse grid then in a finer one, which can be viewed as directly modeling goals and sub-goals. We follow the general scheme of two-step prediction. The plan in our approach is defined to be positions of some well-separated steps in the future trajectory, which can be easily extracted from the data.

Multimodality. Most recent prediction works have emphasized more on modeling the multimodality nature of human motions. [2, 6] directly predict multiple possible maneuvers and generate corresponding future trajectories given each maneuver. [25, 22] use Variational Auto-Encoders [8] and [15, 25, 50, 64] use Generative Adversarial Networks [14, 33] to learn distributions. Many works [29, 44, 46, 54] also focus on developing new datasets, proposing different formulations, utilizing latent variable inference, and exploring new loss functions to account for multimodality. Our work adopts the likelihood-based learning framework with variational inference. We propose a novel way to model the multimodality of human trajectories, by projecting them into a latent space with variational inference and leveraging the strength of latent space energy-based model.

Value Function. Human behaviors are observed as actions, e.g. trajectories, but the actions are actually guided by hidden value functions, revealing human preference and cost over different actions. Some previous works explicitly or implicitly model these types of cost functions as intermediate steps for sampling possible futures. These works generally follow the reinforcement learning formulation of value functions Q. [37] directly uses Q-Learning to learn value functions. [60, 24] formulate trajectory planning and prediction problems as inverse optimal control and GAIL (generative adversarial imitation learning) problems. [31] models social interaction by game theory and attempt to find the hidden human value by fictitious play. P2TIRL [7] is learned by a maximum entropy inverse reinforcement learning (IRL). Our work also follows the basic principle of inverse reinforcement learning to learn human cost functions explicitly in a latent space.

Energy-Based Models. The energy function in the EBM [66, 59, 38, 9, 18] can be viewed as an objective function, a cost function, or a critic [53]. It captures regularities, rules or constrains. It is easy to specify, although optimizing or sampling the energy function requires iterative computation such as MCMC. Recently [40, 41, 42] proposed to learn EBM in a low dimensional latent space, which makes optimizing or sampling the energy function much more efficient and convenient. The current work follows this approach.

3. Model and Learning

3.1. Problem Definition

Let $x_i^t \in \mathbb{R}^2$ denote the position of a person i at time t in a scene where there are n people in total. The history

trajectory of the person i is $\boldsymbol{x}_i = \{x_i^t, t = 1, ..., t_{past}\}$ and $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, ..., n\}$ collects past trajectories of all people in a scene. Similarly, the future trajectory of this person at time t is denoted as y_i^t. $\boldsymbol{y}_i = \{y_i^t, t = t_{past} + 1, ..., t_{pred}\}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_i, i = 1, ..., n\}$ indicate the future trajectory of the person i and all future trajectories, respectively. The goal is to jointly predict the future trajectories of all the agents in the scene or to learn the probabilistic distribution, $p(\boldsymbol{Y}|\boldsymbol{X})$.

Directly modeling $p(\boldsymbol{Y}|\boldsymbol{X})$ is essentially supervised learning or behavior cloning which often fails to capture the multimodality. Instead, we introduce two auxiliary variables. The first is $\boldsymbol{z}_i$ which represents the latent belief of the agent i after observing the trajectory history of his or her own and surrounding agents, $\boldsymbol{X}$. Let $\boldsymbol{Z} = \{\boldsymbol{z}_i, i = 1, ..., n\}$. $\boldsymbol{z}_i$ is a latent variable since we cannot observe one's latent belief. The other auxiliary variable is $\boldsymbol{p}_i$ which denotes the plan of the agent i considering the latent belief $\boldsymbol{z}_i$ and trajectory history $\boldsymbol{X}$. Similarly, let $\boldsymbol{P} = \{\boldsymbol{p}_i, i = 1, ..., n\}$. $\boldsymbol{p}_i$ can be either latent or observed. We choose to use a few well-separated steps of future trajectory, $\boldsymbol{y}_i$, to represent one's plan, making it an observable. Thus, we can extract plan from the data to provide supervision signal, making the learning easier. With the aforementioned setup, we model the following joint distribution,

$$p(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \underbrace{p(\boldsymbol{Z}|\boldsymbol{X})}_{\text{LB-EBM}} \overbrace{p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})}^{\text{Plan}} \underbrace{p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}_{\text{Prediction}}. \quad (1)$$

After learning the model, we can follow the above chain to make trajectory prediction. A well-learned LB-EBM or cost function captures expert's belief distribution given trajectory history and motion behavior of surrounding agents. Sampling from or optimizing this cost function gives a good belief representation taking account into individual behavior pattern and social context. This cost function is inherently multimodal since it learns from the multimodal human trajectories. We can then make a plan with $p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})$ (the plan module) by directly generating a trajectory plan. Lastly, $p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})$ (the prediction module) makes a trajectory prediction given the plan and past history. In the following section, we detail each part of the decomposed distribution and introduce related encoding functions.

3.2. LB-EBM

In our approach, the key step is to learn a cost function defined in a latent belief space. For a latent belief vector $\boldsymbol{z}_i$, the cost function is defined to be

$$C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X})) \quad (2)$$

where α denotes the parameters of the cost function. Two relevant encoding modules are, E_{past} which is used to encode the trajectory history $\boldsymbol{x}_i$ of each agent and P_{social} which is a pooling module that aggregates $\{E_{past}(\boldsymbol{x}_i), i = 1, ..., n\}$ to provide the latent belief space with individual behavior history and social context. $C_\alpha(\cdot)$ takes $[\boldsymbol{z}_i; P_{social}(\boldsymbol{X})]$ as the input where $[\,\cdot\,;\,\cdot\,]$ indicates concatenation.

Assuming we have a well-learned cost function, we can find a $\boldsymbol{z}_i$ by minimizing the cost function with respect to it given $\boldsymbol{X}$, generate a plan with the latent belief, and then make the trajectory plan. The cost function is learned from expert demonstrations projected into the latent space. A plan, $\boldsymbol{p}_i$, extracted from an observed future human trajectory, $\boldsymbol{y}_i$, can be projected to the latent space. Suppose $\boldsymbol{y}_i$ consists of 12 time steps and $\boldsymbol{p}_i$ can take the positions at the 3rd, 6th, 9th, and 12th time steps as the plan. Denote the projected latent vector to be $\boldsymbol{z}_i^+$. α is learned from $\{\boldsymbol{z}_{ij}^+, i = 1, ..., n; j = 1, ..., N\}$ where j indicates the jth scene with N scenes in total. See section 3.6 for the learning details. The projection or inference is done by an inference network $E_{inference}$. The distribution of the inferred latent belief is $q_\phi(\boldsymbol{z}_i|\boldsymbol{p}_i, \boldsymbol{X})$, which is assumed to be a multivariate Gaussian with a diagonal covariance matrix. In particular, the mean function $\mu_\phi(\boldsymbol{p}_i, \boldsymbol{X})$ and covariance matrix $\sigma_\phi^2(\boldsymbol{p}_i; \boldsymbol{X})$ both takes $[E_{plan}(\boldsymbol{p}_i); P_{social}(\boldsymbol{X})]$ as the input and share the neural network module except the last layer. Here E_{plan} is simply an embedding function which encodes the plan $\boldsymbol{p}_i$ into a feature space to be ready to concatenate with $P_{social}(\boldsymbol{X})$.

The LB-EBM assumes the following conditional probability density function

$$p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X})) \quad (3)$$

$$= \frac{1}{Z_\alpha(P_{social}(\boldsymbol{X}))} \exp\left[-C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))\right] p_0(\boldsymbol{z}_i), \quad (4)$$

where $Z_\alpha(P_{social}(\boldsymbol{X})) = \int \exp\left[-C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))\right] d\boldsymbol{z}_i$ is the normalizing constant or partition function and $p_0(\boldsymbol{z}_i)$ is a known reference distribution, assumed to be standard Gaussian in this paper. The cost function C_α serves as the energy function. The latent belief vectors of experts $\boldsymbol{z}_{ij}^+$ are assumed to be random samples from $p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X}))$ and thus has low cost on $C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))$.

The joint distribution of the latent belief vectors of agents in a scene is then defined to be

$$p(\boldsymbol{Z}|\boldsymbol{X}) = \prod_{i=1}^{n} p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X})), \quad (5)$$

where $\{\boldsymbol{z}_i, i = 1, ..., n\}$ given the joint trajectory history $\boldsymbol{X}$ are independent because an agent cannot observe the belief of other agents.

To sample from LB-EBM, we employ Langevin dynamics [35, 65, 39]. For the target distribution $p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X}))$, the dynamics iterates

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + s\nabla_{\boldsymbol{z}} \log p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X})) + \sqrt{2s}\epsilon_k, \quad (6)$$

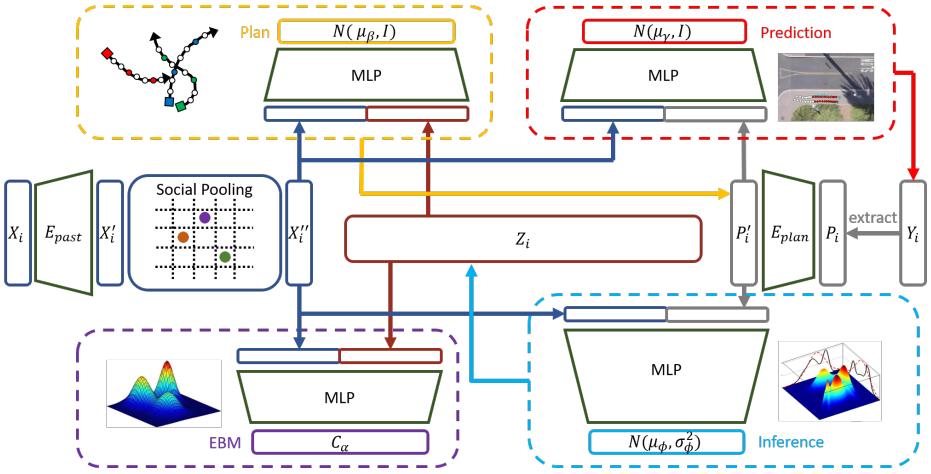


Figure 1. An overview of our model on an individual agent i. The past trajectory x_i (left side in the figure) is encoded by E_{past} to get the individual encoding x_i'. The social pooling module P_{social} is then applied to get the agent's history encoding x_i'' accounting for social context. In training, the ground-truth plan p_i (right side in the figure) is extracted from the future trajectory y_i (e.g., extract the steps 3, 6, 9, 12 from a 12-time-step future as the plan) and then encoded by E_{plan} to get p_i'. The expert plan is then projected into the latent space, conditional on the trajectory history and social context, x_i'', through the inference module (light blue). It takes x_i'' and p_i' as input, parameterized by ϕ, and is only used in training to output the mean μ_ϕ and co-variance matrix σ_ϕ^2 for the posterior distribution, q_ϕ, of the latent vector z_i. Purple part denotes the latent belief energy-based model (LB-EBM) module, C_α, defined on the latent belief vector z_i conditional on x_i''. The LB-EBM learns from the posterior distribution of the projected ground-truth plan q_ϕ. A sample from the posterior (in training) or a sample from LB-EBM (in testing) enters the plan module (yellow) together with x_i''. The plan module is parametrized by β, which is a regular regression model where the mean μ_β is estimated and used as the module prediction. The generated plan together with x_i'' enters the prediction module (red), parameterized by γ. It is also a regular regression model where the mean μ_γ is estimated and used as the module prediction, which is also the trajectory forecast of the whole network.

where k indexes the time step of the Langevin dynamics, s is a small step size, and $\epsilon_k \sim \mathrm{N}(0, I)$ is the Gaussian white noise. Note that the index i for $\boldsymbol{z}$ is removed for notational simplicity. $\nabla_{\boldsymbol{z}} \log p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X}))$ can be efficiently computed by back-propagation. Given the low-dimenionality of the latent space, Langevin dynamics sampling mixes fast. In practice, we run the dynamics for a fixed number of times (20). The small number of steps and the small model size of the LB-EBM make it highly affordable in practice.

3.3. Plan

The distribution of the plan of the agent i is $p_\beta(\boldsymbol{p}_i|\boldsymbol{z}_i, \boldsymbol{X})$, and it is assumed to be a Gaussian distribution with mean $\mu_\beta(\boldsymbol{z}_i, \boldsymbol{X})$ and an identity covariance matrix. In particular the mean function takes as input the concatenation $[\boldsymbol{z}_i; P_{social}(\boldsymbol{X})]$. The joint distribution of the plans of all agents in a scene is

$$p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) = \prod_{i=1}^{n} p_\beta(\boldsymbol{p}_i|\boldsymbol{z}_i, P_{social}(\boldsymbol{X})), \tag{7}$$

where $\boldsymbol{p}_i$ is assumed to be independent of $\{\boldsymbol{z}_j, j \neq i\}$ given $\boldsymbol{z}_i$ and $P_{social}(\boldsymbol{X})$ and $\{\boldsymbol{p}_i, i = 1, ..., n\}$ are assumed to be independent conditional on $\{\boldsymbol{z}_i\}$ and $P_{social}(\boldsymbol{X})$.

3.4. Prediction

The prediction distribution is defined similarly as the plan distribution,

$$p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) = \prod_{i=1}^{n} p_\gamma(\boldsymbol{y}_i|\boldsymbol{p}_i, P_{social}(\boldsymbol{X})), \tag{8}$$

and $p_\gamma(\boldsymbol{y}_i|\boldsymbol{p}_i, P_{social}(\boldsymbol{X}))$ assumes a Gaussian distribution with mean $\mu_\gamma(\boldsymbol{p}_i, \boldsymbol{X})$ and an identity covariance matrix. The input to the mean function is $[E_{plan}(\boldsymbol{p}_i); P_{social}(\boldsymbol{X})]$.

3.5. Pooling

The trajectory history $\boldsymbol{X}$ of agents in a scene is pooled through self-attention [56]. It allows us to enforce a spatial-temporal structure on the social interactions among agents. This enforcement is simply achieved by designing a spatial-temporal binary mask with prior knowledge. We follow the mask design of [32]. The pooling mask M is defined to be,

$$M[i,j] = \begin{cases} 0 & \text{if } \min\limits_{1 \leq s,t \leq t_{past}} \|x_i^t - x_j^s\|_2 > d \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Adjusting the hyperparameter d allows for varying the social-temporal adjacency of social interactions.

3.6. Joint learning

The log-likelihood of data in a single scene, $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{P})$, is

$$\log p(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \log \int_{\boldsymbol{Z}} p(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{10}$$

which involves the latent variable $\boldsymbol{Z}$ and directly optimizing it involves sampling from the intractable posterior $p(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$. We however can optimize a variational lower bound of it in an end-to-end fashion to learn the entire network,

$$L(\theta) = \mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) \tag{11}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) \tag{12}$$

$$- \mathbb{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})||p_0(\boldsymbol{Z})) \tag{13}$$

$$- \mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) - \log Z_\alpha(\boldsymbol{X}), \tag{14}$$

where θ collects the parameters of the whole network. Also note that $p_0(\boldsymbol{Z}) = \prod_i p_0(\boldsymbol{z}_i)$ and $C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) = \sum_i C_\alpha(\boldsymbol{z}_i, \boldsymbol{X})$. Please see the supplementary for the derivation details. The gradients of all terms are straightforward with backpropagation except $\log Z_\alpha(\boldsymbol{X})$. The gradient of it with respect to α is $\mathrm{E}_{p(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})]$. It involves sampling from LB-EBM. This is done with Langevin dynamics (Equation 6). As we discussed earlier, sampling from LB-EBM only requires a small number of steps and the necessary model size is fairly small due to the low dimensionality. Thus the sampling is highly affordable. Although the loss function $-L\{\theta\}$ is optimized end-to-end, let us take a close look at the optimization of the cost function given its core role in our model. Let $\mathcal{J}(\alpha)$ be the loss function of the LB-EBM, the gradient of it with respect to α is,

$$\nabla_\alpha \mathcal{J}(\alpha) \tag{15}$$

$$= \mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})] - \mathrm{E}_{p(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})], \tag{16}$$

where $q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ projects the expert plan $\boldsymbol{P}$ to the latent belief space. α is updated based on the difference between the expert beliefs and those sampled from the current LB-EBM. Thus, the latent cost function is learned to capture expert beliefs given the trajectory history and surrounding context.

4. Experiments

We test our model on two widely used pedestrians trajectory benchmarks (see section 4.2 for details) against a variety competitive baselines. These experiments highlight the effectiveness of our model with (1) improvements over the previous state-of-the-art models on the accuracy of trajectory prediction and (2) the prediction of multimodal and social compliant trajectories as demonstrate in qualitative analysis.

4.1. Implementation Details and Design Choices

The trajectory generator or policy network is an autoregressive model in most prior works [1, 15, 25, 50]. Some recent works explored the use of a non-autoregressive model [32, 45]. We choose to use a non-autoregressive model (MLP) considering its efficiency and the avoidance of exposure bias inherent in autoregressive models. The potential issue of using an non-autoregressive model is that it might fail to capture the dependency among different time steps. However, this is a lesser issue since the proposed LB-EBM is expressive and multi-modal and might be able to model the dependency across multiple time steps. Furthermore, the trajectory prediction is based on a plan over the whole forecasting time horizon, making an auto-regressive model further unnecessary.

The dimension of LB-EBM is 16 and is implemented with 3-layer MLP with an hidden dimension of 200. We always use 20 steps for Langevin sampling from LB-EBM in both training and inference. It is possible to amortize the sampling on the learned cost function by learning an auxiliary latent generator such as using noise contrastive estimation [16]. However, due to the low dimensionality of the latent space, 20 steps are highly affordable. We thus prefer keeping our model and learning method pure and simple.

In both benchmarks, the model aims to predict the future 12 time steps. The plan is extracted by taking the positions at the 3rd, 6th, 9th, and 12th time steps.

All other modules in our model are also implemented with MLPs. The batch size is 512 for the Stanford Drone dataset and is 70 for all the ETH-UCY datasets. The model is trained end-to-end with an Adam optimizer with an learning rate of 0.0003. The model is implemented in Pytorch [43]. Our code is released at `https://github.com/bpucla/lbebm`.

4.2. Datasets

Stanford Drone Dataset. Stanford Drone Dataset [48] is a large-scale pedestrian crowd dataset in bird's eye view. It consists of 20 scenes captured using a drone in top down view around the university campus containing several moving agents such as humans bicyclists, skateboarders and vehicles. It consists of over $11,000$ unique pedestrians capturing over $185,000$ interactions between agents and over $40,000$ interactions between the agent and scene [48]. We use the standard train-test split which is widely used in prior works such as [50, 15, 32].

ETH-UCY. It is a collection of relatively small benchmark pedestrian crowd datasets. It consists of five different scenes: ETH and HOTEL (from ETH) and UNIV, ZARA1, and ZARA2 (from UCY). The positions of pedestrians are in world-coordinates and hence the results are reported in meters. We use the leave-one-out strategy for training and testing, that is, training on four scenes and testing on the

fifth one, as done in previous works [15, 26, 32]. We split the trajectories into segments of 8s and use 3.2s of trajectory history and a 4.8s prediction horizon, with each time step of 0.4s.

4.3. Baseline Models

We compare the proposed approach based on LB-EBM to a wide range of baseline models and state-of-the-art works. The compared work covers very different learning regimes for modeling human trajectory and accounting for multi-modality and social interaction. We briefly describe below the representative baselines.

- S-LSTM [1] is the simplest deterministic baseline based on social pooling on LSTM states.
- S-GAN-P [15] is a stochastic GAN-based simple baseline extended from S-LSTM.
- MATF [64] is a GAN-based convolutional network built upon feature maps of agents and context.
- Desire [25] is an VAE-based sophisticated stochastic model.
- Sophie [50] is a complex attentive GAN modeling both social interactions and scene context.
- CGNS [26] uses conditional latent space learning with variational divergence minimization.
- P2TIRL [7] is learned by maximum entropy inverse reinforcement learning policy.
- SimAug [28] uses additional 3D multi-view simulation data adversarially.
- PECNet [32] is a VAE based state-of-the-art model with goal conditioning predictions.

4.4. Quantitative Results

In this section, we compare and discuss our method's performance against the aforementioned baselines based on the Average Displacement Error (ADE) and Final Displacement Error (FDE) with respect to each time-step t within the prediction horizon.

$$\begin{aligned} \text{ADE}_i &= \frac{1}{T_{\text{pred}}} \sum_{t=1}^{T_{\text{pred}}} d_{l_2}(\hat{y}_i^t, y_i^t) \\ \text{ADE} &= \frac{1}{n} \sum_i \text{ADE}_i \\ \text{FDE}_i &= d_{l_2}(\hat{y}_i^{T_{\text{pred}}}, y_i^{T_{\text{pred}}}) \\ \text{FDE} &= \frac{1}{n} \sum_i \text{FDE}_i \end{aligned} \tag{17}$$

where d_{l_2} indicates the Euclidean distance. Following the evaluation protocol of the prior work [15, 23, 32, 64], we use Best-of-K evaluation. In particular, the minimum ADE and FDE from K randomly sampled trajectories are considered as the model evaluation metrics. And $K = 20$ is used in our experiments. Recently, some researchers [22, 52, 55] propose to use kernel density estimate-based negative log likelihood (KDE NLL) for evaluation. Since only few papers reported NLL results on our considered benchmarks and thus it might not be easy to have a fair comparison with most baselines, we choose to focus on the widely-adopted ADE and FDE. Please see the supplementary for the NLL evaluation of our model.

	ADE	FDE
S-LSTM [1]	31.19	56.97
S-GAN-P [15]	27.23	41.44
MATF [64]	22.59	33.53
Desire [25]	19.25	34.05
SoPhie [50]	16.27	29.38
CF-VAE [3]	12.60	22.30
P2TIRL [7]	12.58	22.07
SimAug [28]	10.27	19.71
PECNet [32]	9.96	15.88
Ours	**8.87**	**15.61**

Table 1. ADE / FDE metrics on Stanford Drone for LB-EBM compared to baselines are shown. All models use 8 frames as history and predict the next 12 frames. The lower the better.

Stanford Drone Dataset: Table 1 summarizes the results of our proposed method against the baselines and state-of-the-art methods. Our proposed method achieves a superior performance compared to the previous state-of-the-art models [3, 7, 32] on ADE by a significant margin of 10.9%. While our improvement over other baselines on FDE is clear, the improvement over the PECNet is not significant. This might be because the PECNet focuses on optimizing the goal or the final step.

ETH-UCY: Table 2 shows the results for the evaluation of our proposed method on the ETH/UCY scenes. We use the leave-one-out evaluation protocol following CGNS [26] and Social-GAN [15]. We observe that the proposed LB-EBM outperforms prior methods, including the previous state-of-the-art [26]. We improve over the state-of-the-art on the average ADE by 27.6% with the effect being the most on ETH (44.4%) and least on ZARA1 (9.1%). We also observe a clear improvement on the FDE.

4.5. Qualitative Results

In this section, we present qualitative results of our proposed method on the Stanford Drone dataset. In Figure 2, we inspect the results under three different setups across 4

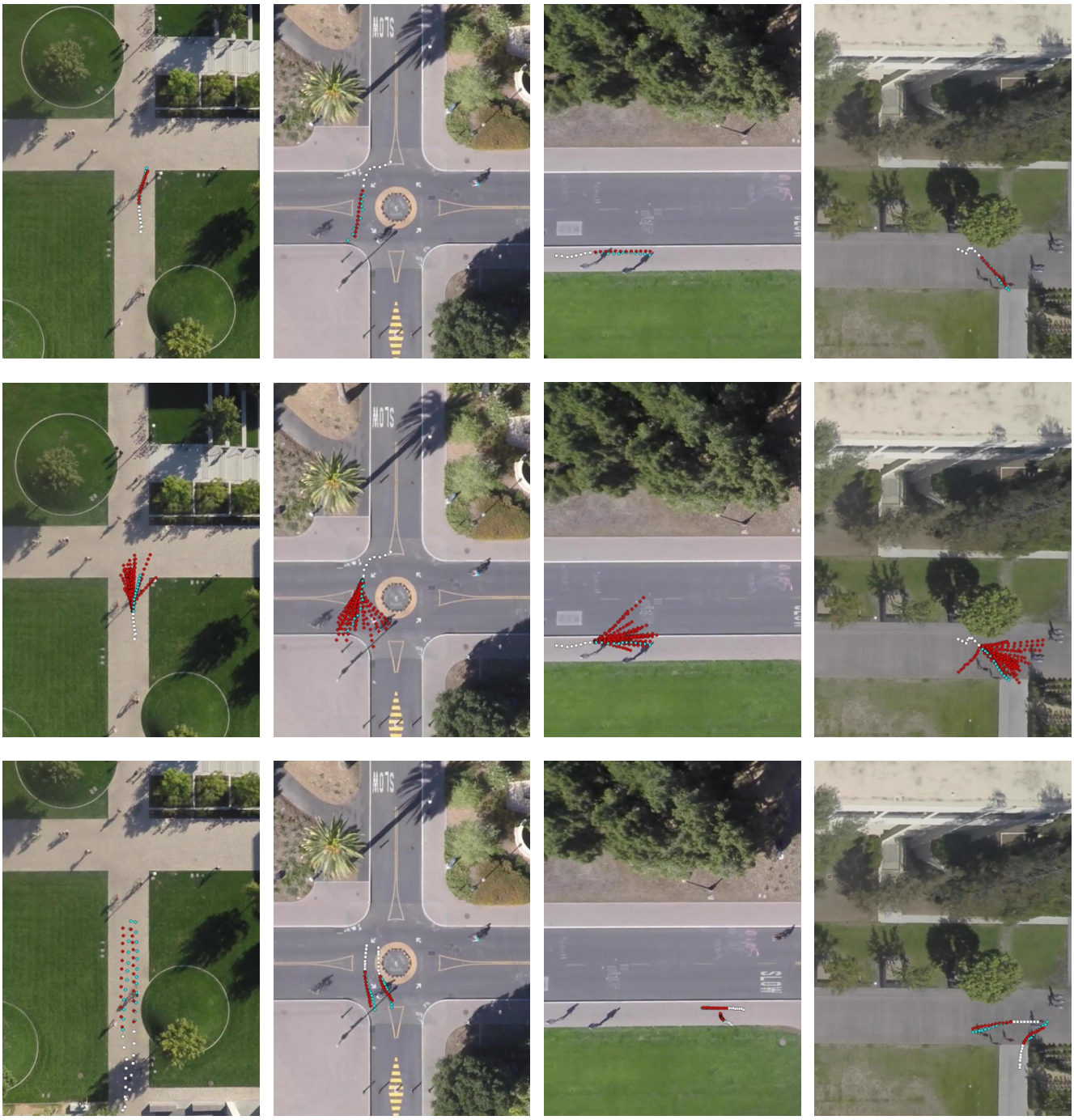


Figure 2. Qualitative results of our proposed method across 4 different scenarios in the Stanford Drone. First row: The best prediction result sampled from 20 trials from LB-EBM. Second row: The 20 predicted trajectories sampled from LB-EBM. Third row: prediction results of agent pairs that has social interactions. The observed trajectories, ground truth predictions and our model's predictions are displayed in terms of white, blue and red dots respectively.

different scenarios. Those scenarios are selected involving various road conditions including crossing, sidewalk and roundabout. The first row presents the best prediction result, among 20 random samples drawn from the LB-EBM with respect to the ADE criterion, for each scenario. Our model is able to produce predictions that are close to the ground-truth trajectories in these scenarios. The second row illustrates the 20 predicted trajectories sampled from our method. By

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear * [1]	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
SR-LSTM-2 * [63]	0.63 / 1.25	0.37 / 0.74	0.51 / 1.10	0.41 / 0.90	0.32 / 0.70	0.45 / 0.94
S-LSTM [1]	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
S-GAN-P [15]	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
SoPhie [50]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
MATF [64]	0.81 / 1.52	0.67 / 1.37	0.60 / 1.26	0.34 / 0.68	0.42 / 0.84	0.57 / 1.13
CGNS [26]	0.62 / 1.40	0.70 / 0.93	0.48 / 1.22	0.32 / 0.59	0.35 / 0.71	0.49 / 0.97
PIF [30]	0.73 / 1.65	0.30 / 0.59	0.60 / 1.27	0.38 / 0.81	0.31 / 0.68	0.46 / 1.00
STSGN [62]	0.75 / 1.63	0.63 / 1.01	0.48 / 1.08	0.30 / 0.65	0.26 / 0.57	0.48 / 0.99
GAT [23]	0.68 / 1.29	0.68 / 1.40	0.57 / 1.29	0.29 / 0.60	0.37 / 0.75	0.52 / 1.07
Social-BiGAT [23]	0.69 / 1.29	0.49 / 1.01	0.55 / 1.32	0.30 / 0.62	0.36 / 0.75	0.48 / 1.00
Social-STGCNN [34]	0.64 / 1.11	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
PECNet [32]	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
Ours	**0.30 / 0.52**	**0.13 / 0.20**	**0.27 / 0.52**	**0.20 / 0.37**	**0.15 / 0.29**	**0.21 / 0.38**

Table 2. ADE / FDE metrics on ETH-UCY for the proposed LB-EBM and baselines are shown. The models with * mark are non-probabilistic. All models use 8 frames as history and predict the next 12 frames. Our model achieves the best average error on both ADE and FDE metrics. The lower the better.

visualizing the results, we can see that LB-EBM is able to generate multi-modal and diverse predictions. Further, we display the prediction results of a pair of agents with social interactions in the third row. Interaction details such as "straight going together", "turning together", "yielding" and "collision avoidance" are captured by our proposed model. It demonstrates the effectiveness of our LB-EBM to model the agent-wise interactions for trajectory predictions.

4.6. Ablation Study

We conduct ablation studies to examine the important components of our model. In particular, we ablate each component of the overall learning objective as specified in Equation 11 - 14. The results are summarized in Table 3. Equation 11 is the basic reconstruction term and has to be kept. But we can replace Equation 11 and 12 with $\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{Y},\boldsymbol{X})} \log p(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X})$. That is, the model predicts the full trajectory directly without generating a plan first. It is corresponding to *EBM without Plan* in Table 3. Equation 13 and 14 together are the KL divergence between the variational posterior $q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})$ and the EBM prior $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ (note that $p_0(\boldsymbol{Z})$ is the base distribution for the EBM). We can replace $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ with a Gaussian distribution conditional on $\boldsymbol{X}$, corresponding to the *Gaussian with Plan* condition. The previous two changes together lead to the *Gaussian without Plan* condition. The ablation results indicate the effectiveness of the latent belief EBM and two-step approach.

In addition, we evaluate the model without the social pooling such that LB-EBM makes predictions only based on an agent's own action history (see the *EBM with Plan without Social* condition in Table 3). The decreased performance in ADE and FDE of this condition indicates that LB-EBM is effective to take into account social cues when provided.

Time Steps	ADE	FDE
Gaussian without Plan	18.61	27.55
EBM without Plan	10.28	18.60
Gaussian with Plan	9.53	16.32
EBM with Plan without Social	9.23	16.57
EBM with Plan	8.87	15.61

Table 3. ADE / FDE metrics on Stanford Drone for different ablation conditions. The lower the better.

5. Conclusion

In this work, we present the LB-EBM for diverse human trajectory forecast. LB-EBM is a probabilistic cost function in the latent space accounting for movement history and social context. The low-dimensionality of the latent space and the high expressivity of the EBM make it easy for the model to capture the multimodality of pedestrian trajectory distributions. LB-EBM is learned from expert demonstrations (i.e., human trajectories) projected into the latent space. Sampling from or optimizing the learned LB-EBM is able to yield a social-aware belief vector which is used to make a path plan. It then helps to predict a long-range trajectory. The effectiveness of LB-EBM and the two-step approach are supported by strong empirical results. Our model is able to make accurate, multimodal, and social compliant trajectory predictions and improves over prior state-of-the-arts performance on the Stanford Drone trajectory prediction benchmark by 10.9% and on the ETH-UCY benchmark by 27.6%.

Acknowledgment

The work is supported by NSF DMS-2015577 and DARPA XAI project N66001-17-2-4029.

References

[1] 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*, pages 961–971, 2016. 2, 5, 6, 8

[2] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr. A combined model and learning based framework for interaction-aware maneuver prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2016. 2

[3] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019. 1, 6

[4] N. Deo, A. Rangesh, and M. M. Trivedi. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *arXiv:1801.06523*, 2018. 2

[5] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *IEEE Computer Vision and Pattern Recognition Workshop on Joint Detection, Tracking, and Prediction in the Wild*, 2018. 2

[6] Nachiket Deo and Mohan M. Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2

[7] Nachiket Deo and Mohan M Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint arXiv:2001.00735*, 2020. 1, 2, 6

[8] Kingma Diederik and Welling Max. Auto-encoding variational bayes. In *ICLR*. 2014. 2

[9] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *CoRR*, abs/1903.08689, 2019. 2

[10] Isht Dwivedi, Srikanth Malla, Behzad Dariush, and Chiho Choi. Ssp: Single shot future trajectory prediction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 2

[11] Chelsea Finn, Paul F. Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *CoRR*, abs/1611.03852, 2016. 1

[12] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58, 2016. 1

[13] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting, 2020. 2

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680. 2014. 2

[15] 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*, pages 2255–2264, 2018. 1, 2, 5, 6, 8, 13

[16] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 5

[17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017. 1

[18] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7978–7987, 2020. 2

[19] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018. 2

[20] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[21] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016. 1

[22] 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*, pages 2375–2384, 2019. 2, 6, 13

[23] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, S Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *arXiv preprint arXiv:1907.03395*, 2019. 6, 8, 13

[24] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. Imitating driver behavior with generative adversarial networks. *Intelligent Vehicles Symposium (IV)*, 2017. 2

[25] 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*, pages 336–345, 2017. 1, 2, 5, 6

[26] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. *arXiv preprint arXiv:1905.01631*, 2019. 6, 8

[27] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020. 2

[28] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from simulation for trajectory prediction. 2020. 1, 6

[29] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition, 2020. 2

[30] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019. 8

[31] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4636–4644. IEEE Computer Society, 2017. 2

[32] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *arXiv preprint arXiv:2004.02025*, 2020. 1, 2, 4, 5, 6, 8, 13

[33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arxiv:1411.1784*, 2014. 2

[34] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020. 8

[35] Radford M Neal. MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011. 3

[36] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *in Proc. 17th International Conf. on Machine Learning*. Citeseer, 2000. 1

[37] D. Ngai and N. Yung. A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 12:509–522, 2011. 2

[38] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, Canada*, 2019. 1, 2

[39] Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference, 2020. 3

[40] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2

[41] Bo Pang, Tian Han, and Ying Nian Wu. Learning latent space energy-based prior model for molecule generation. *arXiv preprint arXiv:2010.09351*, 2020. 2

[42] Bo Pang, Erik Nijkamp, Jiali Cui, Tian Han, and Ying Nian Wu. Semi-supervised learning by latent space energy-based model of symbol-vector coupling. *arXiv preprint arXiv:2010.09359*, 2020. 2

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[44] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[45] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2020. 5

[46] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *The European Conference on Computer Vision (ECCV)*, 09 2018. 2

[47] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[48] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016. 5

[49] 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):895–935, 2020. 1

[50] 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 Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 1, 2, 5, 6, 8

[51] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. Car-net: Clairvoyant attentive recurrent network. *arXiv:1711.10061*, 2017. 2

[52] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020. 6, 13

[53] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 2

[54] Yichuan Charlie Tang and Ruslan Salakhutdinov. Multiple futures prediction. 2019. 2

[55] Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9954–9963, 2019. 6, 13

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 4

[57] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2018*, May 2018. 2

[58] Pengxiang Wu, Siheng Chen, and Dimitris N. Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[59] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2635–2644, 2016. 1, 2

[60] Yifei Xu, Tianyang Zhao, Chris L. Baker, Yibiao Zhao, and Ying Nian Wu. Learning trajectory prediction with continuous inverse optimal control via langevin sampling of energy-based models. *CoRR*, abs/1904.05453, 2019. 2

[61] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? *IEEE Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[62] Lidan Zhang, Qi She, and Ping Guo. Stochastic trajectory prediction with social graph network. *arXiv preprint arXiv:1907.10233*, 2019. 8

[63] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019. 2, 8

[64] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12126–12134, 2019. 1, 2, 6, 8, 13

[65] Song Chun Zhu and David Mumford. Grade: Gibbs reaction and diffusion equations. In *Computer Vision, 1998. Sixth International Conference on*, pages 847–854, 1998. 3

[66] Song Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998. 2

Appendix A: Learning

Model Formulation

Recall that $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, ..., n\}$ indicates the past trajectories of all agents in the scene. Similarly, $\boldsymbol{Y}$ indicates all future trajectories. $\boldsymbol{Z}$ represents the latent belief of agents. $\boldsymbol{P}$ denotes the plans. We model the following generative model,

$$p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \underbrace{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})}_{\text{LB-EBM}} \overbrace{p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})}^{\text{Plan}} \underbrace{p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}_{\text{Prediction}}. \tag{18}$$

Maximum Likelihood Learning

Let $q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_{data}(\boldsymbol{X})$ be the data distribution that generates the (multi-agent) trajectory example, $(\boldsymbol{P}, \boldsymbol{Y}, \boldsymbol{X})$, in a single scene. The learning of parameters ψ of the generative model $p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ can be based on $\min_\psi D_{KL}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}))$ where $D_{KL}(q(x) \parallel p(x)) = \mathrm{E}_q[\log q(x)/p(x)]$ is the Kullback-Leibler divergence between q and p (or from q to p since $D_{KL}(q(x) \parallel p(x))$ is asymmetric). If we observe training examples $\{(\boldsymbol{P}_j, \boldsymbol{Y}_j, \boldsymbol{X}_j), j = 1, .., N\} \sim q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_{data}(\boldsymbol{X})$, the above minimization can be approximated by maximizing the log-likelihood,

$$\sum_{j=1}^{N} \log p_\psi(\boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) = \sum_{j=1}^{N} \log \int_{\boldsymbol{Z}_j} p_\psi(\boldsymbol{Z}_j, \boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) \tag{19}$$

which leads to the maximum likelihood estimate (MLE). Then the gradient of the log-likelihood of a single scene can be computed according to the following identity,

$$\nabla_\psi \log p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \int_{\boldsymbol{Z}} p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{20}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{21}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}|\boldsymbol{X})p_\psi(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}{p_\psi(\boldsymbol{P}|\boldsymbol{X})p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{22}$$

$$= \mathrm{E}_{p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}). \tag{23}$$

The above expectation involves the posterior $p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ which is however intractable.

Variational Learning

Due to the intractablity of the maximum likelihood learning, we derive a tractable variational objective. Define

$$q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \tag{24}$$

where $q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ is a tractable variational distribution, particularly, a Gaussian with a diagnoal covariance matrix used in this work. Then our variational objective is defined to be the tractable KL divergence below,

$$D_{KL}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{25}$$

where $q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ involves either the data distribution or the tractable variational distribution. Notice that,

$$D_{KL}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{26}$$

$$= D_{KL}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{27}$$

$$+ D_{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})) \tag{28}$$

$$\tag{29}$$

which is an upper bound of $D_{KL}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}))$ due to the non-negativity of KL divergence, in particular, $D_{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}))$, and equivalently a lower bound of the log-likelihood.

We next unpack the generative model $p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ and have,

$$D_{KL}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{30}$$

$$= D_{KL}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\alpha(\boldsymbol{Z}|\boldsymbol{X})p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})) \tag{31}$$

$$= \mathrm{E}_{q_{data}(\boldsymbol{X})} \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{32}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{X})} \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{P}|\boldsymbol{Y}, \boldsymbol{X})}{p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})} \tag{33}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{X})} \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}{p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})} \tag{34}$$

Expressions 32, 33, 34 are the major objectives for learning the LB-EBM, plan, and prediction modules respectively. They are the "major" but not "only" ones since the whole network is trained end-to-end and gradients from one module can flow to the other. We next unpack each of the objectives (where $\mathrm{E}_{q_{data}(\boldsymbol{X})}$ is omitted for notational simplicity).

Expression 32 drives the learning of the LB-EBM.

$$\mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{35}$$

$$= \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_0(\boldsymbol{Z}) \exp[-C_\alpha(\boldsymbol{Z}, \boldsymbol{X})]/Z_\alpha(\boldsymbol{X})} \tag{36}$$

$$= D_{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_0(\boldsymbol{Z})) \tag{37}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) + \log Z_\alpha(\boldsymbol{X}) \tag{38}$$

where $Z_\alpha(\boldsymbol{X}) = \int_{\boldsymbol{Z}} \exp(-C_\alpha(\boldsymbol{Z}, \boldsymbol{X}))p_0(\boldsymbol{Z}) = \mathrm{E}_{p_0(\boldsymbol{Z})}(-C_\alpha(\boldsymbol{Z}, \boldsymbol{X}))$.

Let $\mathcal{J}(\alpha) = \mathrm{E}_{q_{data}(\boldsymbol{X})} \mathrm{E}_{q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) + \mathrm{E}_{q_{data}(\boldsymbol{X})} \log Z_\alpha(\boldsymbol{X})$, which is the objective for LB-EBM

learning and follows the philosophy of IRL. And its gradient is,

$$\nabla_\alpha \mathcal{J}(\alpha) \tag{39}$$
$$= \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})] \tag{40}$$
$$- \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})] \tag{41}$$

Thus, α is learned based on the distributional difference between the expert beliefs and those sampled from the current LB-EBM. The expectations over $q_{data}(\boldsymbol{X})$ and $q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})$ are approximated with a mini-batch from the empirical data distribution. The expectation over $q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})$ is approximated with samples from the variational distribution through the reparameterization trick. The expectation over $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ is approximated with samples from Langevin dynamics guided by the current cost function.

Expression 33 drives the learning of the plan module.

$$(33) = -\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X}) \tag{42}$$
$$- H(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X}) \tag{43}$$

where $H(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})$ is the conditional entropy of $q_{data}(\boldsymbol{P}|\boldsymbol{X},\boldsymbol{Y})$ and is a constant with respect to the model parameters. Thus minimizing 33 is equivalent to maximizing the log-likelihood of $p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})$.

Expression 34 drives the learning of the prediction module.

$$(34) = -\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X}) \tag{44}$$
$$- H(\boldsymbol{Y}|\boldsymbol{X}) \tag{45}$$

where $H(\boldsymbol{Y}|\boldsymbol{X})$ is the conditional entropy of $q_{data}(\boldsymbol{Y}|\boldsymbol{X})$ and is constant with respect to the model parameters. We can minimize Expression 44 for optimizing the prediction module. In the learning, $\boldsymbol{P}$ is sampled from the data distribution $q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})$. In practice, we find sampling $\boldsymbol{P}$ from the generative model $p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})$ instead facilitates learning of other modules, leading to improved performance. The objective for learning the prediction module then becomes,

$$-\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})}\mathrm{E}_{p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X}) \tag{46}$$

where

$$\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})} \tag{47}$$
$$= \int_{\boldsymbol{P}} q_{data}(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X}) \tag{48}$$
$$= \mathrm{E}_{q_{data}(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})}q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X}). \tag{49}$$

Appendix B: Negative Log-Likelihood Evaluation

Although Best-of-K on ADE and FDE (e.g., $K = 20$) is widely-adopted [15, 23, 32, 64], some researchers [22, 52, 55] recently propose to use kernel density estimate-based negative log likelihood (KDE NLL) to evaluate trajectory prediction models. This metric computes the negative log-likelihood of the groud-truth trajectory at each time step with kernel density estimates and then averages over all time steps. We compare the proposed LB-EBM to previous works with published results on NLL. They are displayed in Table 4. Our model performs better than S-GAN [15] and Trajectron [22] but underperforms Trajectron++[1] [52]. It might be because Trajectron++ use a bivariate Gaussian mixture to model the output distribution, while our model employs a unimomal Gaussian following most previous works. Our model can also be extended to adopt Gaussian mixture as the output distribution and we leave it for future work.

	S-GAN	Trajectron	Trajectron++	Ours
ETH	15.70	2.99	1.80	2.34
Hotel	8.10	2.26	-1.29	-1.16
Univ	2.88	1.05	-0.89	0.54
Zara1	1.36	1.86	-1.13	-0.17
Zara2	0.96	0.81	-2.19	-1.58
Average	5.80	1.79	-0.74	-0.01

Table 4. NLL Evaluation on ETH-UCY for the proposed LB-EBM and baselines are shown. The lower the better.

[1] Trajectron++ is a concurrent work to ours and was discovered in the reviewing process.