Supplementary Materials for
Trajectory Prediction with Latent Belief Energy-Based Model

1. Learning

1.1. 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,

$$p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \underbrace{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})}_{\text{LB-EBM}} \overbrace{p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})}^{\text{Plan}} \underbrace{p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}_{\text{Prediction}}. \tag{1}$$

1.2. 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,

$$\sum_{j=1}^{N} \log p_\psi(\boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) = \sum_{j=1}^{N} \log \int_{\boldsymbol{Z}_j} p_\psi(\boldsymbol{Z}_j, \boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) \tag{2}$$

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,

$$\nabla_\psi \log p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \int_{\boldsymbol{Z}} p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{3}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{4}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}|\boldsymbol{X})p_\psi(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}{p_\psi(\boldsymbol{P}|\boldsymbol{X})p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{5}$$

$$= \mathrm{E}_{p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}). \tag{6}$$

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

1.3. Variational Learning

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

$$q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \tag{7}$$

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,

$$D_{KL}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{8}$$

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

$$D_{KL}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{9}$$
$$= D_{KL}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{10}$$
$$+ D_{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})) \tag{11}$$
$$\tag{12}$$

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,

$$D_{KL}(q_\phi(\boldsymbol{Z},\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z},\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})) \tag{13}$$

$$= D_{KL}(q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X}) \parallel p_\alpha(\boldsymbol{Z}|\boldsymbol{X})p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X})) \tag{14}$$

$$= \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{15}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})}{p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})} \tag{16}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}{p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X})} \tag{17}$$

Expressions 15, 16, 17 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 15 drives the learning of the LB-EBM.

$$\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{18}$$

$$= \mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}{p_0(\boldsymbol{Z})\exp[-C_\alpha(\boldsymbol{Z},\boldsymbol{X})]/Z_\alpha(\boldsymbol{X})} \tag{19}$$

$$= D_{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X}) \parallel p_0(\boldsymbol{Z})) \tag{20}$$

$$+ \mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}C_\alpha(\boldsymbol{Z},\boldsymbol{X}) + \log Z_\alpha(\boldsymbol{X}) \tag{21}$$

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,

$$\nabla_\alpha \mathcal{J}(\alpha) \tag{22}$$

$$= \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})] \tag{23}$$

$$- \mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})] \tag{24}$$

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 16 drives the learning of the plan module.

$$(16) = -\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X}) \tag{25}$$

$$- H(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X}) \tag{26}$$

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 16 is equivalent to maximizing the log-likelihood of $p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})$.

Expression 17 drives the learning of the prediction module.

$$(17) = -\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X}) \tag{27}$$

$$- H(\boldsymbol{Y}|\boldsymbol{X}) \tag{28}$$

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 27 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,

$$-\mathrm{E}_{q_{data}(\boldsymbol{X})}\mathrm{E}_{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})}\mathrm{E}_{p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X}) \tag{29}$$

where

$$\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})} \tag{30}$$

$$= \int_{\boldsymbol{P}} q_{data}(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X}) \tag{31}$$

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

2. Negative Log-Likelihood Evaluation

Although Best-of-K on ADE and FDE (e.g., $K = 20$) is widely-adopted [1, 3, 4, 7], some researchers [2, 5, 6] 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 1. Our model performs better than S-GAN [1] and Trajectron [2] but underperforms Trajectron++[1] [5]. 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.

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

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

References

[1] 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. 2

[2] 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

[3] 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. 2

[4] 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. 2

[5] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020. 2

[6] 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. 2

[7] 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. 2