Appendix for Collaborative Uncertainty in Multi-Agent Trajectory Forecasting

Bohan Tang[1] **Yiqi Zhong**[2] **Ulrich Neumann**[2] **Gang Wang**[3] **Ya Zhang**[1] **Siheng Chen**[1*]
[1]Shanghai Jiao Tong University [2]University of Southern California [3]Beijing Institute of Technology
[1]tangbohan@alumni.sjtu.edu.cn [1]{sihengc, ya_zhang}@sjtu.edu.cn
[2]{yiqizhon, uneumann}@usc.edu [3]gangwang@bit.edu.cn

A Proof of Laplace Model Design

Proof. Consider the i-th data sample $(\mathrm{X}^i, \mathrm{Y}^i)$, as in the training process of the prediction model the values of X^i and Y^i are given, $p(\mathrm{Y}^i|\mathbf{z}^i, \mathrm{X}^i; \boldsymbol{w})$ is a function of $\mathbf{z}^i \in \mathbb{R}^+$ with the probability density function: $p(\mathbf{z}^i|\mathrm{X}^i) = \frac{1}{\lambda} e^{-\frac{\mathbf{z}^i}{\lambda}}$:

$$p(\mathrm{Y}^i|\mathbf{z}^i, \mathrm{X}^i; \boldsymbol{w}) = f_{\boldsymbol{w}}(\mathbf{z}^i) = \frac{1}{(\mathbf{z}^i)^{\frac{m}{2}}} e^{-\frac{g^i_{\boldsymbol{w}}}{\mathbf{z}^i}},$$

where $g^i_{\boldsymbol{w}} = \frac{1}{2}(\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))[\Sigma^{-1}_{\boldsymbol{w}}(\mathrm{X}^i)](\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))^T$ and $m \in \mathbb{N}^+$ is the number of the agents in the i-th data sample. We need to prove that there should exist a $(\mathbf{z}^i)^* \in \mathbb{R}^+$ to make:

$$\begin{aligned}
p(\mathrm{Y}^i|\mathrm{X}^i; \boldsymbol{w}) &= \int_0^{+\infty} p(\mathrm{Y}^i|\mathbf{z}^i, \mathrm{X}^i; \boldsymbol{w}) p(\mathbf{z}^i|\mathrm{X}^i; \boldsymbol{w}) d\mathbf{z}^i \\
&= \int_0^{+\infty} f_{\boldsymbol{w}}(\mathbf{z}^i) p(\mathbf{z}^i|\mathrm{X}^i) d\mathbf{z}^i \\
&= E_{\mathbf{z}^i}[f_{\boldsymbol{w}}(\mathbf{z}^i)] \\
&= f_{\boldsymbol{w}}((\mathbf{z}^i)^*) \\
&= p(\mathrm{Y}^i|(\mathbf{z}^i)^*, \mathrm{X}^i; \boldsymbol{w}).
\end{aligned}$$

And the existence of $(\mathbf{z}^i)^*$ can be proved by proving a fact that, when $\mathbf{z}^i \in \mathbb{R}^+$, $f(\mathbf{z}^i)$ is a continuous bounded function.

As the $g^i_{\boldsymbol{w}}$ can be reformulated as:

$$\begin{aligned}
g^i_{\boldsymbol{w}} &= \frac{1}{2}(\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))[L'_{\boldsymbol{w}}(\mathrm{X}) L'^T_{\boldsymbol{w}}(\mathrm{X})](\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))^T \\
&= \frac{1}{2}[(\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))L'_{\boldsymbol{w}}(\mathrm{X})][(\mathrm{Y}^i - \mu_{\boldsymbol{w}}(\mathrm{X}^i))L'_{\boldsymbol{w}}(\mathrm{X})]^T \\
&\geq 0,
\end{aligned} \tag{1}$$

where $L'_{\boldsymbol{w}}(\mathrm{X})$ is a lower triangular matrix and the equal sign of (1) is only true when $\mu_{\boldsymbol{w}}(\mathrm{X}^i)$ is equal to Y^i, but in practice, $\mu_{\boldsymbol{w}}(\mathrm{X}^i)$ is hardly equal to Y^i, which means in the training process we have:

$$g^i_{\boldsymbol{w}} > 0. \tag{2}$$

*The corresponding author is Siheng Chen.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Based on (2), let $\mathbf{s}^i = \frac{1}{\mathbf{z}^i}$, then as $\mathbf{z}^i \to 0^+$, we have $\mathbf{s}^i \to +\infty$, so for $\mathbf{z}^i \to 0^+$:

$$\begin{aligned}\lim_{\mathbf{z}^i \to 0^+} f_{\boldsymbol{w}}(\mathbf{z}^i) &= \lim_{\mathbf{s}^i \to +\infty} (\mathbf{s}^i)^{\frac{m}{2}} e^{-\mathbf{s}^i g^i_{\boldsymbol{w}}} \\ &= \lim_{\mathbf{s}^i \to +\infty} \frac{(\frac{m}{2})!}{(g^i_{\boldsymbol{w}})^{\frac{m}{2}} e^{\mathbf{s}^i g^i_{\boldsymbol{w}}}} \\ &= 0 \end{aligned} \tag{3}$$

For $\mathbf{z}^i \to +\infty$:

$$\lim_{\mathbf{z}^i \to +\infty} f_{\boldsymbol{w}}(\mathbf{z}^i) = 0 \tag{4}$$

Furthermore, as the derivative of $f_{\boldsymbol{w}}(\mathbf{z}^i)$ is then:

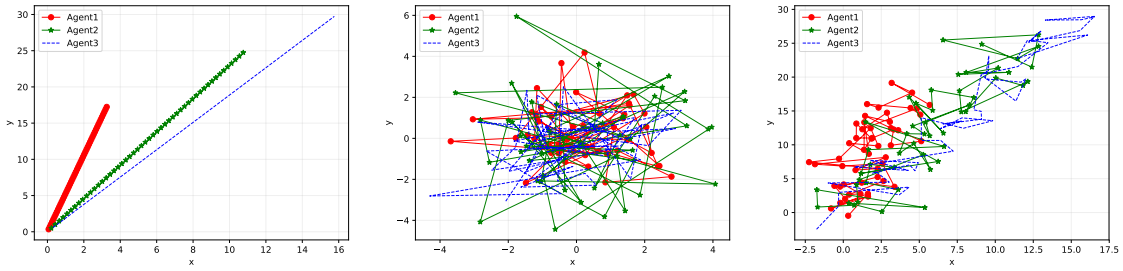
$$f'_{\boldsymbol{w}}(\mathbf{z}^i) = (\mathbf{z}^i)^{-\frac{m}{2}-2} \cdot e^{-(\mathbf{z}^i)^{-1} g^i_{\boldsymbol{w}}} \cdot (g^i_{\boldsymbol{w}} - \frac{m}{2}\mathbf{z}^i). \tag{5}$$

According to (2), (4), (3) and (5), when we set $f'_{\boldsymbol{w}}(\mathbf{z}^i) = 0$, we can get the maximum value of $f_{\boldsymbol{w}}(\mathbf{z}^i)$ is $f_{\boldsymbol{w}}(\frac{2g^i_{\boldsymbol{w}}}{m}) \in \mathbb{R}^+$.

On the basis of above discussions, when $\mathbf{z}^i \in \mathbb{R}^+$, $f(\mathbf{z}^i)$ is a continuous bounded function, which means the $(\mathbf{z}^i)^*$ is existent. □

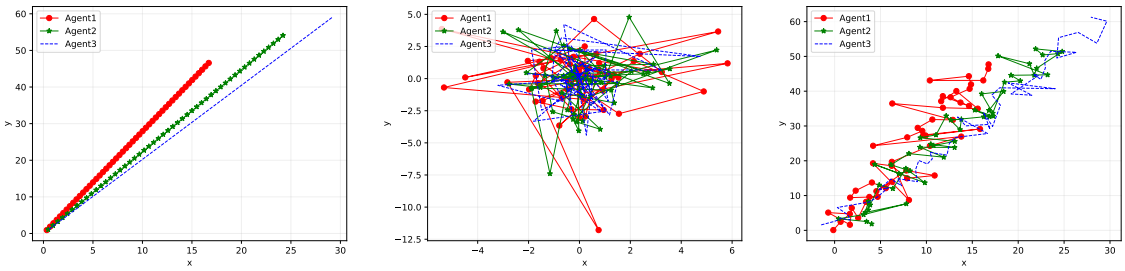
B Toy Problem

B.1 Generation Details of Synthetic Datasets



(a) μ_{gt} (b) ϵ (c) $\mathbf{x}$

Figure 1: **Sample visualization of the generation of Gaussian synthetic dataset.** $\mathbf{x}$ is the sum of μ_{gt} and ϵ.



(a) μ_{gt} (b) ϵ (c) $\mathbf{x}$

Figure 2: **Sample visualization of the generation of Laplace synthetic dataset.** $\mathbf{x}$ is the sum of μ_{gt} and ϵ.

For the Gaussian synthetic dataset, since a random variable $\mathbf{x}$ that obeys a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ can be formulated as the sum of the mean μ and a random variable ϵ: $x = \mu + \epsilon$,

where $\epsilon \sim \mathcal{N}(0, \Sigma)$. For generating the Gaussian synthetic dataset, we firstly generate 50 different two-dimensional coordinates of three agents that move in a uniform straight line. We denote this part of the data as μ_{gt}, and set it as the mean of the multivariate Gaussian distribution to which the trajectories belong. Subsequently, we sample a set of data ϵ from a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma_{gt})$ (where $\Sigma_{gt} \in \mathbb{R}^{3\times3}$), obviously this set of data contains the information of the covariance matrix of its distribution. Finally, we add the data μ_{gt} representing the mean value of the distribution and the data ϵ representing the covariance matrix information of the distribution to get our final data $\mathbf{x}$, which is $\mathbf{x} = \mu_{gt} + \epsilon$. At this time, the data $\mathbf{x}$ we get is equivalent to the data sampled from the multivariate Gaussian distribution $\mathcal{N}(\mu_{gt}, \Sigma_{gt})$. Moreover, following similar steps, we can get the Laplace synthetic dataset.

B.2 Metric Computation Details

ℓ_2 of μ is the average of pointwise ℓ_2 distances between the estimated mean and the ground truth mean. ℓ_1 of Σ is the average of pointwise ℓ_1 distances between the estimated covariance matrix and the ground truth covariance.

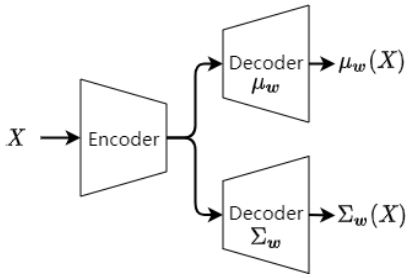
KL is the KL divergence between the ground truth distribution and the estimated distribution $D_{KL}(p_g(X)||p_e(X))$, where $p_e(X) \sim \mathcal{N}(\mu_{p_e}, \Sigma_{p_e})$ is the estimated distribution, $p_g(X) \sim \mathcal{N}(\mu_{p_g}, \Sigma_{p_g})$ is the ground truth distribution, $\Sigma_{p_e} \in \mathbb{R}^{k\times k}$ and $\Sigma_{p_g} \in \mathbb{R}^{k\times k}$. For multivariate Gaussian distribution, we compute it by the following formula (6):

$$\begin{aligned} D_{KL}(p_g(X)||p_e(X)) &= p_g(X) \int_X [\log(p_g(X)) - \log(p_e(X))] dX \\ &= \frac{1}{2}[\log(\frac{|\Sigma_{p_e}|}{|\Sigma_{p_g}|}) - k + (\mu_{p_g} - \mu_{p_e})^T \Sigma_{p_e}^{-1}(\mu_{p_g} - \mu_{p_e}) + trace\{\Sigma_{p_e}^{-1}\Sigma_{p_g}\}]. \end{aligned} \quad (6)$$

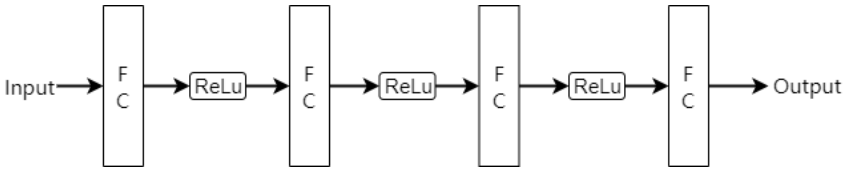
For multivariate Laplace distribution, as the probability density function of it is too complicated, when we compute the KL divergence, we firstly compute the value of $p_g(X)$ and $p_e(X)$ for each given data sample X respectively, and then we compute $D_{KL}(p_g(X)||p_e(X))$ by the following formula (7):

$$D_{KL}(p_g(X)||p_e(X)) = \sum_X p_g(X)[\log(p_g(X)) - \log(p_e(X))]. \quad (7)$$

B.3 Implementation Details



(a) Framework



(b) Four-Layer MLP

Figure 3: **The network architecture used in synthetic datasets.** (a): The framework of the used network. (b): The four-layer multilayer perceptron used to form the encoder and decoders of the used network, where FC denotes the full connected layer and ReLu denotes the ReLu activation function.

Model Structure: For toy problem, the network architecture contains an encoder and two decoders, all of which are four-layer multilayer perceptrons (MLPs), which are shown in Figure 3.

Training Details: For toy problem, we train the model on 1 GTX 1080Ti GPU using a batch size of 72 with the Adam [1] optimizer with an initial learning rate of 5×10^{-3} and the training process finishes at 36 epochs.

C Real World Problem

C.1 Implementation Details

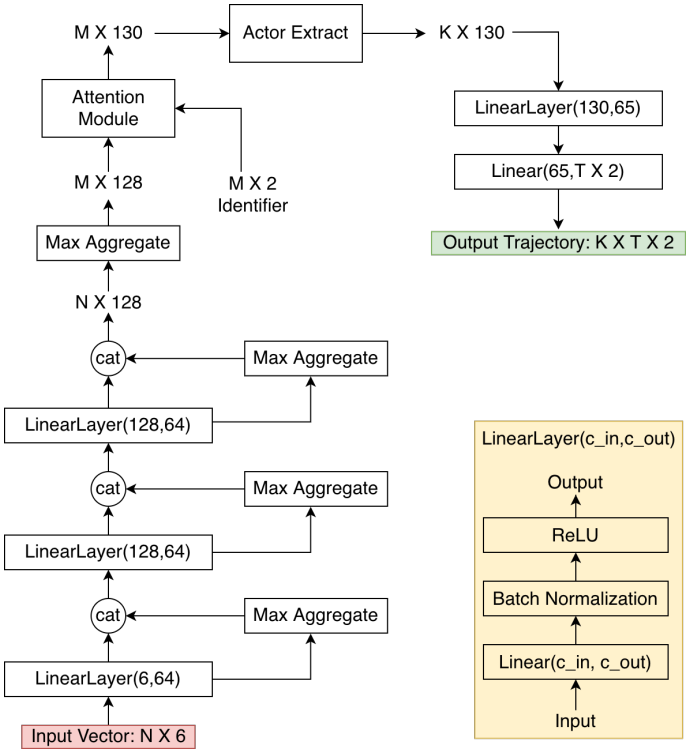


Figure 4: **Implementation structure of VectorNet.** N is the input vector number, M is the polygon number, K is the actor number.More details please refer to [2]

Model Structure: We implemented the LaneGCN according to the structure shown in the appendix of [3]. And the structure of our implemented VectorNet is shown in Figure 4.

Training Details: For LaneGCN in Argoverse, we train the model on 1 GTX 1080Ti GPU using a batch size of 64 with the Adam [1] optimizer with an initial learning rate of 2.5×10^{-4} and the training process finishes at 65 epochs. For LaneGCN in nuScenes, we train the model on 2 GTX 1080Ti GPUs using a batch size of 56 with the Adam [1] optimizer with an initial learning rate of 8×10^{-4} and the training process finishes at 90 epochs. For VectorNet in Argoverse, we train the model on 2 GTX 1080Ti GPUs using a batch size of 128 with the Adam [1] optimizer with an initial learning rate of 2×10^{-3} and the training process finishes at 105 epochs. For VectorNet in nuScenes, we train the model on 2 GTX 1080Ti GPUs using a batch size of 128 with the Adam [1] optimizer with an initial learning rate of 1×10^{-3} and the training process finishes at 500 epochs.

C.2 Additional Results

We compare our proposed approach with the approach not modeling uncertainty, which assumes the covariance Σ is an identity matrix (ID). As the results illustrated in Table 2, our proposed Laplace CU-based framework still enables LaneGCN and VectorNet to achieve the best performances on ADE & FDE metrics on both Argoverse and nuScenes benchmarks in single future prediction.

After the paper is accepted in NeurIPS 2021, we also apply our proposed framework to the official version of LaneGCN (with map information) and test it on the Argoverse benchmark in single future

Table 1: Two special cases with various assumptions about covariance Σ. **ID** denotes the identity matrix (no uncertainty). **DIA** denotes the diagonal matrix (individual uncertainty). **FULL** denotes the full matrix (individual and collaborative uncertainty).

Assumption	Two Special Cases	
	Gaussian Distribution	Laplace Distribution
ID: $\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$	$\|\mathrm{Y}-\mu_{\boldsymbol{w}}(\mathrm{X})\|_2^2$	$\|\mathrm{Y}-\mu_{\boldsymbol{w}}(\mathrm{X})\|_1$
DIA: $\begin{pmatrix} \sigma_{11} & 0 & \cdots & 0 \\ 0 & \sigma_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_{mm} \end{pmatrix}$	$\frac{1}{2}\sum_{i=1}^{m}[\sigma_{ii}^{-2}\|\mathbf{y}_i-\mu_{\boldsymbol{w}}(\mathbf{x}_i)\|_2^2+\log\sigma_{ii}^2]$	$\sum_{i=1}^{m}[\sigma_{ii}^{-2}\|\mathbf{y}_i-\mu_{\boldsymbol{w}}(\mathbf{x}_i)\|_1+\log\sigma_{ii}^2]$
FULL: $\begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \vdots & \vdots & & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mm} \end{pmatrix}$	$\frac{1}{2}[q_{\boldsymbol{w}}(\mathrm{Y},\mathrm{X})-\sum_{j=1}^{m}\log(d_{jj})]$	$\frac{1}{2}[\frac{q_{\boldsymbol{w}}(\mathrm{Y},\mathrm{X})}{\Phi_{\boldsymbol{w}}(\mathrm{X})}+m\log\Phi_{\boldsymbol{w}}(\mathrm{X})-\sum_{j=1}^{m}\log(d_{jj})]$

Table 2: Ablation on assumptions about covariance Σ of chosen probability density functions (PDFs) in single future prediction. **ID** denotes the identity matrix (no uncertainty). **DIA** denotes the diagonal matrix (individual uncertainty). **FULL** denotes the full matrix (individual and collaborative uncertainty). On Argoverse and nuScenes, a model with individual uncertainty surpasses a model without uncertainty; a model with individual and collaborative uncertainty surpasses a model with individual uncertainty only.

Dataset	Method	Assumption about Σ	Type of Chosen PDF			
			Gaussian		Laplace	
			ADE	FDE	ADE	FDE
Argoverse	LaneGCN	ID	1.52	3.32	1.44	3.17
		DIA	1.45	3.19	1.43	3.16
		FULL	**1.42**	**3.14**	**1.41**	**3.11**
	VectorNet	ID	1.67	3.62	1.59	3.44
		DIA	1.63	3.60	1.56	3.42
		FULL	**1.57**	**3.46**	**1.52**	**3.34**
nuScenes	LaneGCN	ID	4.50	10.62	4.47	10.54
		DIA	4.47	10.59	4.34	10.34
		FULL	**4.39**	**10.44**	**4.25**	**10.15**
	VectorNet	ID	4.23	9.91	4.09	9.80
		DIA	4.07	9.86	4.02	9.79
		FULL	**3.99**	**9.57**	**3.81**	**9.22**

prediction. As results shown in the Table 3, our proposed Laplace CU-based framework still enables LaneGCN to achieve the best performances on ADE & FDE metrics on both validate set and test set of Argoverse benchmarks in single future prediction.

C.3 Extra Visualization Results

Visualization of collaborative uncertainty between two agents. In Figure 5, there are 8 actor pairs (blue and orange lines) trajectories (solid lines are the past trajectories and dashed lines are the future trajectories) and their corresponding collaborative uncertainty values changing over the last 30 frames (the heatmap). Pair I to IV and Pair V to VIII are the ones w/o obvious interaction. These results show that the value of collaborative uncertainty is highly related to the amount of the interactive information among agents.

Table 3: Ablation on assumptions about covariance Σ of chosen probability density functions (PDFs) on the basis of the official version of LaneGCN in single future prediction. **ID** denotes the identity matrix (no uncertainty). **DIA** denotes the diagonal matrix (individual uncertainty). **FULL** denotes the full matrix (individual and collaborative uncertainty). On both of the validate set and the test set of Argoverse, a model with individual uncertainty surpasses a model without uncertainty; a model with individual and collaborative uncertainty surpasses a model with individual uncertainty only.

Dataset	Set	Assumption about Σ	Type of Chosen PDF			
			Gaussian		Laplace	
			ADE	FDE	ADE	FDE
Argoverse	Validate	ID	1.36	2.94	1.28	2.78
		DIA	1.32	2.90	1.27	2.76
		FULL	**1.31**	**2.89**	**1.26**	**2.75**
	Test	ID	1.69	3.72	1.64	3.61
		DIA	1.67	3.70	1.62	3.56
		FULL	**1.66**	**3.67**	**1.61**	**3.53**

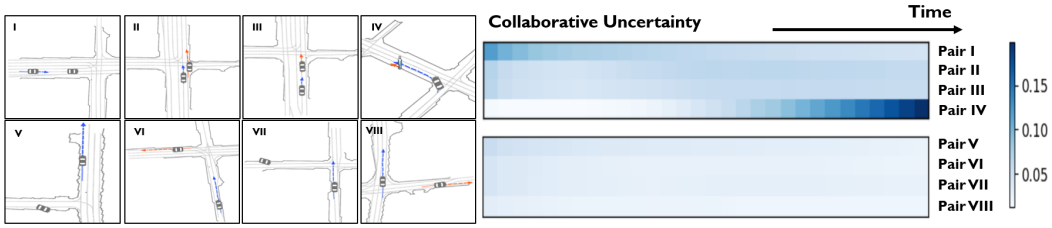


Figure 5: **Visualization of CU on Argoverse dataset**. Pair I: One agent approaching another agent parking at an intersection waiting for green light, as little new interactive information would be generated before the red light turns green, CU decreases over time. Pair II: Agents moving side by side, which might generate complicated interactive information making CU show a non-monotonic change over time. Pair III: Agents driving on the same road, which might generate complicated interactive information making CU show a non-monotonic change over time. Pair IV: Agents moving close to each other, CU increases over time. Pair V to Pair VIII: Agents located in completely different areas on the map, CUs are close to zero.

References

[1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

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

[3] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020.