Appendix of GroupNet: Multiscale Hypergraph Neural Networks for Trajectory Prediction with Relational Reasoning

1. The derivation of ELBO

In our modeling, we propose the ELBO of the log-likelihood of the future trajectories conditioned on the historical trajectories, $\log p(\mathbb{X}^+ \mid \mathbb{X}^-)$, as

$$\log p(\mathbb{X}^+ \mid \mathbb{X}^-) \geq \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)} \log p(\mathbb{X}^+ \mid \mathbf{Z}, \mathbb{X}^-) - \mathrm{KL}(q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-) \| p(\mathbf{Z} \mid \mathbb{X}^-)),$$

Here we prove the derivation of the ELBO.

Proof 1 *Let $\log p(\mathbb{X}^+ \mid \mathbb{X}^-)$ be the log-likelihood of the future trajectories $\mathbb{X}^+$ conditioned on the past trajectories $\mathbb{X}^-$ and $q(\mathbf{Z}|\mathbb{X}^+, \mathbb{X}^-)$ be the approximate posterior of $\mathbf{Z}$. The derivation of the corresponding evidence lower bound (ELBO) in Section 5 in the main text is:*

$$\begin{aligned}
&\log p(\mathbb{X}^+ \mid \mathbb{X}^-) \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}[\log p(\mathbb{X}^+ \mid \mathbb{X}^-)] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}[\log \frac{p(\mathbb{X}^+, \mathbf{Z} \mid \mathbb{X}^-)}{p(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}\big[\log[\frac{p(\mathbb{X}^+, \mathbf{Z} \mid \mathbb{X}^-)}{q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)} \cdot \frac{q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}{p(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}]\big] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}[\log \frac{p(\mathbb{X}^+, \mathbf{Z} \mid \mathbb{X}^-)}{q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}] \\
&\quad + \mathrm{KL}\left(q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-) \| p(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)\right) \\
&\geq \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}[\log \frac{p(\mathbb{X}^+, \mathbf{Z} \mid \mathbb{X}^-)}{q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)}\big[\log[p(\mathbf{Z} \mid \mathbb{X}^-) \cdot \frac{p(\mathbb{X}^+ \mid \mathbf{Z}, \mathbb{X}^-)}{q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-)}]\big] \\
&= \mathbb{E}_{q(\mathbf{Z}|\mathbb{X}^+,\mathbb{X}^-)} \log p(\mathbb{X}^+ \mid \mathbf{Z}, \mathbb{X}^-) \\
&\quad - \mathrm{KL}\left(q(\mathbf{Z} \mid \mathbb{X}^+, \mathbb{X}^-) \| p(\mathbf{Z} \mid \mathbb{X}^-)\right)
\end{aligned}$$

In this way, we obtain the ELBO that needs to be maximize.

2. Further experimental details

In this section, we provide further details of experiments, which include dataset introduction, baseline approaches, implementation details as well as the supplementary explanation for the figure in the main text.

2.1. Datasets

2.1.1 Simulation dataset

Ability to capture group behaviors. In this simulation, we have 6 particles in an x-y plane forming a three-particle group connected by a Y-shape light bar, a two-particle group connected by a spring and an individual particle without any connection. Initially, the center of the Y-shape light bar, the center of the spring and the individual particle are randomly initialized on the area $\{(x, y)|x \in [-10, 10], y \in [-10, 10]\}$. The distance between particles connected by the light bar and the center of the light bar keeps the same, initializing in the range $[1, 3]$ and the three particles are uniformly distributed around the center of the light bar, keeping the angle between two particles is $120°$. The translational velocity and the angular velocity of the light bar are initialized ranging in $[0, \sqrt{2}]$ and $[\pi/6, \pi/3]$, respectively. The two particles connected by the spring will do the simple harmonic motion because of Hooke's law. The frequency of the simple harmonic motion is randomly initialized ranging in $[6, 20]$. The individual particle has a constant velocity initialized ranging in $[0, \sqrt{2}]$. We predicted the particle states at the future 10 timestamps based on the observations of 10 timestamps and generated 50k samples in total for training and testing.

Ability to reason interaction category. In this simulation, we have 3 particles in an x-y plane forming one group connected by a light bar, springs or nothing. Initially, the center of the Y-shape light bar, the center of springs and the individual particles are randomly initialized on the area $\{(x, y)|x \in [-10, 10], y \in [-10, 10]\}$. In the type of 'light bar', the distance between the center of the light bar and any particle is initialized ranging in $[1, 5]$. The translational velocity and the angular velocity of the light bar are initialized ranging in $[0, \sqrt{2}]$ and $[\pi/10, \pi/3]$, respectively. In the type of 'spring', three springs have the same initial length initialized ranging in $[1, 5]$ and particles' angular velocity is initialized ranging in $[10, 30]$. In the type of 'free', the individual particle has a constant velocity initialized ranging in $[0, \sqrt{2}]$. We predicted the particle states at the future 10 timestamps based on the observations of 10 timestamps and generated 50k samples in total for training and testing.

Ability to reason interaction strength. In this simulation, we have 2 charged particles in an x-y plane. The charged particles interact under Coulomb force:$F = C \cdot sign(q_1 \cdot q_2)\frac{(r_1 - r_2)}{||r_1 - r_2||^3}$, where C is a constant and q_1, q_2 is the charged quantity r_1, r_2 is the position. One particle is fixed at the origin point, carrying random initialized positively charged quantity ranging in $[1, 15]$. Another particle has an initialized velocity $v = (1, -1)$ with a fixed positive charged quantity of 1. The constant C is set to be 0.05. The large amount quantity the fixed particle carries, the large repulsive force the moving particle receives, indicating a larger interaction strength. We predicted the particle states at the future 10 timestamps based on the observations of 10 timestamps and generated 5k samples in total for training and testing.

2.1.2 NBA dataset

The NBA SportVU dataset contains player and ball trajectories from the 2015-2016 NBA season collected with the SportVU tracking system. The raw tracking data is in the JSON format, and each moment includes information about the identities of the players on the court, the identities of the teams, the period, the game clock, and the shot clock. Following the protocol in [8], we selected 50k samples in total for training, validation and testing with a split of 65%, 10%, 25%. Each sample contains the historical 8 timestamps (3.2s) and future 12 timestamps (4.8s).

2.1.3 SDD dataset

The Stanford Drone dataset (SDD) consists of 20 scenes captured using a drone in top-down view around the university campus containing several moving agents like humans and vehicles. The coordinates of multiple actors' trajectories are provided in pixels. we use 0.4s as the time interval, and use the first 3.2 seconds (8 timestamps) to predict the following 4.8 seconds (12 timestamps). We use the standard test train split as used in previous works [3,9,12] and include all agent types (pedestrians, cyclists, vehicles).

2.1.4 ETH-UCY dataset

ETH-UCY dataset contains 5 subsets, ETH, HOTEL, UNIV, ZARA1 and ZARA2. They consist of pedestrian trajectories captured at 2.5Hz in multi-agent social scenarios. Following the experimental setting in [3, 13], we split the trajectories into segments of 8s, where we use 0.4s as the time interval, and use the first 3.2 seconds (8 timestamps) to predict the following 4.8 seconds (12 timestamps). We use the leave-one-out approach, training on 4 sets and testing on the remaining set.

2.2. Baseline methods

We compared the performance of our proposed approach with the following baseline methods.

2.2.1 For Synthetic physical Simulations

• Corr.(path): The baseline method in [7] to calculate a correlation matrix between all trajectories and using an ideal threshold to judge whether particles having connections. It is only used for 2-type interaction category recognition since the limitation of thresholding methods.

• Corr.(LSTM): The baseline method in [7] to train a two-layer LSTM with an MSE loss and apply the correlation matrix procedure and threshold on the output of second LSTM layers at the last time step. It is only used for 2-type interaction category recognition since the limitation of thresholding methods.

• NRI: The neural relational inference model [7] which learns a static latent interaction graph. The inferred edge types correspond to a clustering of the interactions.

2.2.2 For real-world datasets

• STGAT [5]: The model is a variant of graph attention network and captures the temporal interactions with an additional LSTM.

• Social-STGCNN [10]: The model proposes a variant of graph convolutional neural network and substitutes the need of aggregation methods by modeling the interactions as a graph.

• Social-Attention [14]: The model formulates the trajectory sequence in a spatial-temporal graph to capture the spatial and temporal dynamics.

• Social-LSTM [1]: The model encodes the trajectories with an LSTM layer whose hidden states serve as the input of a social pooling layer.

• Social-GAN [3]: The model leverages adversarial learning to fit the uncertain human behavior and pools the hidden state with all the other actors involved in the scene.

• Trajectron++ [13]: The approach uses a graph-structured recurrent model and a framework based on conditional variational auto-encoder with the InfoVAE objective function.

• NRI [7]: The neural relational inference model with latent graph re-evaluation at each time step.

• EvolveGraph [8]: The generic framework with explicit relational structure recognition via dynamic latent interaction graphs among heterogeneous agents.

• PECNet [9]: The approach addresses human trajectory prediction by modeling intermediate stochastic goals of endpoints. It uses a novel self-attention based social pooling layer to model the interactions.

• CF-VAE [2]: The approach based on conditional normalizing flow based priors in order to model complex multimodal conditional distributions over sequences.

• SOPHIE [12]: The model proposes a GAN employing attention on social and physical constraints discriminatively considering the impact of other actors to produce human-like motion.

• STAR [15]: The model uses a spatio-temporal graph transformer framework tackling trajectory prediction by only attention mechanisms within the core of transformer-based graph convolution mechanism.

• NMMP [4]: The model uses neural motion message passing infers an interaction graph from agents' trajectories and learns representations for directed interactions between actors.

2.3. Implementation Details

We implement our method with Pytorch [11] deep learning frameworks and the model are trained on a single NVIDIA 3090-TI GPU. We clip the maximum value of the KL divergence down to 2. For NBA dataset, the dimension of the agent feature and the interaction feature at a single scale is $d = 16$, the dimension of the latent code $\mathbf{Z}$ is $d_z = 16$. For SDD and ETH-UCY dataset, the dimension of the agent feature and the interaction feature at a single scale is $d = 64$, the dimension of the latent code $\mathbf{Z}$ is $d_z = 64$. For ETH-UCY dataset, we use a learnable latent code distribution $\mathcal{N}(\mu_q, \sigma_q)$ to replace the standard Gaussian distribution $\mathcal{N}(0, I)$. The μ_q and σ_q are learned from the agent past features $\mathbf{V}^-$ through MLPs. The structure details of feature extraction modules and prediction modules in the system are listed below:

• $\mathcal{F}_\mu$: A four-layer MLP with hidden dimensions of $[512, 256]$ with ReLU non-linearity function.

• $\mathcal{F}_\sigma$: A four-layer MLP with hidden dimensions of $[512, 256]$ with ReLU non-linearity function.

• GRU in residual block: A one-layer GRU with hidden dimensions of 96.

• MLP in residual block: A four-layer MLP with hidden dimensions of $[512, 256]$ with ReLU non-linearity function.

For the synthetic simulation dataset, we use Adam optimizer [6] with the learning rate of 0.001 for 100 epochs. We use two GroupNet one for obtaining the latent distribution and one for obtaining the past feature concatenated with latent code $\mathbf{Z}$. We use the latter to do the relational reasoning. For the validation of capturing group behaviors, we set the interaction category number to be 2 and the scales to be 2 and 3 particles. For the validation of the reasoning interaction category, we set the interaction category number to be 3 and the scales to be 3 particles. For the validation of reasoning interaction strength, we set the interaction category number to be 1 and the scales to be 2 particles. For the real-world datasets, we use Adam optimizer with the initial learning rate of 0.001 and decay $1/10$ per 10 epochs. For the ETH-UCY dataset, we consider a maximum neighbouring agent number of 8. When we apply the GroupNet into previous frameworks, we follow the original training strategy.

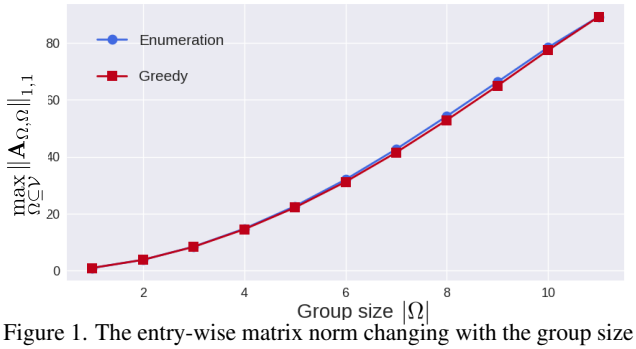


Figure 1. The entry-wise matrix norm changing with the group size based on two optimization algorithms, 1) the enumeration method and 2) the greedy approximation, for the hyperedge forming.

Table 1. Performance of different solving methods for hyperedge forming on the NBA dataset. We report minADE_{20} / minFDE_{20} (Meters).

	Prediction time			
Solving method	1.0s	2.0s	3.0s	4.0s
Enumeration	0.35/0.48	0.63/0.95	0.88/1.31	1.13/1.70
Greedy	0.34/0.48	0.62/0.95	0.87/1.31	1.13/1.69

Table 2. Ablation studies of different iteration numbers in the hyperedge neural message passing on the NBA dataset. We report minADE_{20} / minFDE_{20} (Meters).

	Prediction time			
Iteration	1.0s	2.0s	3.0s	4.0s
1	0.35/0.49	0.62/0.95	0.88/1.32	1.13/1.70
2	0.34/0.48	0.62/0.95	0.87/1.31	1.13/1.69
3	0.35/0.49	0.62/0.95	0.88/1.31	1.13/1.69
4	0.35/0.49	0.62/0.95	0.87/1.30	1.13/1.71

3. Further Quantitative Results

3.1. Greedy approximation in the hyperedge forming

We compare the two methods for solving the optimization problem in (2) in our main text: the enumeration algorithm to search for the optimum solution and a greedy approximation algorithm that adding new nodes sequentially by maximizing entry-wise matrix norm greedily. To see the performance gap between the two methods, Figure 1 compares the entry-wise matrix norm as a function of node selection with two algorithms on the NBA dataset. We see that the greedy approximation has a close hyperedge forming performance with the enumeration algorithm in maximizing entry-wise matrix norm values. Table 1 shows the comparison of the prediction result of two algorithms on NBA datasets. We see that two algorithms have similar performance thus we

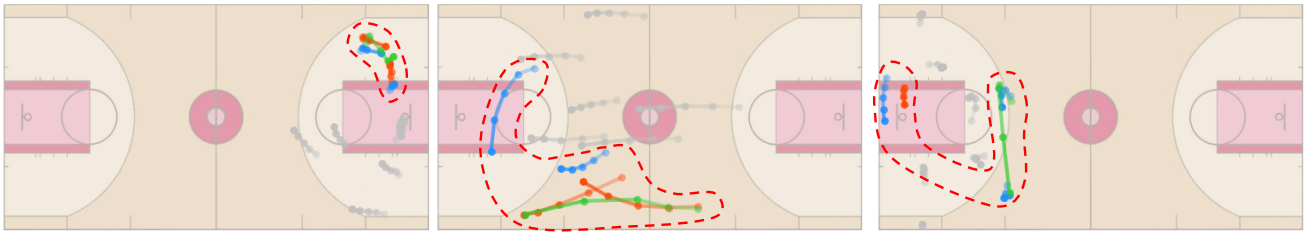
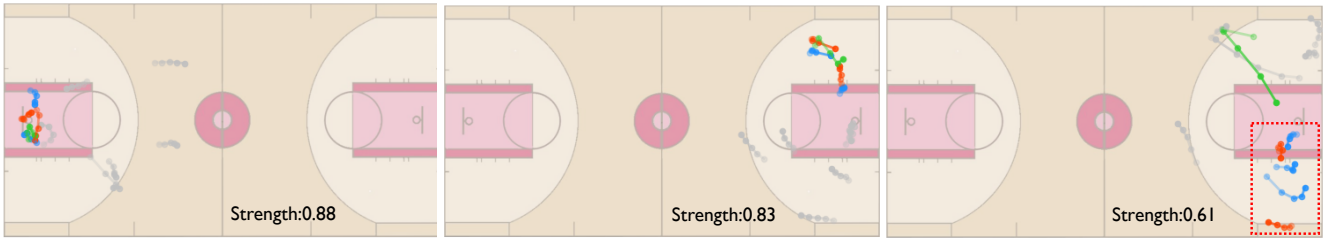


Figure 2. The visualization result of captured group behaviors of 5-player groups on the NBA dataset. The group is presented by the red dotted line. Players in the different teams are colored in red and blue, respectively; besides, the ball is colored in green. Players outside the group are colored in gray.



(a) (b) (c)

Figure 3. Group behaviors under the same neural interaction category but different neural interaction strength. Players in the different teams are colored in red and blue, respectively; besides, the ball is colored in green. Players outside the group are colored in gray. Specifically, in (c) players in the group are in the rectangle and we color the ball additionally for better understanding.

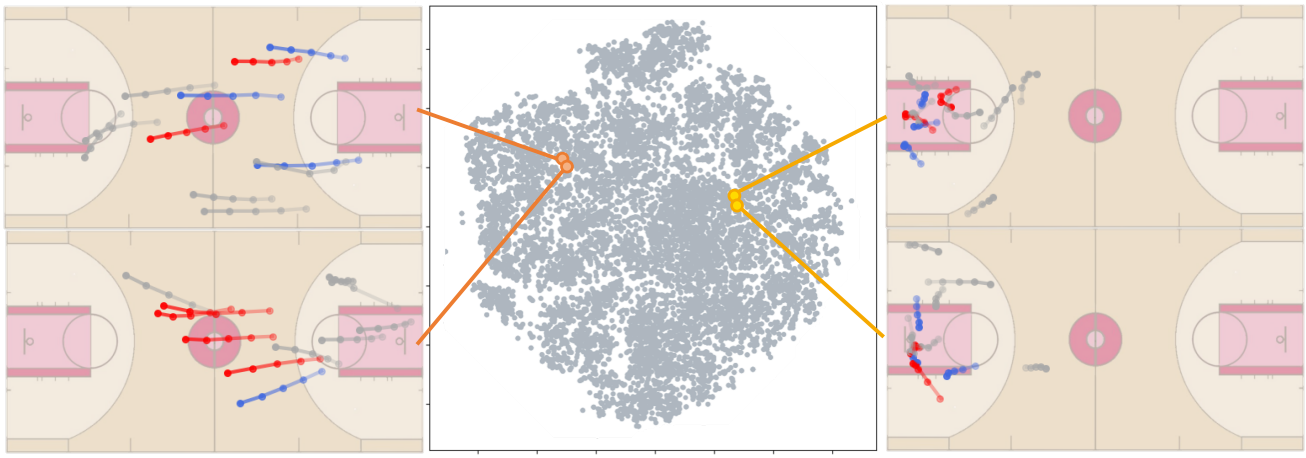


Figure 4. Close group-wise interaction embeddings in the feature domain lead to similar group behaviors in the spatial domain. Each of orange dot and yellow dot represents an interaction and corresponds to one spatial plot in the left or right columns. Players in the different teams are colored in red and blue, respectively and players outside the group are colored in gray.

choose the greedy algorithm as an approximation method to solve the optimization problem (2) in our main text when the total number of agents is large.

3.2. Effects of iteration numbers

Table 2 shows the effects of different iteration numbers for which the hypergraph neural message passing is executed. We see that different iteration numbers have close performance in general and a moderate iteration number achieves the best results.

4. Further Qualitative Results

Here we present more qualitative results on the real-world NBA dataset to reflect the group behavior, neural interaction intensity and interaction embedding on the real-world scenario.

4.1. Visualization of group behavior

Figure 2 shows three examples of our learnt hyperedges at the group size of five players. In the captured group, players in the different teams are colored in red and blue, respec-

tively; besides, the ball is colored in green. Players outside the group are colored in gray. We use dotted lines representing hyperedges. We see that our method could qualitatively capture both short-range and long-range group interactions, including behaviors of confrontation and chasing the ball.

4.2. Visualization of neural interaction strength

Figure 3 presents different group interactions under the same neural interaction category but different neural interaction strength. Figure 3(a)(b)(c) gives three examples of confrontation on the restricted area near the basket, confrontation close to the painted area and confrontation of waiting for the ball, indicating a high to low interaction strength. We see that the more fierce confrontation among the 5 players happens, the higher neural interaction strength is, indicating our MS-HGNN captures the interaction strength matching with human intuition.

4.3. Visualization of interaction embedding

Figure 4 shows the visualization of group interactions in the embedding space and the corresponding groups in the spatial domain in the NBA dataset. The hyperedge embedding $\mathbf{e}_i$ is mapped to 2D coordinates via t-SNE and shown in the middle column. We randomly pick two pairs of close interaction samples representing the five-player groups in the embedding space, which are colored orange and yellow, and plot the corresponding trajectories of group members in the left and right columns. The lighter color denotes the more previous timestamps and the blue/red color denotes two teams' players. The irrelevant players not containing in the hyperedge are colored in gray. We see that (i) close group-wise interaction embeddings in the embedding domain lead to similar group behaviors in the spatial domain. For example, in the left column, players are running through the half court; and (ii) far-away group-wise interaction embeddings in the embedding domain reflect the corresponding group behaviors are clearly different in the spatial domain. For example, both orange dots represent the group behavior of running through the half court and both yellow dots represent the group behavior of laying-up and defense; while two orange dots are far from two yellow dots.

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

[2] 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. 3

[3] 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

[4] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6319–6328, 2020. 3

[5] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6272–6281, 2019. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[7] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018. 2

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

[9] 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. In *European Conference on Computer Vision*, pages 759–776. Springer, 2020. 2

[10] 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. 2

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 3

[12] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 2, 3

[13] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. 2020. 2

[14] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. 2

[15] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020. 3