ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448062

MAPFLOW: MULTI-AGENT PEDESTRIAN TRAJECTORY PREDICTION USING NORMALIZING FLOW

Antonio Luigi Stefani, Niccolò Bisagno, Nicola Conci

University of Trento, Department of Information Engineering and Computer Science, Italy
{antonioluigi.stefani, niccolo.bisagno, nicola.conci}@unitn.it

ABSTRACT

In the task of pedestrian trajectory prediction, multi-modal prediction has recently emerged, demonstrating how a good model should predict multiple socially acceptable futures.With this respect, Normalizing Flows (NFs) have shown remarkable generative capabilities that make them particularly suitable for multi-modal trajectory prediction. By sampling from the learned distribution, NFs can produce multiple socially acceptable trajectories, each one paired with its corresponding likelihood score. Taking advantage of the multi-modal prediction coupled with the likelihood score, with MapFlow we introduce a solution based on NFs that improves the accuracy in prediction by incorporating in the model the social influence of neighboring pedestrians. [1]

Index Terms— Trajectory prediction, Normalizing Flow.

1. INTRODUCTION

Trajectory prediction of pedestrians has been widely studied in recent years [1–3]. Applications such as video surveillance [4], autonomous driving [5], and social-robot navigation [6] deeply rely on the prediction of pedestrian behaviors.

When dealing with this problem it is key to ensure that the predicted data are *socially acceptable* [7]. In fact, a trajectory can be *physically* possible but *socially* unacceptable, for example when invading the social space of the neighbouring subjects. To ensure the social acceptability of predicted trajectories, recent works [7] have exploited multi-modal prediction. While the early approaches in the literature only provided one solution to the prediction problem [1] the current trend in the state-of-the-art concentrates on generative models like Generative Adversarial Networks (GANs), Variational AutoEncoders (VAEs), Diffusion Models and Normalizing Flows (NFs).

The multi-modal generative models differ in how the learning mechanism reproduces the distribution that describes the data. GANs [7, 8] use a competitive strategy between generator and discriminator: however, they suffer from mode collapsing and it is not guaranteed their convergence while learning the real data distribution. VAEs [9, 10] learn from an encoded latent space to reconstruct the future position using an ad-hoc decoder. Diffusion models [11, 12] can jointly learn the distribution of multiple pedestrians in the scene, iteratively refining a noisy sample starting from pure Gaussian noise, thus being able to model complex distributions. Normalizing Flows [3, 13, 14] use a set of invertible transformations to map a complex distribution onto a simpler one. Using an easily invertible element-wise transformation, NFs provide accurate density estimation and sampling. While both GANs and VAEs-based approaches have vastly explored the influence of the social aspect given by neighbor pedestrians, NFs still lack thorough exploitation of the social aspects of the scenes.

Inspired by the work by Schöller et al. in [3], we present a novel model based on Normalizing Flows, which considers not only the motion of a single pedestrian to predict its trajectory but also the social contextual information provided by the observed trajectories of other pedestrians in the scene. We show how exploiting the social information allows reaching state-of-the-art results on the trajectory prediction task on common datasets [15, 16], while compressing the size of the network and obtaining a more socially-acceptable set of predicted trajectories.

The contribution of our work can be summarised as:

- we introduce a novel module in the Normalizing Flow to improve the trajectory prediction task that leverages the social dimension;
- we provide a novel embedding auto-encoder to reduce the size of the network and of the latent space for the NF. As demonstrated in [17], it has been established that this leads to improved performance of NFs;
- we achieve results superior or on-par with the state-of-the-art, on the common UCY/ETH datasets baseline, with a set of predicted trajectories that are more socially acceptable.

[1] The code can be found at github.com/mmlab-cv/MapFlow

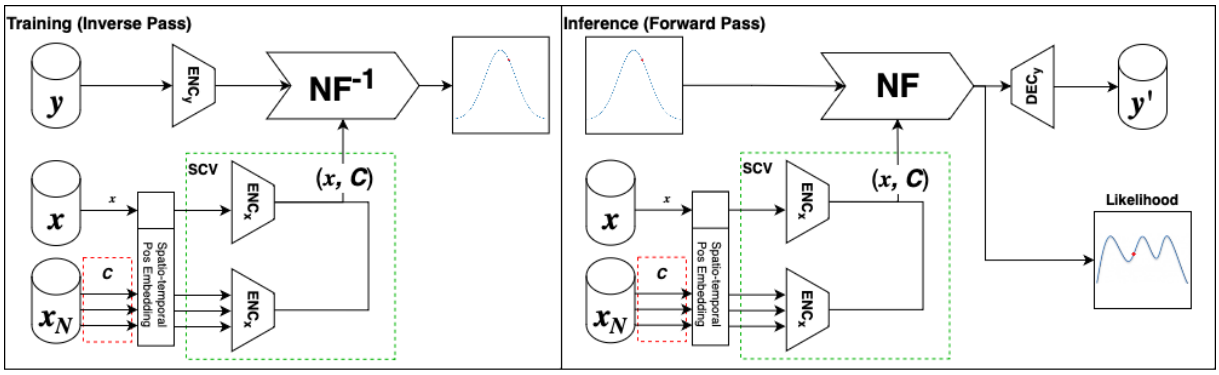

Fig. 1: Overview of the proposed architecture. In red the context vector is highlighted, which is combined with the spatio-temporal embedding to obtain our novel Social Context Vector (SCV), highlighted in green. During the training phase, the model learns to map the ground truth distribution to be predicted into a Gaussian. During the inference phase, we sample the Gaussian, which gives us a predicted trajectory and its likelihood. Both in training and inference, our SCV acts as a conditioning vector on the Normalizing Flow, describing both the past trajectory of the pedestrian of interest and the social context around it.

2. METHOD

2.1. Problem definition

The problem of pedestrian *trajectory prediction* can be summarised as the prediction of future points on a trajectory starting from a set of observed positions. Let's consider the motion of an agent α represented as a sequence of consecutive points: $\phi_\alpha = (\boldsymbol{P_0}, ..., \boldsymbol{P_T})$, where each point $\boldsymbol{P_i}$ has coordinates (p_x, p_y). As commonly done in the literature, the trajectory ϕ is divided into two segments: an observed segment $\boldsymbol{x} = (\boldsymbol{P_0}, ..., \boldsymbol{P_t})$, and a predicted segment $\boldsymbol{y} = (\boldsymbol{P_{t+1}}, ..., \boldsymbol{P_T})$. A context vector $\boldsymbol{C}$ describes the state of the observed scene.

Stochastic models aim at estimating the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{C})$ from the real data, so that sampling $\hat{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{C})$ produces a prediction of the future trajectories $\hat{\boldsymbol{y}} \sim \hat{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{C})$.

2.2. Preliminaries on Normalizing Flows

The core idea of Normalizing Flows is to employ a set of stacked invertible and differentiable transformations to map a complex distribution into a simpler one, typically a Gaussian distribution. This means that Normalizing Flows can be used in two ways: *forward* and *inverse*.

The *forward pass* takes a sample from the simpler distribution and maps it onto the complex one, while the *inverse pass* takes a sample from the complex distribution and maps it onto the simpler distribution.

In the context of trajectory prediction, the goal is to learn the distribution $p(\boldsymbol{y}|\boldsymbol{x})$ presented in Sec. 2.1. To do so, NFs apply the inverse direction of the flow, defined as f^{-1}, to real trajectories. Therefore, they map all the samples $\boldsymbol{y}$ onto samples $\boldsymbol{u}$ of a Gaussian distribution $p_u(\boldsymbol{u})$.

$$\boldsymbol{u} = f^{-1}(\boldsymbol{y}) \; where \; \boldsymbol{u} \sim p_u(\boldsymbol{u}) \tag{1}$$

Hence, we can define the following change of variables:

$$p_u(\boldsymbol{u}) = p_y(\boldsymbol{y}) |det \; J_{f^{-1}}(\boldsymbol{y})|^{-1} \tag{2}$$

with $J_{f^{-1}}(\boldsymbol{y})$ be the Jacobian matrix of the function. The forward pass instead, can be obtained by reversing Eq. 2:

$$p_y(\boldsymbol{y}) = p_u(\boldsymbol{u}) |det \; J_f(\boldsymbol{u})|^{-1} \tag{3}$$

Joining Eq. 3 with Eq. 1 we obtain:

$$p_y(\boldsymbol{y}) = p_u(f^{-1}(\boldsymbol{y})) |det \; J_{f^{-1}}(\boldsymbol{y})|. \tag{4}$$

2.3. Model

Embedding. We define an autoencoder structure to embed the trajectory data into the NF. Using an autoencoder allows us to reduce the dimensions of the representation the NF must learn, simplifying the distribution mapping task.

The Encoder ENC is used to embed a trajectory into a vector of size L. First, each trajectory passes through a linear layer of size L with ReLu activations. Then, each element is iteratively fed to a sequence of G GRU (Gated Recurrent Units) functions to learn the temporal features. The output of the GRUs corresponds to the hidden state at the last iteration. The final output of the GRUs is embedded in a linear layer of size L similar to the initial one. The Decoder DEC applies the opposite process, passing from the latent representation L to the trajectory.

Using the structure described above, we define two different encoders ENC_x and ENC_y. ENC_x is used to embed the conditional trajectories $\boldsymbol{x}$ and $\boldsymbol{C}$ present in the scene; ENC_y is used to embed into a latent representation the ground truth $\boldsymbol{y}$, and it is coupled with a decoder DEC_y.

ENC_x and ENC_y are trained separately. ENC_y and DEC_y are trained offline as an autoencoder to reconstruct to input ground truth trajectory. The latent representation of the

autoencoder is used as an input to the NF instead of the raw data. Differently, ENC_x is trained jointly with the NF, thus no decoder is needed.

Normalizing Flows. Similarly to the work in [3], our model uses the coupling layers (CLs) framework, as follows:

- An input trajectory ϕ is split into two parts: $\boldsymbol{x} = \phi_{0:t}$, $\boldsymbol{y} = \phi_{t+1:T}$ with $\boldsymbol{x}$ the observed part of the trajectory and $\boldsymbol{y}$ the part to be predicted.
- While $\boldsymbol{x}$ is used only as a conditioner, CLs aim at learning the distribution of $\boldsymbol{y}$. Therefore, $\boldsymbol{y}$ is split into two parts of the same dimensions $\boldsymbol{u}_A = \boldsymbol{y}_{0,\dots,K/2}$ and $\boldsymbol{u}_B = \boldsymbol{y}_{(K/2)+1,\dots,K}$.
- Considering only $\boldsymbol{u_A}$, the parameters $\boldsymbol{\theta}$ of an arbitrary neural network are computed: $\boldsymbol{\theta} = NN(\boldsymbol{u_A})$.
- Exploiting $\boldsymbol{\theta}$, the output $\hat{\boldsymbol{y}}$ is computed according to an invertible function g: $\hat{u}_i = g_{\boldsymbol{\theta}_i}(u_{B_i}) \;\; for\; i = (K/2)+1, ..., K$.
- The output $\hat{\boldsymbol{y}}$ is reconstructed considering two parts: $\hat{\boldsymbol{y}}_A = \boldsymbol{u}_A$ and $\hat{\boldsymbol{y}}_{K/2+1:K} = \hat{\boldsymbol{u}}_{K/2+1:K}$.

The function g is a transformation applied element-wise.

Like other approaches in the domain of trajectory prediction [3, 14], our flow is based on monotonic rational-quadratic functions to express the transformation g, introduced in Neural Spline Flow (NSF) [18].

Social context vector (SCV). The main contribution of our work is the design of a social context vector (SCV) as the conditional vector $(\boldsymbol{x}, \boldsymbol{C})$ of the Normalizing Flow. Doing so, it allows the NF to be conditioned by not only the past positions $\boldsymbol{x}$ of the trajectory $\boldsymbol{y}$ we want to predict, but also by the N trajectories $x_1, \dots, x_N$ belonging to neighbor pedestrians in the scene, thus effectively modeling the social interactions of the person of interest. The complete conditional vector $(\boldsymbol{x}, \boldsymbol{C})$ of the NF is made of the concatenation of $\boldsymbol{x}$ and $\boldsymbol{C}$, where the context vector is expressed as $C = x_1 \frown x_2, \dots, \frown x_N$. Each of the multiple trajectories of neighbor pedestrians, as well as the trajectory of the pedestrian of interest $\boldsymbol{x}$ is embedded using ENC_x.

To let the NF interpret correctly the spatiotemporal distribution of the data contained in the conditional vector $(\boldsymbol{x}, \boldsymbol{C})$, we deploy two learnable embeddings: a spatial and a temporal one. The spatial embedding enables the network to capture the relationship between the past trajectory and each of the neighboring pedestrians (the vector representing the movements of neighbor pedestrians is fed into the network from the closest to the furthest neighbor). The temporal embedding allows the network to learn the temporal relations, such that the first point of all trajectories is recorded at the same time step, as well as the relationship between time-wise consecutive points.

Trajectory Augmentation. To increase the accuracy of our model we perform data augmentation scaling the trajectories' dimension for a coefficient λ. This allows us to simulate different velocities. λ is sampled from a truncated Gaussian distribution with boundaries chosen to avoid not realistic velocities.

Trajectory prediction The overview of the architecture for the trajectory prediction task is shown in Fig. 1. The network is trained on real trajectories, with $\boldsymbol{y}$ being the ground truth part of the trajectory we need to predict, $\boldsymbol{x}$ being the observed part of the trajectories and C being the context describing other pedestrians in the scene. When the length of the trajectory of other neighbor pedestrians in the scene does not match the observed part of the trajectories, padding is applied using the last available position.

At training time the network is trained using the so-called inverse pass NF^{-1} of the Normalizing Flows, thus minimizing the distance between the likelihood of each ground truth sample and the center of the estimated Gaussian describing the likelihood distribution. This minimizes the following negative log-likelihood loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} log(p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{C})) \tag{5}$$

with N being the number of samples in a batch.

At inference time, we deploy the Normalizing Flow in its forward pass NF. Thus, we sample the learned Gaussian multiple times to obtain a multimodal prediction $\hat{\boldsymbol{y}}$. The social context vector is constructed in the same way as at training time. The resulting latent representation is decoded with the DEC_y to obtain the final prediction.

3. RESULTS

For our experiments we adopt the ETH/UCY datasets [15, 16] using the standard leave-one-out approach. We evaluate our model according to the standard metrics ADE and FDE, following the setup described in [3]. In particular, we divide the trajectories, using 8 points for observation (3.2 seconds) and 12 points for prediction (4.8 seconds).

We trained our model using the following hyperparameters: a batch size of 128 and a learning rate of 0.001 for 150 epochs.

The quantitative results for the trajectory prediction task are shown in Tab. 1. Our model outperforms other state-of-the-art approaches in most scenes, with the SCV component helping improve the results in the scenarios where the social interactions are more fixed, like in *eth-uni*, where the lane formation and emergent behaviors are especially consistent over time. The improvement is evident with respect to the vanilla FloMo [3]. In other scenes, like *uni* and *zara1* where the movements are more chaotic and crowded, our model performs similarly to the Trajectron++ [2] which leverages the

Method	eth-uni		hotel		uni		zara1		zara2		avg	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
FloMo [3]	0.32	0.52	0.15	0.22	0.25	0.46	0.20	0.36	0.17	0.31	0.22	0.37
S-STGCNN [19]	0.64	1.11	0.49	0.85	0.44	0.79	0.34	0.53	0.30	0.48	0.44	0.75
GraphTCN-G [20]	0.39	0.75	0.18	0.33	0.30	0.60	0.20	0.39	0.16	0.32	0.25	0.48
Trajectron++ [2]	0.39	0.83	**0.12**	0.21	**0.20**	**0.44**	**0.15**	0.33	**0.11**	**0.25**	**0.19**	0.41
Social LSTM [1]	0.73	1.48	0.49	1.01	0.41	0.84	0.27	0.56	0.33	0.70	0.45	0.92
Social Gan [7]	0.59	1.04	0.38	0.80	0.27	0.49	0.18	0.33	0.19	0.35	0.32	0.60
MapFlow (Ours)	**0.29**	**0.44**	***0.13***	**0.19**	0.28	0.50	***0.18***	**0.31**	0.17	***0.30***	***0.21***	**0.35**

Table 1: Quantitative results for the trajectory prediction task. We report all the results in meters. Compared to all other methods in the table, Trajectron++ method also embeds prior knowledge about the environment layout. In **bold** are highlighted the best results, in ***bold-italic*** the best/on-par results without the environment prior.

Ablation	N	L	Data Aug	Spatio-temporal embedding	eth-uni		hotel		uni		zara1		zara2		avg	
					ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
MapFlow	**3**	16	Yes	Yes	0.29	0.44	0.13	0.19	0.28	0.50	0.18	0.31	0.17	0.30	**0.21**	**0.35**
(I)	**5**	16	Yes	Yes	0.32	0.49	0.13	0.19	0.26	0.49	0.19	0.34	0.17	0.30	0.21	0.36
(II)	**10**	16	Yes	Yes	0.35	0.52	0.13	0.20	0.27	0.50	0.18	0.30	0.17	0.29	0.22	0.36
(III)	3	**8**	Yes	Yes	0.28	0.44	0.15	0.22	0.28	0.54	0.17	0.30	0.18	0.32	0.21	0.36
(IV)	3	**32**	Yes	Yes	0.31	0.48	0.15	0.22	0.27	0.49	0.20	0.34	0.17	0.30	0.22	0.37
(V)	3	16	**No**	Yes	0.70	1.07	0.14	0.21	0.27	0.51	0.20	0.35	0.18	0.31	0.30	0.49
(VI)	3	16	Yes	**No**	0.30	0.45	0.16	0.22	0.33	0.63	0.22	0.38	0.18	0.31	0.23	0.40

Table 2: Ablation results for the trajectory prediction task.

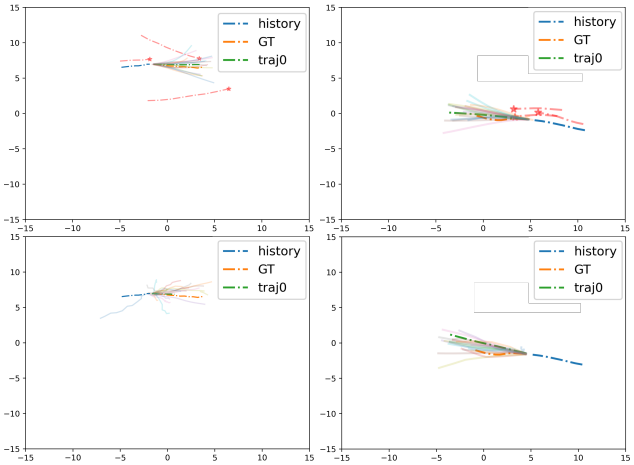

Fig. 2: Qualitative results comparing the trajectories predicted by MapFlow (upper row) concerning FloMo (lower row). The star indicates the ending point of each trajectory of neighbor pedestrians embedded in the Social Context Vector. The SCV allows the model to predict trajectories that are closer to the ground truth overall, avoiding outliers.

transformer attention mechanism to produce more refined local interactions. On average we obtain results in line or better than the state-of-the-art. The Trajectron++ is highly performing because, contrarily to the other benchmarks presented in Tab. 1, it embeds into the graph model a semantic map of the scene. Such an additional feature implies the availability of prior information about the observed environment. For this reason, in Tab. 1 we have highlighted in bold the overall best/on-par results and in bold/italic the best results without exploiting environmental priors. The qualitative results of the multi-modal prediction are shown in Fig. 2.

Ablation study. We perform the ablations of our MapFlow architecture, as shown in Tab. 2. Experiments (I) and (II) reveal that the ideal number of pedestrians N is 3. This suggests that the influence of other agents is generally confined to a reasonably close neighborhood; viceversa, a more extended range of observation tends to bring in noise. Experiments (III) and (IV) explored the impact of varying the encoding vector length, with the best results achieved at $L = 16$. In (V), we show the key effect of the data augmentation for our application, which effectively doubles the size of the dataset. Experiment (VI) demonstrates that removing spatiotemporal embedding led to worse results, emphasizing its importance in modeling spatial and temporal relationships within the Social Context Vector.

4. CONCLUSIONS

We have presented MapFLow, a novel architecture based on Normalizing Flows for trajectory prediction and evaluation. We have introduced a novel module, called Social Context Vector, which can effectively model the social interactions between pedestrians. The validation of the proposed solution, conducted following the standard evaluation pipelines in this research domain, and complemented with an extensive ablation study, has revealed that the introduced SCV improves the quality of the generated trajectories, allowing us to obtain results better or on par with the state-of-the-art results.

5. 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4

[2] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds., Cham, 2020, pp. 683–700, Springer International Publishing. 1, 3, 4

[3] Christoph Schöller and Alois Knoll, "Flomo: Tractable motion prediction with normalizing flows," 2021. 1, 3, 4

[4] Royston Rodrigues, Neha Bhargava, Rajbabu Velmurugan, and Subhasis Chaudhuri, "Multi-timescale trajectory prediction for abnormal human activity detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020. 1

[5] Atanas Poibrenski, Matthias Klusch, Igor Vozniak, and Christian Müller, "M2p3: Multimodal multi-pedestrian path prediction by self-driving cars with egocentric vision," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2020, SAC '20, p. 190–197, ACM. 1

[6] Christoph Rösmann, Malte Oeljeklaus, Frank Hoffmann, and Torsten Bertram, "Online trajectory prediction and planning for social robot navigation," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2017, pp. 1255–1260. 1

[7] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," 2018. 1, 4

[8] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, S. Hamid Rezatofighi, and Silvio Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," 2018. 1

[9] Sebastian Gomez-Gonzalez, Sergey Prokudin, Bernhard Schölkopf, and Jan Peters, "Real time trajectory prediction using deep conditional generative models," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 970–976, 2020. 1

[10] Marion Neumeier, Michael Botsch, Andreas Tollkühn, and Thomas Berberich, "Variational autoencoder-based vehicle trajectory prediction with an interpretable latent space," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 820–827. 1

[11] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al., "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9644–9653. 1

[12] Boris Ivanovic, Karen Leung, Edward Schmerling, and Marco Pavone, "Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295–302, 2020. 1

[13] Anna Mészáros, Javier Alonso-Mora, and Jens Kober, "Trajflow: Learning the distribution over trajectories," 2023. 1

[14] Samuel G Fadel, Sebastian Mair, Ricardo da Silva Torres, and Ulf Brefeld, "Contextual movement models based on normalizing flows," *AStA Advances in Statistical Analysis*, vol. 107, no. 1-2, 2023. 1, 3

[15] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 261–268. 1, 3

[16] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski, "Crowds by example," in *Computer graphics forum*. Wiley Online Library, 2007, vol. 26, pp. 655–664. 1, 3

[17] Andrea Coccaro, Marco Letizia, Humberto Reyes-Gonzalez, and Riccardo Torre, "On the curse of dimensionality for normalizing flows," *arXiv preprint arXiv:2302.12024*, 2023. 1

[18] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios, "Neural spline flows," *Advances in neural information processing systems*, vol. 32, 2019. 3

[19] 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*, 2020, pp. 14424–14432. 4

[20] Chengxin Wang, Shaofeng Cai, and Gary Tan, "Graphtcn: Spatio-temporal interaction modeling for human trajectory prediction," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3450–3459. 4