Transformer Networks for Trajectory Forecasting

Francesco Giuliari[1], Irtiza Hasan[2], Marco Cristani[1], and Fabio Galasso[3]

[1] University of Verona, Italy
{francesco.giuliari,marco.cristani}@univr.it
[2] Inception Institute of Artificial Intelligence (IIAI), UAE
irtiza.hasan@inceptioniai.org
[3] Sapienza University of Rome, Italy
galasso@di.uniroma1.it

Most recent successes on forecasting the people motion are based on LSTM models and *all* most recent progress has been achieved by modelling the social interaction among people and the people interaction with the scene. We question the use of the LSTM models and propose the novel use of Transformer Networks for trajectory forecasting. This is a fundamental switch from the sequential step-by-step processing of LSTMs to the only-attention-based memory mechanisms of Transformers. In particular, we consider both the original Transformer Network (TF) and the larger Bidirectional Transformer (BERT), state-of-the-art on all natural language processing tasks. Our proposed Transformers predict the trajectories of the individual people in the scene. These are "simple" model because each person is modelled separately without any complex human-human nor scene interaction terms. In particular, the TF model *without bells and whistles* yields the best score on the largest and most challenging trajectory forecasting benchmark of TrajNet [41]. Additionally, its extension which predicts multiple plausible future trajectories performs on par with more engineered techniques on the 5 datasets of ETH [33]+UCY [28]. Finally, we show that Transformers may deal with missing observations, as it may be the case with real sensor data.

1 Introduction

Pedestrian forecasting, the goal of predicting future people motion given their past trajectories, has been steadily growing in attention by the research community. Further to being a crucial compound of trackers, especially for the cases of large motion and/or missing observations, the topic serves early action recognition, surveillance and automotive systems.

Starting from [3], Long Short-Term Memory (LSTM) networks have been the workhorse for forecasting and progress has been achieved by devising social pooling mechanisms to model the people social interaction [3,17]. The LSTM is based on sequentially processing sequences and storing hidden states to represent knowledge about the people, e.g. its speed, direction and motion pattern. Most modern approaches have challenged each other on the social interaction of pedestrians, each modelled with a separate LSTM and exchanging information by means of social pooling mechanisms [3,17]. In fact best performing approaches

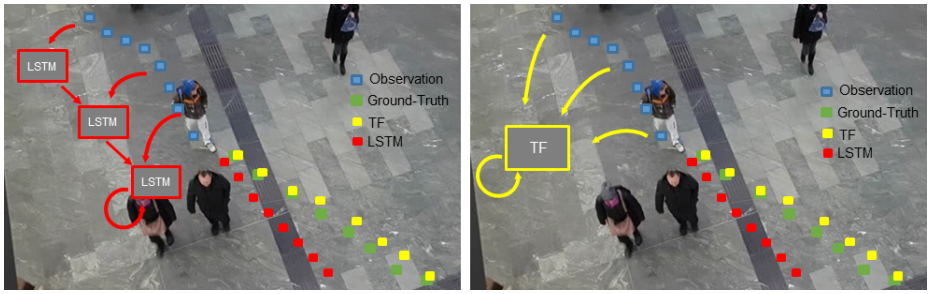

Fig. 1. People trajectory forecasting stands for predicting the future motion of people (*green ground-truth dots*), given an observation interval (*blue dots*). LSTM (left) sequentially processes the observations before starting to predict, while TF analyses in one shot all available observations.

additionally include the semantics of the scene into the LSTMs [27,43,25,42]. However LSTMs have also been target of criticism: their memory mechanism has been criticised [6,30] and, most recently, also their capability of modelling social interaction [45,8,7]. An in-depth understanding of such mechanisms has not been supported by the adopted datasets, such as the 5 datasets of ETH [33] and UCY [28], where performance measures are close to saturation, since leading techniques only report average forecasting errors of ∼20cm across 200m-long pavements.

In this work we side-step social and map mechanisms, and propose to model the trajectories of individual people by Transformer Networks [48], for the first time. Transformer networks have been proposed for Natural Language Processing to model word sequences. These use attention instead of sequential processing. In other words, these estimate which part of the input sentence to focus on, when needing to translate, answer a question or complete the sentence [12,10]. Here we consider for trajectory forecasting the original Transformer Network (TF) and the Bidirectional Transformer (BERT) models, on which state-of-the-art NLP algorithms are based. Fig. 1 illustrates the fundamental difference between TF and LSTM: LSTM sequentially processes the observations before starting to predict auto-regressively, while TF "looks" at all available observations, weighting them according to an attention mechanism.

We assess the performance of TF and BERT on the TrajNet benchmark [41], in order to have a clean evaluation (TrajNet uses a unified evaluation system with a dedicated server) against 42 forecasting approaches, on a large selection of datasets. Our TF outperforms all other techniques, also those including social mechanisms. TF compares favorably also on the ETH+UCY datasets, in particular beating all of the approaches that consider the individual trajectories only. Finally we conduct an ablation to highlight the potential of the Transformers, quantitatively and qualitatively. Of particular interest is the ability of TF to still predict from inputs with missing observation data, thanks to its attention mechanism, which the LSTM cannot do.

2 Related work

Forecasting people trajectories has been studied for over two decades and relevant literature has been surveyed by the work of [7,32]. For the purpose of this paper, we distinguish two main trends of related work: a first which has focused on progressing sequence modelling and a second which has modelled the interactions between the people and between the people and the scene.

Sequence modelling: Trajectory forecasting has experienced a steady progress from hand-crafted energy-based optimization approaches to data-driven ones. Early work on human path prediction have adopted linear [31] or Gaussian regression models [35,50], time-series analysis [34] and auto-regressive models [2], optimizing for hand-crafted energy functions. By contrast later models have been most successful by adoption of LSTM [24] and RNN models, trained with copious amounts of data. Here we argue that Transformer Networks are most suitable to sequence modelling and to forecast trajectories, thanks to their better capability to learn non-linear patterns, especially emerging when large amounts of data is available.

Social models and context: Enabled by the flexibility of the LSTM machinery, best performance has been recently achieved by modelling the social interaction [3,17,49] among people and the scene context [42,44,27], aided by tracking dynamics [40] and the spatio-temporal relations among neighboring people [20,46]. Much literature has recently criticised the capability of LSTM to model the human-human interaction [45,8,7], maintaining that this limits the model generalization capability [45]. Our work side-steps social and environmental interactions and focuses on the prediction of the motion of each person individually. Somehow surprisingly, our "simple" approach achieves best performance on the most challenging benchmark of TrajNet.

In this work, we leverage findings and state-of-the-art techniques developed within the NLP field to model word sequences. In particular, we consider here for trajectory forecasting the original Transformer Networks [48], first to model sequences merely by attention mechanisms. Aside TF, we consider the Bidirectional Transformers BERT [12], which forms the basis for the current performer on most NLP tasks [29]. To the best of our knowledge, this is the first work adopting NLP technique for trajectory forecasting. We experiment on TF and BERT in Sec. 4 and describe the models in more details in the next section.

3 The Transformer Model

We propose a multi-agent framework where each person is modelled by an instance of our transformer network. Each Transformer Network predicts the future motion of the person as a result of its previous motion.

We describe in this section the model input and output (Sec. 3.1), the encoder-decoder Transformer Network (TF) (Sec. 3.2) and the just-encoder BERT model (Sec. 3.3) and the implementation details (Sec. 3.4).

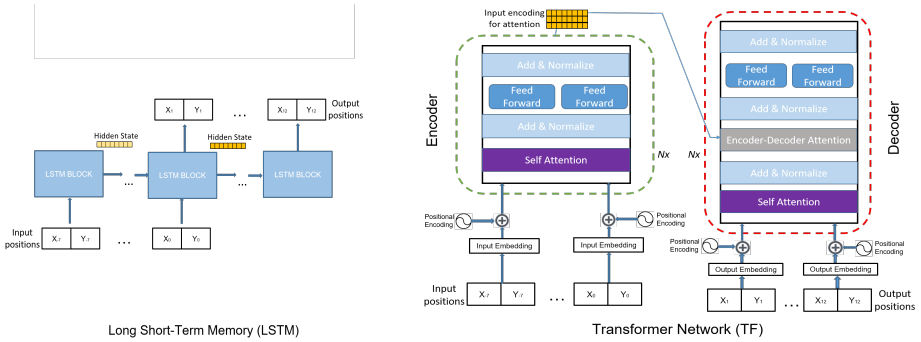

Fig. 2. Model illustration of LSTM (*left*) and TF (*left*). At each time step, LSTM leverages the current-frame information and its hidden state. By contrast, TF leverages the encoder representation of the observed input positions and the previously predicted outputs. In purple and grey are the self-attention and encoder-decoder attention modules, that allow TF to learn on which past position it needs to focus to predict a correct trajectory.

3.1 Model input and output

For each person, the transformer network outputs the predicted future positions by processing their current and prior positions (observations or motion history). We detail here each of the input and output information and parallel those with the established LSTM, with reference to Fig. 2.

Observed and predicted trajectories In formal terms, for person i, we are provided a set $\mathcal{T}_{obs} = \{\mathbf{x}_t^{(i)}\}_{t=-(T_{obs}-1)}^{0}$ of T_{obs} observed current and prior positions in Cartesian coordinates $\mathbf{x} \in \mathcal{R}^2$, and we are required to predict a set $\mathcal{T}_{pred} = \{\mathbf{x}_t^{(i)}\}_{t=1}^{T_{pred}}$ of T_{pred} predicted positions. In order to let the transformer deal with the input, this is embedded onto a higher D-dimensional space by means of a linear projection function ϕ with a matrix of weights $W_{\mathbf{x}}$, followed by a ReLU activation function:

$$\mathbf{e}_{obs}^{(i,t)} = \phi\left(\mathbf{x}_t^{(i)}, W_{\mathbf{x}}\right) \tag{1}$$

In the same way, the output of our transformer model for person i at time t is the D-dimensional vector $\mathbf{e}_{pred}^{(i,t)}$, which is back-projected to the Cartesian person coordinates $\mathbf{x}_t^{(i)}$. LSTM and TF share this aspect.

Positional encoding The transformer does not contain any recurrence – as it the case for LSTM – nor convolution but it encodes time for each past and future time instant t with a "positional encoding". In other words, with the positional encoding, each input embedding $\mathbf{e}_{obs}^{(i,t)}$ is time-stamped with its time t. And a same encoding is used to prompt the model to predict into future instants, as we detail in the next section.

More formally, the input embedding $\mathbf{e}_{obs}^{(i,t)}$ is time-stamped at time t by adding a positional encoding vector $\mathbf{p}^t$, of the same dimensionality D:

$$\xi_{obs}^{(i,t)} = \mathbf{p}^t + \mathbf{e}_{obs}^{(i,t)} \tag{2}$$

We follow the formulation of [48] and use sine/cosine functions to define $\mathbf{p}^t$:

$$\mathbf{p}^t = \{p_{t,d}\}_{d=1}^{D} \tag{3}$$

$$\text{where} \quad p_{t,d} = \begin{cases} \sin\left(\frac{t}{10000^{d/D}}\right) & \text{for } d \text{ even} \\ \cos\left(\frac{t}{10000^{d/D}}\right) & \text{for } d \text{ odd} \end{cases} \tag{4}$$

In other words, each dimension of the positional encoding varies in time according to a sinusoid of different frequency, from 2π to $10000 \cdot 2\pi$, ensuring a unique time stamp for sequences of up to 10000 elements. The time stamps extend to unseen lengths of sequences and allow the model to process input by relative positions, i.e. for a given offset t_o the encoding vector $\mathbf{p}^{t+t_o}$ may be represented as a linear function of $\mathbf{p}^t$.

In this aspect, TF differs greatly from LSTM, cf. Fig. 2. LSTM processes the input sequentially and the order of input positions determine the flow of time. It does not therefore need a positional encoding. However, LSTM needs to "unroll" at training time, i.e. back-propagate the signal sequentially across the LSTM blocks processing the observations. By contrast, TF may learn from all time instants in parallel, which results in more efficient and scalable training.

Notably, thanks to the positional encoding which time-stamps the input, TF may deal with missing observations. Missing data is just neglected, but the model is aware of the relative time-stamps of the presented observations. In Sec. 4, we experiment on this unique feature, important when dealing with real sensor data.

Regression Vs. classification Regression Vs. classification is a recurrent question in trajectory forecasting. Regression techniques, predicting the (x,y) coordinates directly, generally outperform classification-based approaches, where the inputs are quantized into classes and the input data represented as one-hot-vectors. We test both approaches and confirm the better performance of regression. However, a classification approach, which we dub TF_q, provides a probabilistic output across the quantized motions. We leverage therefore TF_q to sample multiple future predictions, which we assess both quantitatively and qualitative in Sec. 4.

TF_q outputs softmax'ed probabilities which differ from the LSTM Gaussian probabilistic output. TF_q outputs are in fact multi-modal, as being generated directly by a DNN, while LSTM only predicts means and variances of Gaussians, forcing the predictions to a single mode. We illustrate this in Sec. 4.

3.2 Encoder-decoder Transformer (TF)

As illustrated in Fig. 2, TF is a modular architecture, where both the encoder and the decoder are composed of 6 layers, each containing three building blocks:

i. an attention module, **ii.** a feed-forward fully-connected module, and **iii.** two residual connections after each of the previous blocks.

The capability of the network to capture sequence non-linearities lies mainly in the attention modules. Within each attention module, an entry of a sequence, named "query", is compared to all other sequence entries, named "keys" by a scaled dot product, scaled by the equal query and key d_k embedding dimensionality. The output is then used to weight the same sequence entries, named now "values". In practice, each sequence entry is considered as query, and all entries are gathered into matrices of queries Q, keys K and values V, to yield attention according to equation:

$$\text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{5}$$

The goal of the encoding stage is to create a representation for the observation sequence, which makes the model *memory*. To this goal, after the encoding of the T_{obs} input embeddings $\xi_s^{(i,t)}$, the network outputs two vectors of keys K_{enc} and values V_{enc} which would be passed on to the decoder stage.

The decoder predicts auto-regressively the future track positions. At each new prediction step, a new decoder query Q_{dec} is compared against the encoder keys K_{enc} and values V_{enc} according to Eq. (5) (encoder-decoder attention) and against the previous decoder prediction (self-attention), to yield the next-step prediction.

Note the important difference w.r.t. LSTM: TF maintains the encoding output (memory) separate from the decoded sequence, while LSTM accumulates both into its hidden state, steering what to memorize or forget at each time. We believe this may contribute to explain how TF outperforms LSTM in long-term horizon predictions, cf. Sec. 4.

3.3 BERT

We consider for trajectory forecasting a second Transformer model, BERT [12]. Differently from TF, BERT is only composed of an encoder and it trains and infers thanks to a masking mechanism. In other words, the model hides (masks) from the self-attention the output positions which it targets for prediction as the TF decoder also does. During training the model learns to predict masked positions. At inference, the model output predictions for the masked outputs.

BERT is the *de-facto* reference model for state-of-the-art NLP methods, but larger than TF ($\sim$2.2 times larger). As we would illustrate in Sec. 4, training BERT on the current largest trajectory forecasting benchmarks does not keep up to the expectations. We draw inspiration from transfer learning and test therefore also how a BERT pre-trained on an NLP task performs on the target task. In particular, we take the lower-cased English text using Whole-Word-Masking; we substitute for the word embedding from dictionary keys with similar linear modules encoding (x,y) positions; and then we similarly convert also the output into (x,y) positions.

3.4 Implementation details

Our TF implementations adopts the parameters of the original Transformer Networks [48], namely $d_{model} = 512$, 6 layers and 8 attention heads. We adopt an L2-loss between the predicted and annotated pedestrian positions and train the network via backpropagation with the Adam optimizer, linear warm-up phase for the first 5 epoch and a decaying learning rate afterward; dropout value of 0.1. The normalization of the network input influences its performance, as also maintained in [51,16]. So we normalize the people speeds by subtracting the mean and dividing by the standard deviation of the train set. For the TF_q, we quantize the people motion by clustering speeds into 1000 joint (x,y) bins, then encode the position by 1000-way one-hot vectors. In order to get a good cluster granularity, we augment the training data by random scaling uniformly with scale $s \in [0.5, 2]$. The lower-cased English text pre-trained BERT from which we fine-tune was trained extensively on the Wikipedia + BookCorpus for $\sim$1M steps

4 Experimental Evaluation

We show the capabilities of the proposed Transformer networks for trajectory forecasting on two recent and large datasets: the TrajNet Challenge [41] dataset and the ETH+UCY dataset [33,28]. Additionally, we perform an ablation study to quantify the model robustness, also in comparison with the widely-adopted LSTM. This includes varying the observation horizon and testing the model on missing data, the latter occurring when some observation samples are missing due to frame-rate drops or excessive uncertainty in the tracking data.

4.1 The Trajnet Challenge

The TrajNet Dataset: At the moment of writing, the TrajNet Challenge[4] [41] does represent the largest multi-scenario forecasting benchmark [39]; the challenge requests to predict 3161 human trajectories, observing for each trajectory 8 consecutive ground-truth values (3.2 seconds) *i.e.*, $t-7, t-6, \ldots, t$, in world plane coordinates (the so-called *world plane Human-Human* protocol) and forecasting the following 12 (4.8 seconds), *i.e.*, $t+1, \ldots, t+12$. The 8-12-value protocol is consistent with the most trajectory forecasting approaches, usually focused on the 5-dataset ETH-univ + ETH-hotel [33] + UCY-zara01 + UCY-zara02 + UCY-univ [28]. Trajnet extends substantially the 5-dataset scenario by diversifying the training data, thus stressing the flexibility and generalization one approach has to exhibit when it comes to unseen scenery/situations. In fact, TrajNet is a superset of diverse datasets that requires to train on four families of trajectories, namely 1) BIWI Hotel [33] (orthogonal bird's eye flight view, moving people), 2) Crowds UCY [28] (3 datasets, tilted bird's eye view, camera mounted on building or utility poles, moving people), 3) MOT PETS [13]

[4] `http://trajnet.stanford.edu/`

(multisensor, different human activities) and 4) Stanford Drone Dataset [37] (8 scenes, high orthogonal bird's eye flight view, different agents as people, cars etc.), for a total of 11448 trajectories. Testing is requested on diverse partitions of BIWI Hotel, Crowds UCY, Stanford Drone Dataset, and is evaluated by a specific server (ground-truth testing data is unavailable for applicants). As a proof of its toughness, it is worth noting that many recent studies restrict on subsets of TrajNet [11,19,36], adopting their train/test splits [22]. We instead consider the whole TrajNet dataset for our experiments. TrajNet allows to consider concurrent trajectories, so that it is compliant with "social" approaches, that can apply. Conversely, it does not allow use raw images, so that approaches which infer on maps as [42,25,44] cannot apply.

Metrics: In agreement with most literature on people trajectory forecasting, the TrajNet performance is measured in terms of: Mean Average Displacement (MAD, equivalently Average Displacement Error ADE [25]), measuring the general fit of the prediction w.r.t. the ground truth, averaging the discrepancy at each time step; Final Average Displacement (FAD, equivalently Final Displacement Error FDE [25]), to check how near to the ground-truth the prediction will be at the last step. The product of MAD and FAD (averaged over the 3161 trajectories) gives a final score which induces a ranking of the approaches.

Results on TrajNet: We report in Table 1 the complete list of 22 *referred* comparative approaches, for a total of 39 approaches at the moment of writing; we omit the 17 unreferred results, which nonetheless had lower performance than the previous top-scoring approach REDv3 [7]. In the table, "Rank" indicates the absolute ranking over all the approaches, including the unreferred ones; "Year" the year of publication of the method; "Context" indicates whether the additional social context (the trajectories of the other co-occurring people) is taken into account ("s") or not ("/").

The scores in *blue italic* refer to the methods proposed in this work (TF, TF_q, BERT, BERT_NLP_pretrained). Surprisingly, the TF model is the new best, with an advantage in terms of both MAD and FAD w.r.t. the second REDv3 [7] and reducing the total error across the 3161 test tracks by ∼145 meters.

It is of interest that the top four approaches (including ours) are individual ones, so no social context is taken into account. These results undoubtedly suggest that in ∼3 seconds of individual observation of an individual, much information about his future can be extracted, and TF is the most successful in doing it. In fact, social approaches appear at lower ranks: the first among them is the SR-LSTM [51], then the highly-cited Social Forces [23] (rank 9 and 27), the MX-LSTM [21] and Social GAN [18]. The quantized TF_q ranks 16th, very probably due to quantization errors. For the trajnet challenge the TF_q was used in its deterministic mode, i.e. the class with highest confidence was selected for the 12 predictions. This is done so because TrajNet is not set up to evaluate best-of-N metric and only a single prediction can be evaluated by the server.

BERT trained from scratch on trajectories ranks 25th; its NLP-pretrained version, fine-tuned on TrajNet, follows immediately. The BERT performance

Table 1. TrajNet Challenge results (world plane Human-Human TrajNet challenge, websites accessed on 3/3/2020). *Blue italic indicates approaches proposed in this work.*

Rank	Method	Avg	FAD	MAD	Context	Cit.	Year
1	*TF*	***0.776***	***1.197***	***0.356***	*/*		*2020*
2	REDv3	0.781	1.201	0.360	/	[7]	2019
4	REDv2	0.783	1.207	0.359	/	[7]	2019
6	RED	0.798	1.229	0.366	/	[7]	2018
7	SR-LSTM	0.816	1.261	0.37	s	[51]	2019
9	S.Forces (EWAP)	0.819	1.266	0.371	s	[23]	1995
12	N-Lin. RNN-Enc-MLP	0.827	1.276	0.377	/	[7]	2018
13	N-Lin. RNN	0.841	1.300	0.381	/	[7]	2018
15	Temp. ConvNet (TCN)	0.841	1.301	0.381	/	[6]	2018
16	*TF_q*	*0.858*	*1.300*	*0.416*	*/*		*2020*
17	N-Linear Seq2Seq	0.860	1.331	0.390	/	[7]	2018
18	MX-LSTM	0.887	1.374	0.399	s	[21]	2018
21	Lin. RNN-Enc.-MLP	0.892	1.381	0.404	/	[7]	2018
22	Lin. Interpolation	0.894	1.359	0.429	/	[7]	2018
24	Lin. MLP (Off)	0.896	1.384	0.407	/	[7]	2018
25	*BERT*	*0.897*	*1.354*	*0.440*	*/*	[12]	*2020*
26	*BERT_NLP_pretrained*	*0.902*	*1.357*	*0.447*	*/*		*2020*
27	S.Forces (ATTR)	0.904	1.395	0.412	s	[23]	1995
29	Lin. Seq2Seq	0.923	1.429	0.418	/	[7]	2018
30	Gated TCN	0.947	1.468	0.426	/	[6]	2018
31	Lin. RNN	0.951	1.482	0.420	/	[7]	2018
32	Lin. MLP (Pos)	1.041	1.592	0.491	/	[7]	2018
34	$LSTM^5$	1.140	1.793	0.491	/		2019
36	S-GAN	1.334	2.107	0.561	s	[18]	2018
40	Gauss. Process	1.642	1.038	2.245	/	[47]	2010
42	N-Linear MLP (Off)	2.103	3.181	1.024	/	[7]	2018

may indicate that the model does require a way larger amount of training data, which at the present moment is absolutely not comparable to the size of an NLP dataset. For this reason, in the rest of the experiments we will concentrate on the TF, more promising at the present moment.

4.2 The ETH+UCY Benchmark

Prior to TrajNet, most literature have benchmarked forecasting performance on a set of 5 datasets, namely the ETH-univ and ETH-hotel [33] video sequences and the UCY-zara01, UCY-zara02 and UCY-univ [28] videos.

Datasets and metrics: The ETH+UCY datasets consist overall of 5 videos taken from 4 different scenes (Zara1 and Zara2 are taken from the same camera but at a different time). Following the evaluation protocol of [3] we sample from the data each 0.4 seconds to get the trajectories. We observe each pedestrian for

3.2 seconds (8 frames) and get ground-truth data for the next 4.8 seconds (12 frames) to evaluate the predictions. The pedestrian positions are converted to world coordinates in meters from the original pixel locations using homography matrices released by the authors. The evaluation is done with a LOO approach training for 4 dataset and testing on the remaining one. Recent works brought up some issues with the ETH+UCY dataset, [45] showed that Hotel contains trajectory that go in a different direction than most of the ones in the other 4 dataset, so learning an environmental prior can be difficult without data augmentation like rotation; [51] bring up the issue that ETH is an accelerated video and so by using a sampling rate of 0.4 seconds the trajectory behave in a different way than the ones in the other 4 datasets, they showed how by reducing the sampling rate they were able to improve their results. We do not take any measure to fix these issues, in order to have a fair comparison against all the other methods that use these dataset using the standard protocol, but during our internal testing we noticed similar improvement when using their sampling rate for ETH. Performance is evaluated using MAD and FAD, in meters.

Results: In Table 2, we compare on the ETH+UCY against the most recent and best performing approaches: S-GAN [17], Social-BIGAT [27] and Trajectron++ [44]. Additionally we include the "individual" version of S-GAN [18], which does not leverage the social information. Note in the Table the trend to include and model as much information as possible. The three leading techniques of S-GAN and Trajectron are in fact "social", and the best performing ones, Social-BIGAT and Trajectron++, additionally ingest the semantic map of the environment ("+map"). Additionally, note that best results are obtained by sampling 20 multiple plausible futures and selecting the best one according to best test performance. We dub this here the best-of-20 protocol, which any technique in Table 2 adopts.

The rightmost column in Table 2 shows our proposed TF_q model, the only which allow to sample distributions of trajectories. TF_q achieves the second best performance, only 0.10 behind in terms of MAD and 0.10 in terms of FAD.

Consistently with the TrajNet challenge, an individual forecasting TF_q technique yields a performance surprisingly ahead or comparable with the best social techniques, even if enclosing additional map information. And trend is also reflected by S-GAN [18], slightly under-performing its individual counterpart.

Note that the best-of-20 protocol is a sort of upper-bound reachable by sampling-based approaches; therefore, we analyze the behavior of our Transformer-based predictors TF in the single-trajectory deterministic regime as in [44], where each method gives a single prediction. Results are reported in Table 3.

The message is clear: when it comes to individual approaches, the transformer predictor is better than any individual LSTM-based approach. Notably, TF is better than the Social-LSTM [18], and it outperforms the Social Attention [27] in terms of FAD too, by a large margin. Notably, the only case in which LSTM compares favorably with TF is on Zara1, which is the less structured of the datasets of the benchmark, mostly containing straight lines.

Table 2. Comparison against SoA models following the best-of-20 protocol. The entirety of SoA approaches is rooted on LSTM, and leverages additional information (social, segmented maps). The mere quantized Transformer TF_q is superior to all the social approaches, second only to Trajectron++ which has access also to maps. Actually, only S-GAN-ind [18] and TF_q have the same input and are directly comparable; all of the other performances are reported as reference, written in cursive.

	LSTM-based				TF-based
	Individual	Social	Soc.+ map		Ind.
	S-GAN-ind [18]	S-GAN [18]	Soc-BIGAT [27]	Trajectron++ [44]	TF_q
ETH	0.81/1.52	*0.87/1.62*	*0.69/1.29*	*0.35/0.77*	**0.61 / 1.12**
Hotel	0.72/1.61	*0.67/1.37*	*0.49/1.01*	*0.18/0.38*	**0.18 / 0.30**
UCY	0.60/1.26	*0.76/1.52*	*0.55/1.32*	*0.22/0.48*	**0.35 / 0.65**
Zara1	0.34/0.69	*0.35/0.68*	*0.30/0.62*	*0.14/0.28*	**0.22 / 0.38**
Zara2	0.42/0.84	*0.42/0.84*	*0.36/0.75*	*0.14/0.30*	**0.17 / 0.32**
Avg	0.58/1.18	*0.61/1.21*	*0.48/1.00*	*0.21/0.45*	**0.31 / 0.55**

4.3 Ablation study and qualitative results

Here we conduct an ablation study on the proposed TF model for forecasting, compare it with the LSTM, and finally illustrate qualitative results.

Changing the Prediction Lengths As a first study case, we compare the stability of the TF and LSTM models when predicting longer temporal horizons. Unfortunately, TrajNet does not allow to set the prediction horizon. We set therefore to pursue a test-time experiment of models trained on the large and complex TrajNet on longer video dataset. We collect these from the 5 datasets of ETH+UCY, by selecting those datasets which are not part of the TrajNet training set, namely ETH and Zara01. In Table 4.3, we vary the observation sequence, from 12 frames (4.8s) to 32 frames (12.8) at a step of 1.8s. Both TF and LSTM have been trained one-dataset-out with training sequences of 8 samples and 12 for the prediction.

On Table 4 are reported the average MAD and FAD values over the ETH-univ and UCY-zara1. Obviously, performances are generally decreasing. TF has a consistent advantage at every horizon Vs. LSTM and the decrease with the horizon of LSTM is approximately 25% worse, as LSTM degrades from 0.78 to 4.13 MAD, while TF degrades from 0.71 to 2.98 MAD.

Missing and noisy data To the best of our knowledge, the problem of having missing coordinates in *coordinate-based* long-term forecasting[6] has been never

[6] *Coordinate-based* forecasting takes as input floor coordinates of people, and is different to *image-based* forecasting, where images are processed to extract bounding boxes locations on the image plane such as [26].

Table 3. Comparison against SoA models following the single trajectory deterministic protocol (numbers of other approaches are taken from [44]). Regular font indicates approaches which are comparable with our Transformer-based predictors, since they use a single individual observed trajectory as input. The other approaches have performance in italic, and are displayed as a reference.

	Linear	LSTM-based					TF-based
	Individual	Individual		Social		Soc.+ map	Individual
	Interpolat.	LSTM [18]	S-GAN-ind [18]	Social LSTM [18]	Soc. Att. [27]	Trajectron++ [44]	Trasformer TF (ours)
ETH	1.33/2.94	1.09/2.94	1.13/2.21	*1.09/2.35*	*0.39/3.74*	*0.50/1.19*	**1.03/2.10**
Hotel	0.39/0.72	0.86/1.91	1.01/2.18	*0.79/1.76*	*0.29/2.64*	*0.24/0.59*	**0.36/0.71**
UCY	0.82/1.59	0.61/1.31	0.60/**1.28**	*0.67/1.40*	*0.20/0.52*	*0.36/0.89*	**0.53**/1.32
Zara1	0.62/1.21	**0.41/0.88**	0.42/0.91	*0.47/1.00*	*0.30/2.13*	*0.29/0.72*	0.44/1.00
Zara2	0.77/1.48	0.52/1.11	0.52/1.11	*0.56/1.17*	*0.33/3.92*	*0.27/0.67*	**0.34/0.76**
avg	0.79/1.59	0.70/1.52	0.74/1.54	*0.72/1.54*	*0.30/2.59*	*0.34/0.84*	**0.54/1.17**

Table 4. MAD and FAD errors when letting the TF and the LSTM models predict longer horizons, i.e. from 12 to 32 time steps. Both models were trained on the TrajNet train set, while errors are reported over the union of ETH and Zara1 sequences (not part of the TrajNet train set).

Pred.	**TF (ours)** MAD / FAD	LSTM [1] MAD / FAD
12	**0.71/1.56**	0.78/1.70
16	**0.95/2.15**	1.15/2.72
20	**1.27/2.90**	1.64/3.99
24	**1.66/3.76**	2.29/5.55
28	**2.27/5.09**	3.07/7.46
32	**2.98/4.52**	4.13/9.96

taken into account. On the contrary, the problem of missing data is common in short term-forecasting (*i.e.* tracking [5]), or forecasting of heterogeneous data [4,9,38,14,15], where in general is treated by designing ad-hoc extensions for filling properly the missing entries (the so called *hindsighting* [4]). Compared to these techniques, our transformer architecture represents a novel view, since *it does not need to fill missing data*; instead, it exploits the remaining samples knowing when they have been observed thanks to the positional encoding. For example, supposing the $t-k$th sample being missed in the observation sequence, the transformer will use the remaining $t-7,\ldots,t-k-1,t-k+1,\ldots,t$, with $1 \leq k \leq 8$ to perform the prediction of $t+1,\ldots,t+12$. This structural ability is absent in LSTM and RNN in general (they cannot work with missing data), and in this sense the Transformer is superior. If replacements of missing values can be

Table 5. Evaluation of missing data results for TF on TrajNet. We experiment dropping a varying number of most recent observed samples, either including or excluding the current frame. For example, in the case of dropping 3 frames, we drop $\mathcal{T}_{obs} = \{\mathbf{x}_t^{(i)}\}_{t=-2)}^{0}$ and $\mathcal{T}_{obs} = \{\mathbf{x}_t^{(i)}\}_{t=-3)}^{-1}$ respectively.

# most recent frames dropped	Drop most recent obs. *including* current frame (FAD/MAD)	Drop most recent obs. *excluding* current frame (FAD/MAD)
0	1.197 / 0.356	1.197 / 0.356
1	1.305/ 0.389	1.267 / 0.373
2	1.409 / 0.429	1.29 / 0.38
3	1.602 / 0.495	1.303 / 0.384
4	1.787 / 0.557	1.313 / 0.387
5	1.897/ 0.593	1.327 / 0.329
6	2.128 / 0.669	1.377 / 0.406

computed, we found that simple linear interpolation gives slight improvements to the results.

Having witnessed the superiority of the transformer over LSTM in absolute sense (on TrajNet, and see Tab. 1) and varying the forecasting horizons (Sec. 4.3), we continue this analysis focusing on our proposed model on the same TrajNet dataset. The idea is to systematically drop one element at observation time, at a fixed position, from the most recent (time t, indicated also as the *current frame*, after that it starts the prediction) to the furthest ($t-7$). Results are reported in Tab. 5.

The results show that, in a complex scenario such as TrajNet, dropping input frames impact the prediction performance, matching the intuition: the more dropped frames, the larger the performance decrease. Interestingly, the current frame plays a key importance, as it is the most recent observed input, from which future predictions start. In fact, dropping the current frame together with the most recent 6 nearly doubles the error, i.e. degrades performance by 91%, from 0.356 to 0.669 MAD. By contrast dropping 6 observed frames but keeping the current one only degrades the performance by 16% (from 0.356 to 0.406 MAD), although the TF may now only leverage 2 observations (farther and closest in time).

Qualitative results Qualitative results can further motivate the numerical results presented so far. In Fig. 3, we report two predictions assessed on TrajNet, built by using the official visualizer of the benchmark[7]. In particular, we artificially superpose the predicted trajectories of LSTM and TF to highlight their different behavior. In Fig. 3 a), the subject is going south, with a minimal acceleration (not immediately visible by the figure, but numerically present); LSTM

[7] `http://trajnet.stanford.edu/visualize.php`

takes this gentle acceleration, predicting a uniform acceleration toward south. TF captures better the dynamics, despite at the very end the final direction is not correct.

In Fig. 3 b) a similar behavior caused LSTM to predict a faster straight trajectory, while TF followed in this case the bending of the GT more precisely.

In general, we observed that LSTM generates trajectories way more regular than those predicted by TF, and this is certainly motivated by its unrolling, opposed to the encoder+decoder architecture of TF. This is also the reason why LSTM is so effective on Zara1, consisting essentially in straight trajectories, and so scarce on Hotel (and in general on TrajNet) if compared to TF.

To further motivate this, in Fig. 3 c) and d), we show 100 sampled trajectories by TF_q on Zara1, for two different cases. Fig. 3 c) presents essentially a monomodal distribution, with the samples concentrated around the GT, enriched by few articulated trajectories, that have low probability (they are few), but are still plausible. Fig. 3 d) shows that TF has learnt a multimodal distribution, which has at least three modes, one turning north, the other going diagonal, the third (with larger number of trajectories) going east.

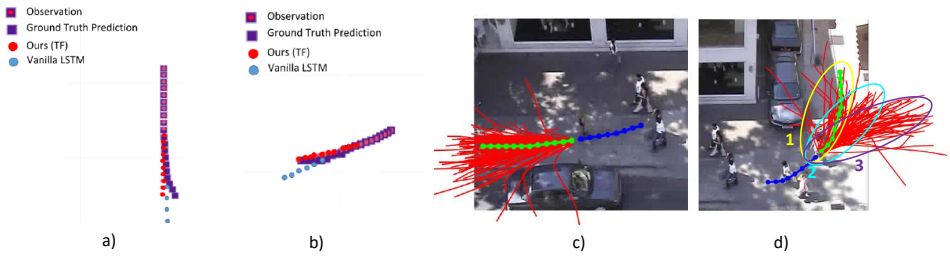

Fig. 3. Qualitative results: a) and b) showcasing failures of LSTM, c) and d) illustrating the trajectory distributions learned by TF_q. Best viewed in colors.

5 Conclusions

We have proposed to Transformers Networks based on attention mechanisms to predict people future trajectories. The Transformers, state-of-the-art on all NLP tasks, also perform best on trajectory forecasting. We believe that this questions the widespread use of LSTMs for modelling people motion and that this questions the current formulation of complex social and environmental interactions, which our model does not need for best performance.

Further to best performance on people forecasting datasets, the proposed Transfomers have shown better long-term prediction behavior, the capability to predict sensible multiple future trajectories and the unique feature of coping with missing input observations, as it may happen when dealing with real sensor data. Equipped with the better temporal models, we envisage potential to address even

larger datasets of long-term sequences, where the importance of social terms may play more crucial roles.

References

1. LSTM MATLAB implementation. `https://it.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html`, accessed: 2019-11-08
2. Akaike, H.: Fitting autoregressive models for prediction. Annals of the institute of Statistical Mathematics **21**(1), 243–247 (1969)
3. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human trajectory prediction in crowded spaces. In: CVPR (2016)
4. Anava, O., Hazan, E., Zeevi, A.: Online time series prediction with missing data. In: International Conference on Machine Learning. pp. 2191–2199 (2015)
5. Bae, S.H., Yoon, K.J.: Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. IEEE transactions on pattern analysis and machine intelligence **40**(3), 595–610 (2017)
6. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
7. Becker, S., Hug, R., Hübner, W., Arens, M.: An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. arXiv preprint arXiv:1805.07663 (2018)
8. Becker, S., Hug, R., Hubner, W., Arens, M.: Red: A simple but effective baseline predictor for the trajnet benchmark. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
9. Chen, H., Grant-Muller, S., Mussone, L., Montgomery, F.: A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. Neural Computing & Applications **10**(3), 277–286 (2001)
10. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint:1904.10509 (2019)
11. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert pre-training of deep bidirectional transformers for language understanding (2019)
13. Ferryman, J., Shahrokni, A.: Pets2009: Dataset and challenge. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. pp. 1–6 (Dec 2009). https://doi.org/10.1109/PETS-WINTER.2009.5399556
14. Ghazi, M.M., Nielsen, M., Pai, A., Cardoso, M.J., Modat, M., Ourselin, S., Sørensen, L.: Robust training of recurrent neural networks to handle missing data for disease progression modeling. arXiv preprint arXiv:1808.05500 (2018)
15. Golyandina, N., Osipov, E.: The "caterpillar"-ssa method for analysis of time series with missing values. Journal of Statistical planning and Inference **137**(8), 2642–2653 (2007)
16. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
17. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018), cONF
18. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018), cONF

19. Haddad, S., Lam, S.K.: Self-growing spatial graph networks for pedestrian trajectory prediction. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 1151–1159 (2020)
20. Hang Su, Jun Zhu, Y.D.B.Z.: Forecast the plausible paths in crowd scenes. In: IJCAI (2017)
21. Hasan, I., Setti, F., Tsesmelis, T., Del Bue, A., Galasso, F., Cristani, M.: Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses. In: CVPR (2018)
22. van der Heiden, T., Nagaraja, N.S., Weiss, C., Gavves, E.: Safecritic: Collision-aware trajectory prediction. arXiv preprint arXiv:1910.06673 (2019)
23. Helbing, D., Molnar, P.: Social force model for. Physical review E **51**(5), 4282 (1995)
24. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
25. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2375–2384 (2019)
26. Kitani, K., Ziebart, B., Bagnell, J., Hebert, M.: Activity forecasting. In: ECCV (2012)
27. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., Savarese, S.: Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: Advances in Neural Information Processing Systems. pp. 137–146 (2019)
28. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum (2007)
29. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta a robustly optimized bert pretraining approach. arXiv:1907.11692 (2019)
30. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3569–3577 (2018)
31. McCullagh, P., Nelder, J.A.: Generalized linear models, no. 37 in monograph on statistics and applied probability (1989)
32. Morris, B.T., Trivedi, M.M.: A survey of vision-based trajectory learning and analysis for surveillance. IEEE Trans. on Circuits and Systems for Video Technology **18**(8), 1114–1127 (2008)
33. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV (2009)
34. Priestley, M.B.: Spectral analysis and time series. Academic press (1981)
35. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate gaussian process regression. Journal of Machine Learning Research **6**(12), 1939–1959 (2005)
36. Ridel, D., Deo, N., Wolf, D., Trivedi, M.: Scene compliant trajectory forecast with agent-centric spatio-temporal grids. IEEE Robotics and Automation Letters **5**(2), 2816–2823 (2020)
37. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European conference on computer vision. pp. 549–565. Springer (2016)
38. Rodrigues, P.C., De Carvalho, M.: Spectral modeling of time series with missing data. Applied Mathematical Modelling **37**(7), 4676–4684 (2013)

39. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: A survey. arXiv preprint arXiv:1905.06113 (2019)
40. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: Learning to track multiple cues with long-term dependencies. arXiv preprint arXiv:1701.01909 (2017)
41. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)
42. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: 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. pp. 1349–1358 (2019)
43. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control (2020)
44. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. arXiv preprint arXiv:2001.03093 (2020)
45. Schöller, C., Aravantinos, V., Lay, F., Knoll, A.: What the constant velocity model can teach us about pedestrian motion prediction. IEEE Robotics and Automation Letters (2020)
46. Su, H., Dong, Y., Zhu, J., Ling, H., Zhang, B.: Crowd scene understanding with coherent recurrent neural networks. In: IJCAI (2016)
47. Trautman, P., Krause, A.: Unfreezing the robot: Navigation in dense, interacting crowds. In: IROS (2010)
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Transformer attention is all you need. In: NIPS (2017)
49. Vemula, A., Muelling, K., Oh, J.: Social attention: Modeling attention in human crowds. In: 2018 IEEE international Conference on Robotics and Automation (ICRA). pp. 1–7. IEEE (2018)
50. Williams, C.K.I.: Prediction with gaussian processes: From linear regression to linear prediction and beyond. In: Learning in graphical models, pp. 599–621. Springer (1998)
51. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12085–12094 (2019)