How many Observations are Enough? Knowledge Distillation for Trajectory Forecasting

Alessio Monti[1] Angelo Porrello[1] Simone Calderara[1] Pasquale Coscia[2]
Lamberto Ballan[2] Rita Cucchiara[1]

[1]University of Modena and Reggio Emilia, Italy [2]University of Padova, Italy

Abstract

Accurate prediction of future human positions is an essential task for modern video-surveillance systems. Current state-of-the-art models usually rely on a "history" of past tracked locations (e.g., 3 to 5 seconds) to predict a plausible sequence of future locations (e.g., up to the next 5 seconds). We feel that this common schema neglects critical traits of realistic applications: as the collection of input trajectories involves machine perception (i.e., detection and tracking), incorrect detection and fragmentation errors may accumulate in crowded scenes, leading to tracking drifts. On this account, the model would be fed with corrupted and noisy input data, thus fatally affecting its prediction performance.

In this regard, we focus on delivering accurate predictions when only few input observations are used, thus potentially lowering the risks associated with automatic perception. To this end, we conceive a novel distillation strategy that allows a knowledge transfer from a teacher network to a student one, the latter fed with fewer observations (just two ones). We show that a properly defined teacher supervision allows a student network to perform comparably to state-of-the-art approaches that demand more observations. Besides, extensive experiments on common trajectory forecasting datasets highlight that our student network better generalizes to unseen scenarios.

1. Introduction

Pedestrian trajectory forecasting deals with predicting future paths through the exploitation of individual trajectory information and mutual influence between pedestrians. This task has several practical applications in advanced surveillance systems [24], behavioral analysis [32], intrusion detection [39], smart vehicles and autonomous systems [4,37].

While several recent works focused on novel deep-network architectures tailored for this task [14, 15, 20, 35, 49, 52], we believe the inference phase of a trajectory predictor has not been thoroughly addressed and investigated yet. Typically, data-driven models are trained and evaluated on large public datasets of tracked trajectories refined with a human intervention to correct missed detections and identity switches. However, this process is unfeasible at inference time: therefore, the input trajectories required to condition the prediction have to be automatically extracted by a tracking system.

In this regard, the widely adopted 8-12 protocol [1,15,20, 22,34,48,52] (*i.e.*, 8 input time steps and 12 ones for prediction), which require data collected at 2.5 FPS, does not provide a large margin of correction for the above situations. In real-time applications, a visual tracking system may provide inaccurate observations for sequences of such length [12]: occlusions, false detections and non-rigid shape deformations pose non-trivial issues.

To overcome the aforementioned limitations, one potential solution is to reduce the length of the input trajectories to the extent we can minimize the tracking associations errors. Based on this intuition, this work proposes an approach based on Knowledge Distillation [18], which recovers a reliable proxy of the same information obtained with more input observations. We show that it allows for an effective inference schema, requiring fewer samples than the training one. We also demonstrate that properly conditioning a model on shorter input trajectories provides more room to generalize across different experimental settings.

From a technical perspective, this work implements our idea by deploying a teacher-student paradigm [18]: a student network is trained to mimic a teacher's behaviour using less input observations. Each network devises a transformer-based architecture that accounts for both *spatial* and *temporal* interactions through an attention mechanism. To deal with a limited number of observations, we propose a distillation procedure that acts on both encoder and decoder stacks of the transformer architecture. Finally, our objective function takes into account ground-truth data and distillation losses to conveniently match teacher and student internal representations.

We remark the following contributions: *i)* to the best of our knowledge, this is the first attempt to in-depth analyze the effectiveness (at inference time) of the evaluation protocol commonly employed by current trajectory prediction models; *ii)* we introduce a novel distillation strategy to reduce the length of the input trajectories while keeping the prediction accurate; *iii)* we explore the student's capabilities in adapting and transferring its knowledge to scenarios exhibiting different levels of complexity in terms of human dynamics and scene interactions. Indeed the experiments highlights that is possible to construct a solid trajectory forecasting system that at inference time is fed only with just 2 observation (i.e. last two) per-pedestrian. This is possible only through the thoughtful exploitation of the global knowledge that can be inferred from training data and distilled into the inference model.

2. Related Work

Social models. Modelling human-human interactions plays a fundamental part in predicting plausible trajectories. Pioneering works take advantage of hand-crafted relations, energy-based features or rule-based models [3, 9, 17, 27, 41, 45], which fail to adapt to scene changes and model complex crowd dynamics. In recent years, data-driven approaches received increasing attention: in their seminal work, Alahi *et al*. [1] capture these interactions by aggregating the hidden states of neighbouring agents with a dedicated grid-based "Social Pooling". Gupta *et al*. [15] improved this mechanisms which is extended to all the agents involved in the scene by max-pooling their hidden states. Such modules have also been extended with attention-based mechanisms [2], while other works propose architectures combining social pooling with context information (*e.g*. scene semantic, groups or head poses) [4, 8, 16, 26, 34].

Recent improvements in graph machine learning [21, 25, 28, 43] motivated the adoption of such flexible structures to model agents relationships. Several solutions [6, 20, 22, 38, 40, 44] consider agents as graph nodes whose features are represented by their hidden states. This solution enables the use of message-passing mechanisms and grants the possibility to aggregate information at each node with powerful Graph Neural Networks (GNNs) like Graph Attention Networks [43]. Zhang *et al*. [52] similarly treat the pedestrian space as a fully-connected graph but design a custom message-passing solution that integrates a *motion gate* to perform a feature selection based on pedestrian movements. Finally, Yu *et al*. [48] only rely on attention mechanisms to predict future locations exploiting recent advances in transformer-based architectures [42].

Knowledge distillation. Knowledge distillation has been firstly investigated as an approach for model compression [10, 18]: a small model (*student*) has to mimic the behaviour of an over-parameterized one (*teacher*). As a result, the student manifest a smaller memory footprint without experiencing a large drop in the overall performance. [31] aims to reduce both student and teacher feature maps; [18] suggests to match the soft-targets before the final classification layer; [50] matches the features of attention regions.

In this work, we employ knowledge distillation in a different fashion. Inspired by [13, 51], our aim is not to compress a model yet to improve its performance. This procedure is usually referred to as *self-distillation*, since the student network shares the same architecture of its teacher. Similarly to [7, 29], our approach sets up asymmetric networks: the student is encouraged to overcome its knowledge gap by following the guide of its teacher, eventually boosting its performance. This is done in the specific context of trajectory forecasting, and we demonstrate that knowledge distillation can lead to effective predictions even when the model has access to very few observations.

3. Model

Trajectory forecasting is usually defined as a time-series prediction problem [32]. The task is particularly challenging because: *i)* human motion is inherently multi-modal, and *ii)* agents simultaneously interact with each other and with static scene elements.

To meet these two points, we design a novel approach modelling both temporal and spatial relationships occurring between agents. Specifically, this section describes how we extend the original Transformer [42] architecture to deal with trajectory forecasting.

3.1. Vanilla Transformer for Trajectory Prediction

To deal with sequence-to-sequence tasks, transformers follow the well established encoder-decoder paradigm. Instead of relying on internal recurrent layers, input sequences are processed as a whole through a purely attentive mechanism. Self-attention aims to discover relationships between every pair of elements in the sequence: this reduces the risk of forgetting past information and allows the network to learn long-range dependencies [42].

From a technical perspective, each embedding e_t (from time step $t = 1$ up to $t = T$) is linearly projected into a triplet of vectors: a query q_t, a key k_t and a value v_t. Then, transformers exploit the dot product between queries and keys to compute attention coefficients (*scaled dot-product attention*), the latter being used to weight the corresponding values and provide the final output. This operation is performed h times (*heads*) on different linear projections of Q, K and V to attend information from several representations at different positions.

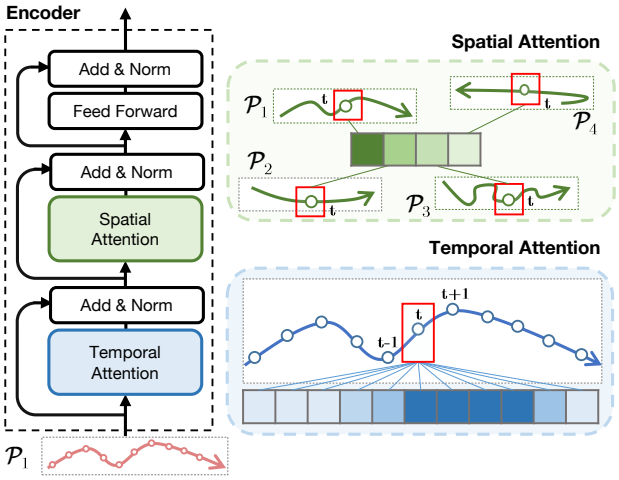

Figure 1. Our *spatio-temporal* attention module. A *temporal* encoder exploits temporal relationships between subsequent time steps of input sequences while a *spatial* encoder collects human-human interactions that occur between agents at a fixed time step.

3.2. Spatio-Temporal Transformer (STT)

As in [42], our proposal firstly attends on the temporal axis with a scaled dot-product attention module. This way, it is able to recover temporal dependencies across different time steps and capture characteristic motion patterns of the monitored agents. However, this vanilla sequence-to-sequence model does not explicitly account for high-level spatio-temporal structure, i.e., no interactions are considered. For this reason, the output of our temporal attention is fed to a second self-attention module that acts on the spatial axis. While in the temporal attention queries, keys and values refer to different time steps of a specific agent, here Q, K and V refer to the embeddings of all the agents at a fixed time step. This way, each agent can additionally attend on the information of its neighbours, recovering useful spatial information. Fig. 1 shows a visual representation of our encoder architecture (the same applies to the decoder).

Relation with previous works. While the approach discussed in [14] consists of a transformer network treating each pedestrian separately (thus handling only temporal information), our self-attention mechanism takes into consideration also spatial interactions between pedestrians. This is somewhat similar to what has been devised by the authors of [48], who equipped a spatio-temporal transformer with an auxiliary memory retaining representations of previous predictions. However, our approach significantly differs in the design of the decoder: while [48] adopts a fully connected layer, we stay close to the original transformer [42] and mirror the encoder into the decoder.

4. Distilling the Observations (DTO)

Our goal is to set up a model capable of accurately predicting future positions when only a few observations are available: this way, we can address the inference-time shortcomings outlined in Sec. 1. More specifically, we devise a two-fold approach (depicted in Fig. 2):

- *firstly* (Sec. 4.1), we train a teacher network to estimate trajectories given 8-length observation sequences;
- *secondly* (Sec. 4.2), we freeze its parameters and attempt to transfer its predictive capability to a student network. Importantly, the latter is forced to operate with an information gap, *i.e.*, using only a small fraction of available inputs (*e.g.*, two last observations).

4.1. Teacher training

To train our teacher network, we follow the standard protocol and consider 8 observation time steps and 12 prediction time steps. The network is trained by *teacher forcing*, *i.e.*, when predicting the next time step, the decoder is conditioned on past ground-truth samples rather than its own predictions. We mark the beginning of the prediction sequence with a *start* token and mask the information related to the future time steps. Mean Squared Error (MSE) between predictions and ground-truth positions is used as loss function while training our teacher network:

$$\mathcal{L}_{\text{GT}} = \frac{1}{P} \sum_{p=0}^{P-1} \left\| \mathbf{x}_{p,[:]} - \hat{\mathbf{x}}_{p,[:]} \right\|^2, \tag{1}$$

where P is the number of pedestrians and $\mathbf{x}_{p,[:]}$ ($\hat{\mathbf{x}}_{p,[:]}$) represents the sequence of ground-truth (predicted) positions of a pedestrian p at time t.

By contrast, the inference procedure resembles an auto-regressive model. The decoder forecasts the first future position using the last hidden state of the encoder stack and an input sequence initially composed only by the *start* token. At each step, the predicted position $\hat{\mathbf{x}}_t$ is concatenated to the current input sequence: this partial sequence is fed again to the decoder to predict the next position $\hat{\mathbf{x}}_{t+1}$.

4.2. Student training

To preserve teacher's predictive capabilities given only few observations, our training strategy relies on transferring the knowledge lying in the entire input sequence: to achieve this, we act on both encoder and decoder stacks.

Encoder distillation. Firstly, we force the student encoder to mimic the behaviour of its teacher's counterpart. Given the information gap between the two networks, *the higher* the transfer occurring at this level, *the higher* the capability of the encoder to infer the missing information from the (few) spatio-temporal interactions it observes. Technically,

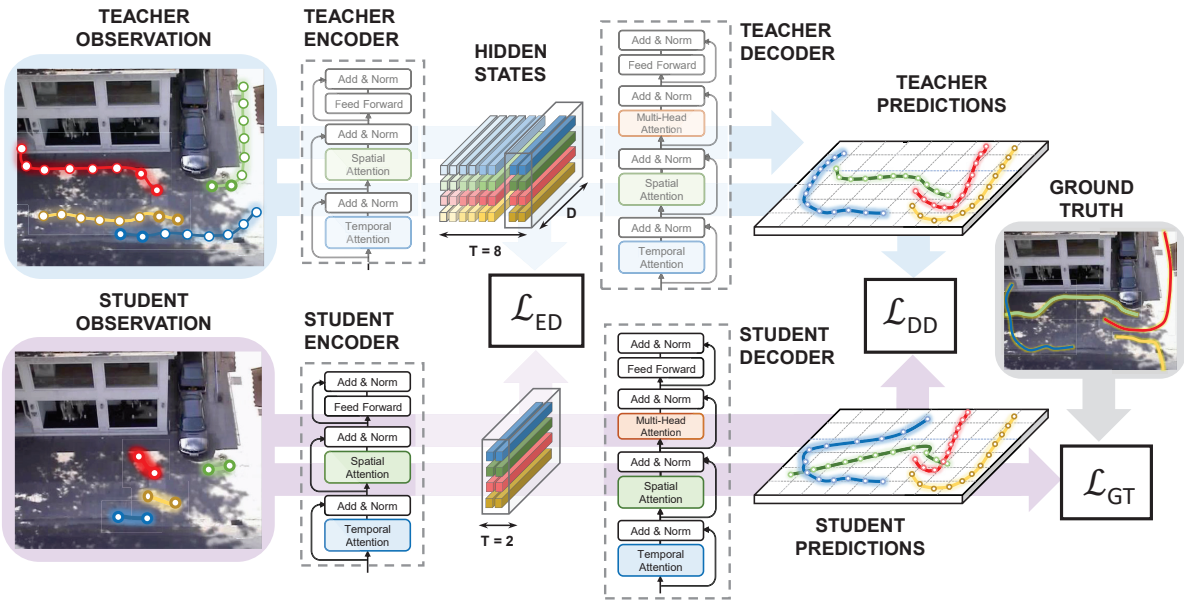

Figure 2. A comprehensive picture of our framework, termed Distilling the Observations (DTO), which provides a training strategy for obtaining accurate trajectory predictions when only few observations are available.

we focus on the final hidden representations produced by the encoder stack (*i.e.*, the outputs of the fully connected layer) and match those corresponding to the common time steps using the following loss:

$$\mathcal{L}_{\text{ED}} = \frac{1}{P}\sum_{p=0}^{P-1}\left\|h^T_{p,[T-K:T]} - h^S_{p,[0:K]}\right\|^2, \qquad (2)$$

where $h^T_{:,:}$ are the activations of the teacher encoder, $h^S_{:,:}$ are the student' ones, while T and K are the number of observations we feed to the teacher and the student, respectively.

Decoder distillation. At the same time, we focus on matching the function space spanned by both teacher and student decoders. We pursue our goal relying on two terms: on one hand, we match the activations prior to the fully connected layer that give the final prediction, *i.e.* $\hat{\mathbf{x}}_{p,t} = \text{FC}(o^T_{p,})$. On the other hand, as proposed in [46], we exploit the self-attention coefficients $\mathbf{A}^T_{p,[:]}$ of the last decoder layer as an additional learning guidance. The corresponding objective function is defined as follows:

$$\mathcal{L}_{\text{DD}} = \frac{1}{P}\sum_{p=0}^{P-1}\left\|o^T_{p,[:]} - o^S_{p,[:]}\right\|^2 + \left\|\mathbf{A}^T_{p,[:]} - \mathbf{A}^S_{p,[:]}\right\|^2. \qquad (3)$$

Overall objective. Finally, the student objective consists of a weighted sum of the prediction loss, which takes into account ground-truth positions, and the distillation losses:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{GT}} + \beta\mathcal{L}_{\text{ED}} + \gamma\mathcal{L}_{\text{DD}}, \qquad (4)$$

where α, β and γ are three hyperparameters balancing the contribution of each term.

5. Experiments

Metrics. We consider two standard error metrics in our comparisons: the *Average Displacement Error* (ADE) and the *Final Displacement Error* (FDE) [27]. While the ADE indicates the average Euclidean distance between all the predicted time steps and the ground-truth ones, the FDE expresses instead only the error regarding the final position.

5.1. Datasets

ETH/UCY. As usually done [1, 15], we stitch together two scenes from ETH [27] (ETH and Hotel) and three scenes from UCY [23] (Univ, Zara-1, Zara-2). The resulting dataset contains more than 1500 pedestrians, taking linear and non-linear paths in outdoor scenarios. We follow the common leave-one-scene-out protocol, training on 4 scenes and testing on the remaining one.

Stanford Drone Dataset (SDD). [30] is a large scale dataset collected by a drone monitoring crowded university campus scenarios. It contains multiple interacting agents (*e.g.*, pedestrians, cyclists, cars) and is composed of a large diversity of urban scenes (*e.g.*, intersections and parks) where people exhibit complex dynamics. We split the SDD World Plane Human-Human dataset [33] into *train* (70 %), *val* (10 %) and *test* (20 %) sets, respectively.

Lyft Prediction Dataset. [19] is one of the largest collection of traffic agent motion data. It includes tracks of cars, pedestrians and other traffic agents recorded by cameras and lidar sensors of Lyft autonomous fleet. We split the reduced version of this dataset (1000 agents) in *train* (70 %), *val* (10 %) and *test* (20 %) sets.

	ETH	Hotel	Univ	Zara-1	Zara-2	AVG
CVM [36]	1.07 / 2.28	0.32 / 0.61	0.52 / 1.17	0.43 / 0.95	0.32 / 0.72	0.53 / 1.15
ST-GAT 1V-1 [20]	0.69 / 1.36	0.44 / 0.90	0.58 / 1.23	0.47 / 1.02	0.40 / 0.86	0.52 / 1.07
Ind-TF [14]	0.60 / 1.25	0.27 / 0.50	0.64 / 1.23	0.57 / 1.09	0.42 / 0.81	0.50 / 0.96
SR-LSTM [52]	0.63 / 1.25	0.37 / 0.73	**0.51 / 1.10**	0.41 / 0.90	0.32 / **0.70**	0.45 / 0.94
STAR [48]	0.56 / 1.11	0.26 / 0.50	0.52 / 1.13	**0.40 / 0.89**	**0.31** / 0.71	**0.41 / 0.87**
STT (8 obs)	**0.54 / 1.10**	**0.24 / 0.46**	0.57 / 1.15	0.45 / 0.94	0.36 / 0.77	0.43 / 0.88
STT (2 obs)	0.72 / 1.45	0.48 / 0.48	0.53 / 1.09	0.64 / 1.21	0.44 / 0.88	0.57 / 1.12
STT + DTO (2 obs)	0.62 / 1.22	0.29 / 0.56	0.58 / 1.14	0.45 / 0.98	0.34 / 0.74	0.46 / 0.93

Table 1. Comparisons (in terms of ADE/FDE) on ETH/UCY. Our teacher network (STT) trained according to the standard protocol shows comparable results w.r.t the competitors, while our student network (STT + DTO) shows similar performance despite its knowledge gap.

5.2. Comparison with the State-of-the-art

Since our proposal is deterministic (*i.e.*, it gives a single future sample), we leave aside stochastic methods [15, 22, 34] and compare our model to the following state-of-the-art deterministic solutions:

- Constant Velocity Model (CVM) [36]: a simple but effective baseline that estimates future positions by considering solely the latest two timesteps;
- ST-GAT [20]: graph-based attention network using LSTMs to model temporal correlations. We consider the 1V-1 version, *i.e.* without variety loss and with one output sample per input;
- Ind-TF [14]: vanilla transformer without explicit interactions modelling;
- SR-LSTM [52]: LSTM-based network integrating a system of motion gates that refines cells' hidden states using neighbourhood information;
- STAR [48]: encoder-decoder architecture based on a transformer network to model temporal information and spatial interactions. We consider its deterministic version obtained removing the Gaussian noise.

Tab. 1 and Tab. 2 report our results: when trained according to the common protocol (8 observations – 12 predictions), our teacher network (STT – 8 obs) performs comparably to state-of-the-art approaches. Notably, the student network (STT + DTO – 2 obs, last row of Tab. 1 and Tab. 2) shows remarkable results: it approaches the teacher on all the datasets, suggesting that the last two observations are an informative summary of the input trajectory. Notably, trivial strategies using only two observations do not achieve the accuracy of our approach: both the CVM and training from scratch with short sequences (STT – 2 obs) deliver higher errors. Instead, our training procedure successfully bridges the huge informative gap simulated at inference time.

It is worth noting that **two observations as input do not necessary generate straight lines as output**: in this case, our approach would have achieved results in line with those of the Constant Velocity Model (CVM) (which predicts straight lines by design). Instead, our results show that this does not happen: even observing two observations solely, indeed, our method takes also spatial relationships among pedestrians into account. This aspect is further corroborated by the superiority of our proposal w.r.t. Ind-TF, which treats every trajectory independently. In this regard, we argue that handling spatial interactions interacts well our distillation technique, closing the gap between using 8-samples input trajectories (Ind-TF) and 2-samples alone (STT – 2 obs): the teacher may drive the student towards novel robust representations (*e.g.* a better understanding of the spatial interactions within the local neighborhood).

Finally, to support our choice of considering only deterministic methods, we rank multiple trajectories based on other criteria which do not require ground-truth annotations at inference time (*e.g.*, for a V-RNN model, the aggregated KL-divergence between the approximate posterior and the

	SDD	Lyft
Ind-TF [14]	0.74 / 1.46	0.31 / 0.62
CVM [36]	0.69 / 1.39	0.29 / 0.61
SR-LSTM [52]	0.72 / 1.47	**0.20 / 0.43**
STT (8 obs)	**0.63 / 1.26**	0.24 / 0.53
STT (2 obs)	0.73 / 1.44	0.31 / 0.56
STT + DTO (2 obs)	0.64 / 1.27	0.27 / 0.55

Table 2. ADE/FDE results on SDD and Lyft.

	Sampling strategy	ADE / FDE
VRNN-1	one sample	0.73 / 1.49
VRNN-20	argmin KL($q \parallel p$)	0.75 / 1.51
VRNN-20	argmin MSE($\cdot$, GT)	**0.58 / 1.17**
STT (ours)	one sample	0.63 / 1.26

Table 3. Comparison between our approach and the V-RNN [11].

Scene	Ground Truth		Tracked trajectories			
	STT (8 obs)	DTO (2 obs)	STT (8 obs)	STT (2 obs)	CVM (2 obs)	DTO (2 obs)
bookstore	**0.48 / 0.97**	0.49 / 0.95	0.58 / 1.08	0.54 / 1.01	0.55 / 1.02	**0.53 / 0.99**
nexus	**0.64 / 1.26**	0.72 / 1.39	1.38 / 2.10	1.33 / 2.07	1.38 / 2.15	**1.29 / 2.03**
deathCircle	**0.76 / 1.55**	0.85 / 1.74	0.99 / 1.83	0.97 / 1.83	0.99 / 1.86	**0.94 / 1.82**
gates	**0.75 / 1.63**	0.82 / 1.72	1.20 / 2.15	1.00 / 1.94	1.10 / 1.97	**0.94 / 1.84**
hyang	**0.37 / 0.80**	0.38 / 0.78	**0.39 / 0.82**	0.48 / 0.95	0.41 / 0.85	0.46 / 0.89
coupa	0.20 / 0.40	**0.20 / 0.38**	0.28 / 0.49	0.21 / 0.41	0.26 / 0.44	**0.20 / 0.38**
overall	**0.55 / 1.12**	0.60 / 1.19	0.84 / 1.46	0.80 / 1.40	0.84 / 1.40	**0.77 / 1.37**

Table 4. On the SDD's scenarios, a comparison (ADE / FDE) between teacher (STT) and student (STT + DTO) on both ground-truth and tracked trajectories.

conditional prior). In this regard, Tab. 3 proposes a comparison between our proposal and a V-RNN model with different sampling strategies: remarkably, only the (unviable) criterion based on ground-truth trajectories yields higher accuracy w.r.t. DTO.

5.3. Towards an "in-the-wild" evaluation: a tracker in the middle

As outlined in Sec. 1, on-line scenarios cannot rely on human intervention to correct detection and re-identification errors at inference time. Based on this motivation, we advocate for employing shorter temporal horizons while estimating future trajectories (2 time steps in place of the common 8 ones), since a tracker can still provide reliable predictions for so short fragments. To shed light on this point, we conduct an experiment on the Stanford Drone Dataset: more precisely, we focus on input trajectories and replace ground-truth associations with Deep SORT's [47] output, which is a tracking-by-detection algorithm that leverages a deep metric for modelling appearance. For each scene, we extract all the detections contained in the observation history; then, we run this tracker on the obtained detections and select as our new observation sequence the most similar tracklet to the ground-truth one. For the sake of simplicity, we restrict our analysis to examples that are successfully followed for at least 8 time steps (hence, we discard cases suffering from identity switches).

In this setting, we evaluate the performance of teacher (namely, STT fed with 8-length tracklets) and student networks (namely, STT trained via DTO fed with 2-length tracklets). As reported in Tab. 4, while DTO appears not advantageous in ideal scenarios (*i.e.* using ground-truth observations), switching to a fully-automatic inference (*i.e.* using tracked trajectories) turns the table: in almost all SDD scenes, our model trained via DTO experiences a diminished degradation in performance w.r.t. the teacher. Its worsening is mainly due to errors accumulation occurring on long sequences: as depicted in Fig. 3a, tracker's errors

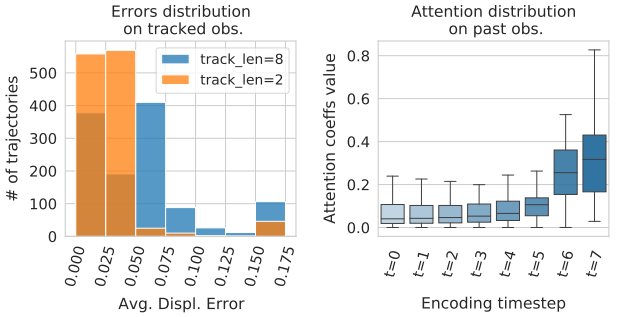

Figure 3. a) histograms of displacements errors between ground-truth trajectories and their estimations provided by Deep SORT; b) spread of the attention coefficients assigned by the decoder to each encoder state.

(measured as ADE between ground-truth and re-tracked input trajectories) spread on higher values for longer tracklets. By contrast, when restraining the model to just a few observations, the average association error tends to be lower, thus affecting less the downstream forecasting model.

Additionally, Tab. 4 draws a comparison between DTO and two baselines that use only 2 observations: STT trained from scratch with 2 observations and the Constant Velocity Model (CVM). As reported, the effectiveness of DTO in a fully-automatic context is not merely due to the use of few time steps, but, more interestingly, also derives from the exploitation of our knowledge distillation paradigm.

5.4. Why Distilling the Observations works

Results reported above suggest that information about future positions can be often recovered by looking solely at the most recent observations. This finding is also investigated by Schöller *et al.* [36], who report that forecasting methods retain only partial input data. Furthermore, Becker *et al.* [5] show that the contribution of the latest time step is 80.3% while for the second-latest is only 8.3%.

Dataset	Training	obs=2	obs=3	obs=4	obs=5	obs=6	obs=7	obs=8
ETH UCY	From scratch	0.56 / 1.12	0.51 / 1.08	0.48 / 1.01	0.47 / 0.90	0.46 / 0.96	0.45 / 0.95	***0.43 / 0.88***
	Variable obs.	0.64 / 1.33	0.63 / 1.31	0.61 / 1.28	0.62 / 1.28	0.62 / 1.29	0.63 / 1.31	0.64 / 1.31
	Past generation	0.50 / 1.06	0.47 / 1.01	0.46 / 0.98	0.46 / 0.96	0.45 / 0.95	0.45 / 0.91	-
	DTO	**0.46 / 0.93**	**0.44 / 0.91**	**0.43 / 0.88**	**0.43 / 0.88**	**0.43 / 0.88**	**0.43 / 0.88**	0.43 / 0.91
Lyft	From scratch	0.31 / 0.56	0.30 / 0.60	0.28 / 0.58	0.27 / 0.57	0.26 / 0.60	0.26 / 0.58	***0.24 / 0.53***
	Variable obs.	0.43 / 0.83	0.41 / 0.76	0.41 / 0.72	0.36 / 0.67	0.36 / 0.67	0.43 / 0.73	0.57 / 0.87
	Past generation	0.36 / 0.67	0.35 / 0.72	0.36 / 0.81	0.36 / 0.73	0.32 / 0.70	0.28 / 0.64	-
	DTO	**0.27 / 0.55**	**0.26 / 0.52**	**0.25 / 0.52**	**0.24 / 0.54**	**0.25 / 0.54**	**0.24 / 0.55**	0.25 / 0.55

Table 5. Comparison (ADE/FDE) between different training strategies; all methods are trained and tested on the same number of time steps, reported in the header. Best results are in bold. The distillation teacher is in underlined italic.

To verify if this behaviour also affects our spatio-temporal transformer, Fig. 3b reports an analysis of the values assumed by the coefficients in the encoder-decoder self-attention, *i.e.*, the coefficients that represent the contribution of each encoder state to the decoding of future positions. Similarly to [36], we observe that, while earlier steps exert an (albeit small) influence, subsequent states provide a higher contribution. In this regard, we conjecture that the robustness of DTO resides in how the model handles the earlier information: at training time, initial time steps are not drastically discarded (as would happen when training from scratch on fewer steps) but, instead, the student learns to dispense with their limited informative content.

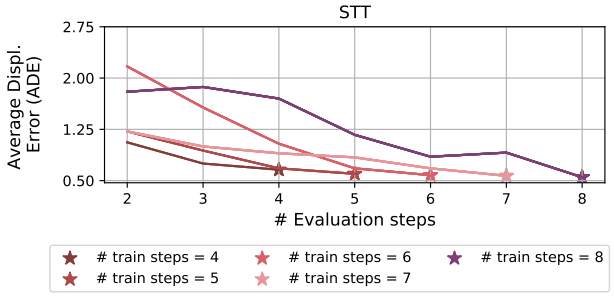

Figure 4. Performance varying the number of observations during evaluation (ETH). The optimum always occurs when training and test conditions match.

5.5. On the "length-shift" problem

We also argue that exploiting longer sequences overly binds the model to the amount of data considered at training time. To prove our intuition, we investigate how models behave when the number of input time steps changes at evaluation time: as shown in Fig. 4, reducing the number of past observations results in a sudden and huge performance drop, even for small variations as removing a single time step. This behaviour – which we refer as "**length-shift problem**" – is a common trait among different splits (8–12, 7–12, etc.) and architectures [1]. This issue could reduce the applicability of these models when limited or partial annotations are available. For this reason, in the following, we explore several strategies that attempt to mitigate this issue: among all, DTO appears the most promising paradigm.

Addressing the length-shift problem. The naïve approach for dealing with this problem is to directly train a forecasting model using fewer time steps (*i.e.*, the same number of observations expected at inference time). However, as reported in Sec. 5.2 and Tab. 5, this choice does not allow the model to extract valuable motion patterns. To this end, we explore a second strategy, training a single instance of our STT with a variable number of observations (from 2 to 8 time steps). As reported in Tab. 5 (*Variable observations*), this strategy brings no benefits: we conjecture that the model learns an average set of motion features, thus granting predictions that are less sensitive to changes in the number of time steps; however, it is far from extracting the set of features that is optimal for each specific input length.

A third approach (*Past generation*) reckons on an auxiliary network to fill the input sequence with a set of generated observations: namely, when the number of observations is less than the one used at training time, we employ a secondary model that predicts the missing part of the input trajectory, which is then concatenated to available positions and fed to the primary forecasting model. This represents a step forward, but still delivers unsatisfactory results: we conjecture that the main limitation of this approach regards the amount of noise injected by the auxiliary module, which then spreads to the model that generates future locations.

Finally, we found particularly beneficial the supervision inherent with our distillation strategy. While the student can collect novel motion patterns from few observations as

[1] Suppl. materials report assessments also on V-RNN and SR-LSTM

(a) *deathCircle_0* (b) *bookstore_0* (c) *deathCircle_0* (d) *deathCircle_1*

Figure 5. A qualitative comparison between predicted trajectories generated by our teacher and its 2–obs distilled student (on Stanford Drone Dataset). In (a) and (b), our student generates more realistic samples, while in (c) and (d), due to more complex dynamics, few observations are not enough to forecast positions close to the ground-truth trajectories.

if it was trained from scratch, it also inherits the broader knowledge lying in teacher's activations. This strategy leads to a remarkable gain in performance, outperforming other solutions and reaching teacher's results (8 – 12).

5.6. On the "domain shift" problem – Knowledge Transfer

Regarding the generalization capabilities of the student, we discuss here its higher degree of robustness to **domain-shifts** (*i.e.*, a change in the underlying data distribution between training and test sets). We expect that exploiting only a few observations limits an excessive specialization on the dataset-specific statistics, thus granting a superior strength to distribution shifts. We validate our claim by investigating the following experimental setting: we use SDD as training set and then test both teacher and student networks on novel scenarios, embodied by the test sets of ETH and Lyft.

As reported in Tab. 6, the presence of DTO favours knowledge transfer and outperforms its distillation-less counterpart. Moreover, in some cases, this strategy even outperforms the results provided by the upper bound, *i.e.*, when there is a match between source and target datasets (*e.g.*, ETH $\rightarrow$ ETH). On the one hand, we conjecture that this is due to the fact that Stanford Drone collects more representative instances of motion dynamics: indeed, the complexity of the learned motion patterns eases the network effort when evaluated on more straightforward scenarios, such as ETH. On the other hand, our framework proves to be beneficial against shifts, thus providing a solution that addresses scenarios with limited available labels.

5.7. Qualitative analysis

In some cases (Fig. 5a and 5b), the gap occurring between our networks does not impact the corresponding predictions. In presence of complex dynamics, *e.g.*, pronounced turns (see Fig. 5c and 5d), the student incurs some issues: here, observing only few positions does not provide enough information to grasp such sophisticated dynamics.

Train. Set	DTO	# of training/evaluation obs						
		2	3	4	5	6	7	8
ETH	✗	0.72	0.68	0.66	0.60	0.58	0.57	0.54
SDD	✗	0.71	0.59	**0.57**	0.58	0.57	0.57	0.55
SDD	✓	**0.66**	**0.57**	0.58	**0.54**	**0.56**	**0.54**	**0.55**

(a) Performance on the test set of ETH.

Train. Set	DTO	# of training/evaluation obs						
		2	3	4	5	6	7	8
Lyft	✗	0.31	0.30	0.28	0.27	0.26	0.26	0.24
SDD	✗	0.70	0.53	0.41	**0.41**	0.44	0.46	0.41
SDD	✓	**0.35**	**0.30**	**0.30**	0.42	**0.36**	**0.28**	**0.34**

(b) Performance on the test set of Lyft.

Table 6. Results of transfer knowledge among different datasets (ADE). We highlight in bold the highest results obtained in case of dataset-shift (*i.e.*, second and third rows of each of the two tables).

6. Conclusion

This paper proposes an in-depth analysis of the evaluation protocol usually employed to assess trajectory prediction models. We conceive a novel training strategy to train a transformer-based architecture to deal with scenarios where only a few observations are available. Our teacher-student paradigm reduces the information gap experienced by the student, thus providing a practical and viable inference scheme for on-line scenarios. We also investigate issues that could emerge at inference time. Our experiments suggest that our strategy also enables a better knowledge transfer capability across different training scenarios.

Acknowledgments. Funded by the PREVUE "PRediction of activities and Events by Vision in an Urban Environment" project (CUP E94I19000650001), PRIN National Research Program, Italian Ministry for Education, University and Research (MIUR).

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 *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 4

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 2

[3] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 2006. 2

[4] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. In *Proc. of the IAPR International Conference on Pattern Recognition (ICPR)*, 2018. 1, 2

[5] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. RED: A simple but effective baseline predictor for the trajnet benchmark. In *Proc. of the European Conference on Computer Vision Workshops*, 2018. 6

[6] Alessia Bertugli, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. AC-VRNN: Attentive conditional-vrnn for multi-future trajectory prediction. *Computer Vision and Image Understanding*, 2021. 2

[7] Shweta Bhardwaj, Mukundhan Srinivasan, and Mitesh M Khapra. Efficient video classification using fewer frames. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[8] Niccoló Bisagno, Bo Zhang, and Nicola Conci. Group LSTM: Group trajectory prediction in crowded scenarios. In *Proc. of the European Conference on Computer Vision Workshops*, 2018. 2

[9] Federico Bolelli, Stefano Allegretti, and Costantino Grana. One dag to rule them all. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[10] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proc. of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006. 2

[11] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 5

[12] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint*, 2020. 1

[13] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proc. of the International Conference on Machine Learning (ICML)*, 2018. 2

[14] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *Proc. of the IAPR International Conference on Pattern Recognition (ICPR)*, 2020. 1, 3, 5

[15] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 5

[16] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[17] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 1995. 2

[18] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Proc. of the NeurIPS Deep Learning and Representation Learning Workshop*, 2015. 1, 2

[19] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska. One thousand and one hours: Self-driving motion prediction dataset. https://level5.lyft.com/dataset/, 2020. 4

[20] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 5

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017. 2

[22] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 5

[23] A. Lerner, Y. Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26, 2007. 4

[24] Yuanman Li, Rongqin Liang, Wei Wei, Wei Wang, Jiantao Zhou, and Xia Li. Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction. *IEEE Transactions on Network Science and Engineering*, 2021. 1

[25] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2016. 2

[26] Matteo Lisotto, Pasquale Coscia, and Lamberto Ballan. Social and scene-aware trajectory prediction in crowded spaces. In *Proc. of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019. 2

[27] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2009. 2, 4

[28] Angelo Porrello, Davide Abati, Simone Calderara, and Rita Cucchiara. Classifying signals on irregular domains via con-

volutional cluster pooling. In *Proc. of the Int'l Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. 2

[29] Angelo Porrello, Luca Bergamini, and Simone Calderara. Robust re-identification by multiple views knowledge distillation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2020. 2

[30] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016. 4

[31] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014. 2

[32] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M. Kitani, Dariu M. Gavrila, and Kai O. Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 2020. 1, 2

[33] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018. 4

[34] 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 *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[35] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020. 1

[36] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *Proc. of the IEEE international Conference on Robotics and Automation (ICRA)*, 2020. 5, 6, 7

[37] Xiao Song, Kai Chen, Xu Li, Jinghan Sun, Baocun Hou, Yong Cui, Baochang Zhang, Gang Xiong, and Zilie Wang. Pedestrian trajectory prediction based on deep convolutional lstm network. *IEEE Transactions on Intelligent Transportation Systems*, 2020. 1

[38] Chen Sun, Per Karlsson, Jiajun Wu, Josh Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019. 2

[39] Jingchen Sun, Jiming Chen, Tao Chen, Jiayuan Fan, and Shibo He. PIDNet: An efficient network for dynamic pedestrian intrusion detection. In *Proc. of the ACM International Conference on Multimedia*, 2020. 1

[40] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[41] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *ACM Transactions on Graphics (TOG)*, 2006. 2

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3

[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018. 2

[44] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *Proc. of the IEEE international Conference on Robotics and Automation (ICRA)*, 2018. 2

[45] Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2006. 2

[46] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[47] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proc. of the IEEE international conference on image processing (ICIP)*, 2017. 6

[48] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 5

[49] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[50] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017. 2

[51] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[52] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5