Explainable Sparse Attention for Memory-Based Trajectory Predictors

Francesco Marchetti, Federico Becattini(✉), Lorenzo Seidenari, and Alberto Del Bimbo

Università degli Studi di Firenze, Florence, Italy
{francesco.marchetti,federico.becattini,lorenzo.seidenari,alberto.bimbo}@unifi.it

Abstract. In this paper we address the problem of trajectory prediction, focusing on memory-based models. Such methods are trained to collect a set of useful samples that can be retrieved and used at test time to condition predictions. We propose Explainable Sparse Attention (ESA), a module that can be seamlessly plugged-in into several existing memory-based state of the art predictors. ESA generates a sparse attention in memory, thus selecting a small subset of memory entries that are relevant for the observed trajectory. This enables an explanation of the model's predictions with reference to previously observed training samples. Furthermore, we demonstrate significant improvements on three trajectory prediction datasets.

Keywords: Trajectory prediction · Memory networks · Explainability · Attention

1 Introduction

Decision-making in autonomous vehicles is often hard to explain and interpret due to the black-box nature of deep learning models, which are commonly deployed on self-driving cars. This makes it hard to assess responsibilities in case of accidents or anomalous behaviors and thus poses an obstacle in ensuring safety.

In this paper we focus on the task of trajectory prediction, which is a core component dedicated to safety in an autonomous driving system. In particular, we study trajectory prediction methods based on Memory Augmented Neural Networks (MANN) [32–34,54], which recently have obtained remarkable results. What makes these models interesting is the capacity to offer a certain degree of explainability about the predictions. Memory-based trajectory predictors leverage a storage of past trajectory representations to obtain cues about previously observed futures [32,33], likely endpoint goals [54] or information about the social context [34]. Methods such as MANTRA [32,33] and MemoNet [54] create a persistent memory with relevant observations corresponding to training samples. How memory is accessed plays a pivotal role in the effectiveness of the model and implies a form of attention to select relevant memory cells. Traditionally, MANNs generate a read key which is compared with all the elements

L. Karlinsky et al. (Eds.): ECCV 2022 Workshops, LNCS 13805, pp. 543–560, 2023.
https://doi.org/10.1007/978-3-031-25072-9_37

stored in memory. In such a way, individual samples can be retrieved at inference time to condition predictions based on known motion patterns. This realizes an effective access-by-content mechanism but does not allow multiple elements to jointly concur to produce an output.

At the same time, each output can be explained by attributing a certain relevance to memory samples. Thus, selecting a small subset of samples from memory provides a way of explaining a decision, since the exact training samples that lead to a certain prediction can be identified. In this paper we propose an improved memory controller that we used to read relevant samples. Our controller is based on sparse attention between the read key and stored samples. Differently from prior work, this allows the model to combine cues from different samples, considerably improving the effectiveness of the predictor. By forcing the attention to be sparse, we can inspect the model decisions with reference to a restricted number of sample, thus making the model explainable. For this reason we refer to our proposed method as ESA (Explainable Sparse Attention).

The main contributions of this paper are the following:

- We present ESA, a novel addressing mechanism to enhance memory-based trajectory predictors using sparse attentions. This enables a global reasoning involving potentially every sample in memory yet focusing only on relevant instances. The advantages are twofold: at training time it reduces redundancies in memory and at test time it yields significant improvements in trajectory prediction benchmarks.
- To address the multimodal nature of the task, we predict multiple futures using a multi-head controller that attends in different ways to the samples in memory.
- To leverage information from all memory entries and at the same time exploit cues from a limited set of training samples, we enhance the memory controller with a sparsemax activation function [35] instead of a softmax. This allows the model to condition predictions on a linear combination of a restricted subset of stored samples.
- Memory Augmented Neural Networks offer explainability by design. We explore how future predictions can be explained with reference to stored samples and how sparsemax further improves the quality of the explanations.

2 Related Works

2.1 Trajectory Prediction

The task of trajectory prediction can be formalized as the problem of calculating the future state of an object, given the observable past states of that same object plus additional observable variables such as the surrounding map and other agents' past states. The main source of knowledge resides in past agents' motion [1,15]. Moreover, an accurate representation of the environment is often sought to provide physical constraints and obtain physically correct predictions [5,6,25,47,49]. Finally, a lot of effort has gone into modelling the interaction between moving agents, addressing so-called social dynamics [1,18,20,21,25,30,44,55].

Trajectory prediction may address pedestrian-only scenarios as well as automotive settings. For the latter a model of the road, such as lane configuration, is critical [2,5,6,10]. Indeed, road layouts physically constrain the motion of vehicles. Especially when dealing with pedestrian motion, a correct model of social interaction allows to improve future trajectory accuracy. When looking at the autonomous driving scenario, for which most of the trajectories to be forecasted are from vehicles, there is no clear evidence that social interaction modelling is beneficial [6,25]. Indeed, a vast number of approaches focused on modelling pedestrian interactions [1,18,20,21,28,38,45]. An effective and natural approach to social interaction modelling is applying Graph Neural Networks (GNN) [22,36,48], modelling agents as nodes.

Recently, a different take to trajectory forecasting has gained traction. Instead of regressing a sequence of future steps, goal based methods estimate intentions, i.e. the spatial goal the agent seeks [9,19,31,54,56]. These approaches proved a high effectiveness attaining state-of-the-art results in many benchmarks.

2.2 Memory and Attention

Traditional Neural Networks have been recently augmented with Memory Modules yielding a new class of methods: Memory Augmented Neural Networks (MANN) [16,53]. By augmenting a NN with a memory we enable the capability to retain a state, similarly to Recurrent Neural Networks (RNN) but with more flexibility. Differently from RNNs a MANN will rely on an external addressable memory instead of exploiting a latent state. This is beneficial in terms of explainability since it is easier to establish a correspondence between memorized features and inputs. Moreover, external memories can retain information during the whole training, making it possible to learn rare samples and deal with long-tail phenomena. The Neural Turing Machine (NTM) is the first known instance of a MANN, demonstrating the capability to retain information and perform reasoning on knowledge stored in memory. For this tasks NTMs are superior to RNNs. NTMS have been recently extended and improved [17,46,50,53]. As we will show in this work MANNs are extremely flexible and can address a large variety of problems: person re-identification [39], online learning [40], visual question answering [24,29] and garment recommendation [8,13].

MANNs have been also proposed to perform trajectory forecasting. MANTRA is a MANN specifically developed to perform multiple trajectory prediction [32,33]. Trajectories are encoded with recurrent units and the external memory is populated during training. At inference time the key-value store of the memory is elegantly exploited to obtain multiple predictions out of a single input past, simply by addressing multiple features in memory. Recently, an approach from the same authors, exploited an external differentiable memory module to learn social rules and perform joint predictions [34]. In this case the memory is emptied at each episode and controllers are trained to store relevant features for each agent, allowing the model to exploit the social interactions of observed agents.

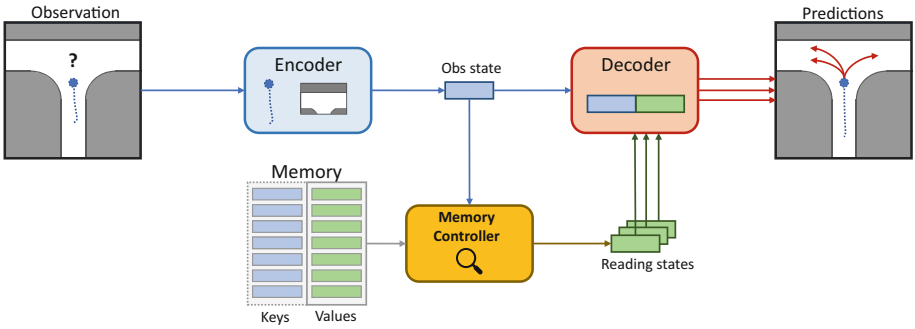

Fig. 1. General architecture of memory-based trajectory predictors.

Transformer based architectures have shown remarkable performance on sequence to sequence problems [12,52]. Attention based approaches are especially interesting since they allow to easily provide explainable outputs. In trajectory forecasting transformers were applied successfully using both their original and bidirectional variant [15]. Obtaining explainable outputs is critical, especially for autonomous driving systems. Recently, using discrete choice models has shown potential to derive interpretable predictions for trajectory forecasting [23].

3 Memory-Based Trajectory Predictors

Memory-based trajectory predictors are a class of trajectory forecasting models based on Memory-Augmented Neural Networks (MANN). The main idea is that, during a training phase, the network learns to build a knowledge base from the training samples. To this end, a controller is trained to store relevant samples in an external memory. The memory can then be accessed at inference time to retrieve relevant information to forecast the future of a given observation. The success of this approach lies in the fact that the model can match a past trajectory with multiple memory entries and obtain cues about possible outcomes of previously observed similar patterns. The actual prediction of the model can be conditioned on this recalled information to leverage additional knowledge rather than the observed sample alone.

Several variants of memory-based trajectory predictors have been proposed in literature [32,33,54]. All these methods share the same structure: (i) first, an encoder learns meaningful representations of the input data; (ii) a writing controller is trained to store relevant and non redundant samples in memory; (iii) a reading controller retrieves relevant information; (iv) finally, a decoder translates the encoded data into a future trajectory. In order to read meaningful cues to inform the decoding process, memory banks are always treated as an associative storage divided into keys and values. Existing methods [32,33,54] mainly differ in the kind of data that memory keys and values represent. Keys, however, must share a common feature space with the data fed as input to

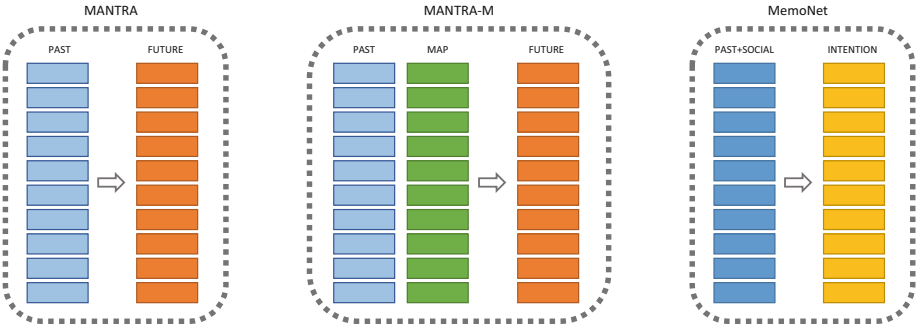

Fig. 2. Memory content for MANTRA [32], MANTRA-M [33] and MemoNet [54]. Memories are divided into keys (left) and values (right).

the model. In fact, at inference time, the current observations is encoded into a latent state which is used to access memory via a similarity function. To comply with the multimodal nature of the trajectory forecasting task, memory-based trajectory predictors can generate multiple futures by retrieving K diverse observations from memory to condition the decoder in different ways. A general scheme of a memory-based trajectory predictor is shown in Fig. 1.

In this work we consider three different models:

- MANTRA [32] was the first to propose a memory-based trajectory predictor. Memory is populated using a writing controller which decides whether or not to store individual samples based on their usefulness for the prediction task. Memory keys are encodings of past trajectories, while memory values are encodings of the respective futures.
- The MANTRA model has been improved in [33] to include contextual information from the surrounding environment. The model now stores as memory keys both encodings of the past and of the semantic segmentation of the surrounding map. We refer to this model as MANTRA-M.
- MemoNet [54] adds a social encoder to produce the latent features, thus allowing joint predictions for multiple agents and relies on a different memory structure. Memory keys are still represented by past trajectories but memory values represent intentions, i.e. final goals where the agent may be directed to. In addition, MemoNet uses a trainable reading controller and a clustering-based decoding process to improve the diversity of the predictions.

Overall, whereas the three models share the same structure, they differ in the information that they store in memory. Most importantly MANTRA and MANTRA-M use the memory bank to inform the decoder with whole future trajectories while MemoNet informs the decoder with intentions. Figure 2 summarizes what kind of information is stored in memory for each model.

All the models access memory through a similarity function: cosine similarity for MANTRA and MANTRA-M and a learnable addresser for MemoNet. It is important to underline that all the models use their similarity function to retrieve

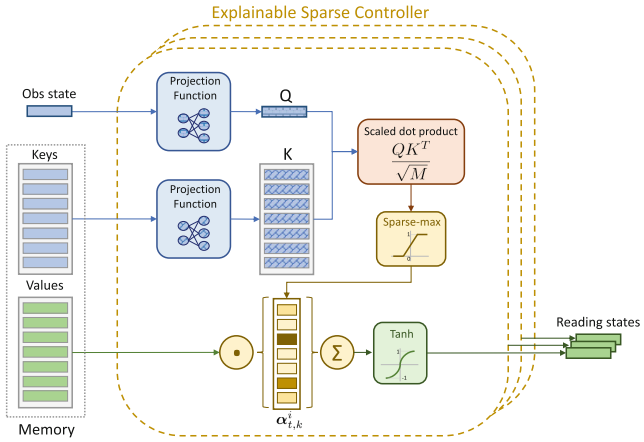

Fig. 3. Explainable Sparse Attention. Each head of the controller attends to memory samples in different ways to generate a prediction.

the top-K samples, which are used to individually generate different futures. In this paper, we propose to remove the top-K mechanism in favor of a multi-head sparse attention that combines information from all memory samples at once.

4 Method

In this paper we propose a novel reading controller for memory-based trajectory predictors named Explainable Sparse Attention (ESA). ESA is divided into multiple read heads. Heads are dedicated to extracting different information from memory, which will then be decoded in parallel into multiple diverse futures. Each head is fed with the encoding of the current observation. A projection layer maps the input into a latent query vector $Q \in \mathcal{R}^M$. Similarly, the same projection is applied to each memory key to obtain a key matrix $K \in \mathcal{R}^{N \times M}$, where N is the number of samples in memory and M the dimension of the projection space. The query Q and the key matrix K are compared using a scaled dot product, followed by a sparsemax activation function [35] to obtain attention weights over memory locations:

$$\alpha_i = \text{sparsemax}\left(\frac{Q_i K_i^T}{\sqrt{M}}\right) \tag{1}$$

The final reading state is a weighted sum of the memory values V with the attention scores $\alpha = \{\alpha_0, ..., \alpha_{N-1}\}$ followed by a tanh activation to regularize the output reading state r_s:

$$r_s = \tanh\left(\sum_i \alpha_i V_i\right) \tag{2}$$

Each head of the ESA controller learns different projections of queries and keys, thus learning different ways to attend to the samples in memory. An overview of the ESA controller is shown in Fig. 3.

Differently from prior work, the ESA controller outputs a reading state that can potentially depend on the whole memory content. This is an advantage since it allows the model to achieve better generalization capabilities by leveraging future information from multiple samples instead of just one. This makes the predictions also more robust to outliers or corrupted samples that might poison the memory bank and condition the output towards wrong predictions.

4.1 Sparsemax vs Softmax

The ESA controller uses a sparsemax activation function instead of the more common softmax. Sparsemax is an activation function that is equivalent to the Euclidean projection of the input vector onto the probability simplex, i.e., the space of points representing a probability distribution between mutually exclusive categories:

$$sparsemax(\mathbf{z}) = \underset{\mathbf{p} \in \Delta^{K-1}}{\operatorname{argmin}} \|\mathbf{p} - \mathbf{z}\|^2, \tag{3}$$

where $\mathbf{z} \in \mathbb{R}^K$ and $\Delta^{K-1} := \{\mathbf{p} \in \mathbb{R}^K | \mathbf{1}^\top \mathbf{p} = 1, \mathbf{p} \geq 0\}$ is a $(K-1)$-dimensional simplex.

It has been shown that, when the input is projected, it is likely to hit the boundary of the simplex, making the output sparse [35]. In addition to producing sparse activations, sparsemax shares most properties of softmax: by definition, the output is a probability distribution; relative ordering between input elements is maintained; the output is invariant to constant addition; differentiability.

For memory-based trajectory predictors that perform a top-K sample retrieval, softmax or sparsemax can be exchanged without affecting the output, since only ordering is important rather than the actual attention values. The usage of sparsemax in the attention mechanism of ESA, however, affects the model by changing the importance of individual samples in the decoding process. There are two important considerations to be made. First, the decoder will receive a reading state with less noise. Being r_s a linear combination of all memory samples, zeroing out the coefficients of most samples, will allow the decoder to focus only on elements that are indeed relevant to the current prediction. Second, attending only to a small subset of elements enables a better model explainability. Attended samples can in fact be used to interpret why the predictor produces its outputs.

5 Experiments

We demonstrate the effectiveness of the ESA controller by experimenting with MANTRA [32], MANTRA-M [33] and MemoNet [54]. We compare the original models against our improved version with the ESA controller. We use the

original experimental setting for each model, testing the models on different trajectory prediction datasets. We first provide an overview of the experimental setting, including datasets and metrics, and we then evaluate the model both quantitatively and qualitatively.

5.1 Evaluation Metrics and Datasets

We report results using two common metrics for vehicle trajectory prediction: *Average Displacement Error* (ADE) and *Final Displacement Error* (FDE). ADE is the average L2 error between all future timesteps and FDE is the error at the final timestep. ADE indicates the overall correctness of a prediction, while FDE quantifies the quality of a prediction at a given future horizon. Following recent literature [25,34,54], we take the best out of K predictions to account for the intrinsic multimodality of the task. To evaluate our models we use the following datasets:

KITTI [14] The dataset consists of hours of navigation in real-world road traffic scenarios. Object bounding boxes, tracks, calibrations, depths and IMU data were acquired through Velodyne laser scanner, GPS localization system and stereo camera Rig. From these data the trajectories of the vehicles were extracted and divided into scenarios of fixed length. For the evaluation phase, we considered the split used in [25,32]. Each example has a total duration of 6 s where the past trajectory is 2 s long and the future trajectory 4 s. The train dataset contains 8613 examples while the test dataset 2907.

Argoverse [6] This dataset is composed by 325k vehicle trajectories acquired in an area of 1000 km^2 in the cities of Pittsburgh and Miami. In addition to the trajectories, HD maps containing lane centerlines, traffic direction, ground height and drivable areas are available. Each example has a duration of 5 s, 2 s for the past and 3 s for the future. The dataset is split into train, validation and test. We report results on the validation set v1.1, for which ground truth data is publicly available.

SDD [42] The Stanford Drone Dataset is composed of pedestrians and bicycles trajectories acquired by a bird's eye view drone at 2.5 Hz on a university campus. The split commonly adopted by other state-of-the-art methods (Trajnet challenge [43]) was used for the experiments. The dataset size is 14k scenarios where each trajectory is expressed in pixels. Each example is divided into past trajectories of 3.2 s and future trajectories of 4.8.

5.2 Results

Table 1, Table 2 and Table 3 show the results obtained on the KITTI [14], Argoverse [6] and SDD [42] datasets respectively. For KITTI, we added the ESA controller on MANTRA [32]. The same procedure was done for the Argoverse dataset, where we have used the MANTRA-M model [33], which leverages both trajectory and map information. We have used MemoNet [54] for demonstrating the capabilities of the ESA controller in the SDD dataset. We refer to the three enhanced models as MANTRA+ESA, MANTRA-M+ESA and MemoNet+ESA.

Table 1. Results on the KITTI dataset. ESA leads to considerable improvements against the standard version of MANTRA [32] varying the number of predictions K.

	Method	ADE				FDE			
		1 s	2 s	3 s	4 s	1 s	2 s	3 s	4 s
K = 1	Kalman [32]	0.51	1.14	1.99	3.03	0.97	2.54	4.71	7.41
	Linear [32]	0.20	0.49	0.96	1.64	0.40	1.18	2.56	4.73
	MLP [32]	**0.20**	**0.49**	0.93	1.53	**0.40**	1.17	2.39	4.12
	DESIRE [25]	-	-	-	-	0.51	1.44	2.76	4.45
	MANTRA [32]	0.24	0.57	1.08	1.78	0.44	1.34	2.79	4.83
	MANTRA+ESA	0.24	0.50	**0.91**	**1.48**	0.41	**1.13**	**2.30**	**4.01**
K = 5	SynthTraj [3]	0.22	0.38	0.59	0.89	0.35	0.73	1.29	2.27
	DESIRE [25]	-	-	-	-	**0.28**	0.67	1.22	**2.06**
	MANTRA [32]	**0.17**	0.36	0.61	0.94	0.30	0.75	1.43	2.48
	MANTRA+ESA	0.21	**0.35**	**0.55**	**0.83**	0.31	**0.66**	**1.20**	2.11
K = 20	DESIRE [25]	-	-	-	-	-	-	-	2.04
	MANTRA [32]	**0.16**	0.27	0.40	0.59	0.25	0.49	0.83	1.49
	MANTRA+ESA	0.17	0.27	**0.38**	**0.56**	**0.24**	**0.47**	**0.76**	**1.43**

Table 2. Results on Argoverse varying the number of predictions K. Errors in meters.

	Method	ADE		FDE		Off-road (%)	Memory size
		1 s	3 s	1 s	3 s		
K = 1	MANTRA-M [33]	0.72	2.36	1.25	5.31	**1.62%**	75,424
	MANTRA-M+ESA	**0.58**	**1.76**	**0.96**	**3.95**	1.84%	**9,701**
K = 6	MANTRA-M [33]	0.56	1.22	0.84	2.30	3.27%	12,467
	MANTRA-M+ESA	**0.47**	**0.93**	**0.68**	**1.57**	**2.32%**	**2,337**
K = 10	MANTRA-M [33]	0.53	1.00	0.77	1.69	4.17%	6,566
	MANTRA-M+ESA	**0.44**	**0.80**	**0.63**	**1.20**	**2.98%**	**1,799**
K = 20	MANTRA-M [33]	0.52	0.84	0.73	1.16	7.93%	2,921
	MANTRA-M+ESA	**0.45**	**0.73**	**0.65**	**0.88**	**3.14%**	**1,085**

In Table 1 we can observe that MANTRA+ESA significantly lowers the prediction error compared to its MANTRA counterpart, especially for a small number of predictions and for long term prediction horizons (4 s). Indeed, for the single prediction case (K = 1), FDE at 4s decreases from 4.83 m to 4.01 m, with an improvement of 0.82 m (17.18%). With a higher number of predictions (K = 5), the FDE error drops from 2.48 m to 2.11 m with a reduction of 0.37 m (14.91%). In almost all metrics, MANTRA+ESA achieves state-of-the-art results. Similarly, we observe significant improvements for MANTRA-M+ESA on the Argoverse dataset, compared to its original formulation [33] (Table 2). We report gains up to 1.36 m in the K = 1 FDE error at 4 s (25.61% improvement). An even larger relative error decrement is reported for K = 6 with an improvement of 31.74%. Similar considerations can be drawn for the other evaluation settings.

Moreover, we show that the amount of information that the network saves in memory with the ESA controller is drastically reduced. In Table 2, we can observe that memory size of MANTRA-M+ESA is significantly smaller than that of the original model, with a difference of 81.25% (10,130 elements). The

Table 3. Results on SDD varying the number of predictions K. Errors in pixels.

K = 20					
Method	ADE	FDE	Method	ADE	FDE
Social-STGCNN [36]	20.60	33.10	SimAug [27]	10.27	19.71
Trajectron++ [45]	19.30	32.70	MANTRA [32]	8.96	17.76
SoPhie [44]	16.27	29.38	PCCSNet [51]	8.62	16.16
NMMP [36]	14.67	26.72	PECNet [31]	9.96	15.88
EvolveGraph [26]	13.90	22.90	LB-EBM [37]	8.87	15.61
EvolveGraph [26]	13.90	22.90	Expert-Goals [19]	**7.69**	14.38
CF-VAE [4]	12.60	22.30	SMEMO [34]	8.11	13.06
Goal-GAN [9]	12.20	22.10	MemoNet [54]	8.56	**12.66**
P2TIRL [11]	12.58	22.07	MemoNet+ESA	**8.02**	12.97

K = 5		
Method	ADE	FDE
DESIRE [25]	19.25	34.05
Ridel et al. [41]	14.92	27.97
MANTRA [32]	13.51	27.34
PECNet [31]	12.79	25.98
PCCSNet [51]	12.54	-
TNT [56]	12.23	21.16
SMEMO [34]	**11.64**	**21.12**
MemoNet [54]	13.92	27.18
MemoNet+ESA	**12.21**	**23.03**

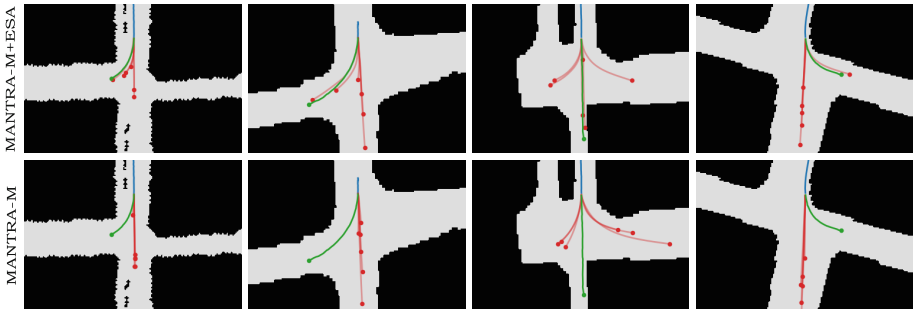

Fig. 4. Comparison between MANTRA-M+ESA and MANTRA-M [33] on Argoverse. Past trajectories are in blue, ground-truth in green and predictions in red. The gray region represents the drivable area of the map. (Color figure online)

ESA controller, thanks to its sparse attention and learned weighted combination of future states read from memory, is able to further reduce redundancy and space occupancy, as well as guaranteeing better performance. In addition, we also have a relevant improvement in the number of generated predictions that do not go off-road. Interestingly, with 20 predictions, we manage to reduce the number trajectories that go astray by half (7.93% vs 3.14%). As we can see from the qualitative examples in Fig. 4 and Fig. 5, the ESA controller allows to generate trajectories with a better multi-modality than the classic MANTRA versions. This yields a lower error and demonstrates the importance of having a model that is able to explore all the plausible directions and speeds, given a certain past movement and context.

In the experiment with the SDD dataset (Table 3), we can observe large gains for K = 5. For K = 20 the results generated by Memonet+ESA are similar to the ones obtained with the original MemoNet formulation, with an improvement for ADE and a slight drop for FDE when predicting 20 different futures. Nonetheless, using the ESA controller, we are able to reduce significantly the memory footprint. Indeed, in Table 4, we can observe that with only 7104 elements in

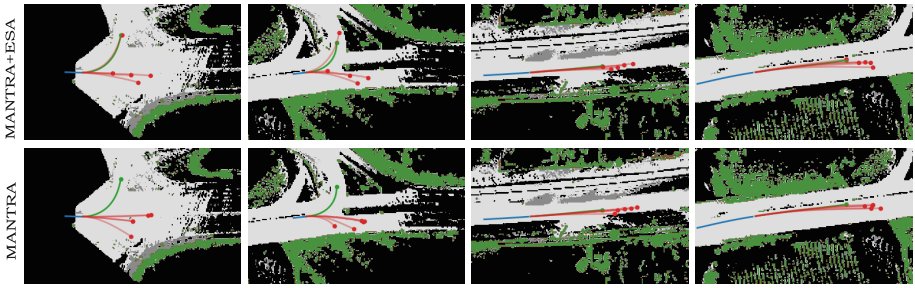

Fig. 5. Comparison between MANTRA+ESA and MANTRA on KITTI. Past trajectory is in blue, ground-truth in green and predictions in red. (Color figure online)

Table 4. Comparison of the results as the ratio of the Memonet thresholds varies with Memonet+ESA.

MemoNet	$\theta_{past}/\theta_{int}$	ADE/FDE	Memory size
w/o ESA	0	8.65/12.84	17970 (100.0%)
	0.5	8.59/12.70	15442 (85.9%)
	1	8.56/12.66	14652 (81.5%)
	5	9.22/14.29	10698 (59.5%)
	10	9.64/15.57	6635 (36.9%)
w/ ESA	-	8.02/12.97	7104 (39.5%)

Fig. 6. MemoNet+ESA and MemoNet comparison on SDD. Blue: past; green: ground-truth; red: predictions. (Color figure online)

memory we can reach similar results to MemoNet, which instead requires 14,652 elements in memory (46% difference).

In MemoNet, memory is initialized by writing all past and intention features available in the training data and then a filtering algorithm erases redundant memory instances. The algorithm removes all those elements with similar starting and ending points. The metric used is the L2-norm and the proximity threshold is determined by two configurable parameters, θ_{past} and θ_{int}, related to starting point and destination distance respectively. In our MemoNet+ESA model we retain the classic memory controller used with key-value memory augmented networks [7,13,32,33], which directly optimizes redundancies with a task loss and does not require memory filtering. Our final memory has a size of 7104 samples. In MemoNet instead, as the ratio of the filtering algorithm thresholds varies, the results change significantly. With a memory size similar to ours ($\theta_{past}/\theta_{int} = 10$, memory size of 6635) FDE is about 17% lower. At the same time, MemoNet requires twice the samples in memory compared to MemoNet+ESA to obtain a comparable error rate. In Fig. 6 we show a qualitative comparison on SDD between MemoNet and MemoNet+ESA.

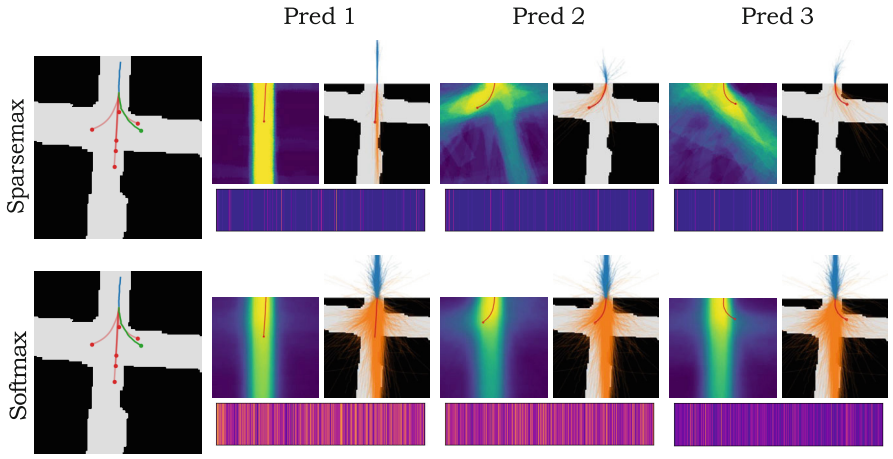

Fig. 7. Explainability analysis on Argoverse. The first column contains the predictions generated with sparsemax and softmax. For each prediction we show the attention vector over memory locations (bottom) and we plot the sum of the semantic maps (left) and the trajectories (right), both weighed by ESA attention.

5.3 Explainability

Providing explainable outputs is a fundamental and crucial aspect for autonomous driving. Since predictors are to be deployed in safety-critical systems, what causes a behavior must be interpretable and observable. One of the advantages of using memory-based trajectory predictors is to explicitly have a link between the memory features used to generate the prediction and the associated training samples. Indeed, by design we can identify the specific samples that allow the generation of a given future trajectory. This in general is not possible with a neural network, where knowledge is distilled in its weights during training. In our work, thanks to the ESA controller, we improve the quality of the explanations. Thanks to the sparsemax activation, each future feature fed to the decoder is a linear combination of a small but significant subset of memory elements. In fact, sparsemax allows the model to generate sparse attentions over the memory. On the contrary, with a softmax activation function, the attention would be smoother and identifying individual sample responsibilities would be harder.

By using the sparsemax activation, we want there to be an evident cause-effect relationship between what is read from memory and the generated output. The quality of the explanation however depends on the architecture of the prediction and in particular on what kind of information is stored in memory. In the following we show how predictions can be interpreted using the MANTRA-M+ESA and MemoNet+ESA models. In Fig. 7 we show an example from the Argoverse dataset, comparing the usage of sparsemax and softmax in the MANTRA-M+ESA model. In the figure, we show attention over memory cells, as well as past, future and semantic maps of the samples retrieved from memory. For each prediction, we represent the attention as a heatmap, normal-

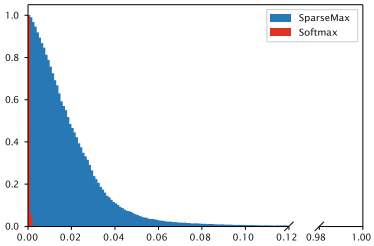

Fig. 8. Normalized histograms of attention values, binned in 0.001-width intervals (x-axis).

Table 5. Quantative analysis of attention values generated by the ESA controller with softmax and sparsemax

Att. Values	Softmax	Sparsemax
>0 (%)	100%	2.37%
>0 (mean)	2328	55
Max	0.21	0.47
Mean	0.0004	0.0180

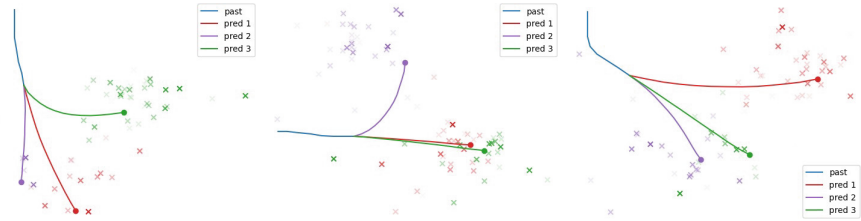

Fig. 9. Explainability for MemoNet+ESA. X-dots are attention-weighted memory intentions associated with predictions of the same color.

ized by its maximum value. We also plot directly on the map all past and future trajectories in memory, weighing their intensity with the respective normalized attention value. In a similar way, we show a semantic heatmap generated by a weighted sum of all maps. Interestingly, the predictions generated by the model using the two activations are very similar. The substantial difference lies in which memory samples are used to generate such predictions. With sparsemax, most attention values are equal to 0 (blue lines in the attention heatmap). Maps and trajectories corresponding to positive attentions can be interpreted as a scenario consistent with the generated prediction. Instead, using softmax, we have a soft attention vector and no element in memory is clearly identifiable as responsible of the prediction. The semantic heatmaps appear similar for all the futures.

We can make a quantitative analysis of this behavior. In Table 5 we report the average number of attention values greater than 0, the maximum attention value and the average attention value. Using sparsemax, only 2.37% of the attention values is positive with an average of 55 elements for each example. On the other hand, with softmax, all memory attention values are positive, the maximum attention value is halved and on average attention values are 45 times lower. This demonstrates that sparsemax allows the model to focus only on relevant memory elements, thus providing interpretable insights about the model's behavior. The same conclusion can be drawn by looking at Fig. 8. We show the normalized histogram of attention values, binned in intervals of width 0.001. Attention values with softmax concentrate close to 0, while with sparsemax are more spreaded.

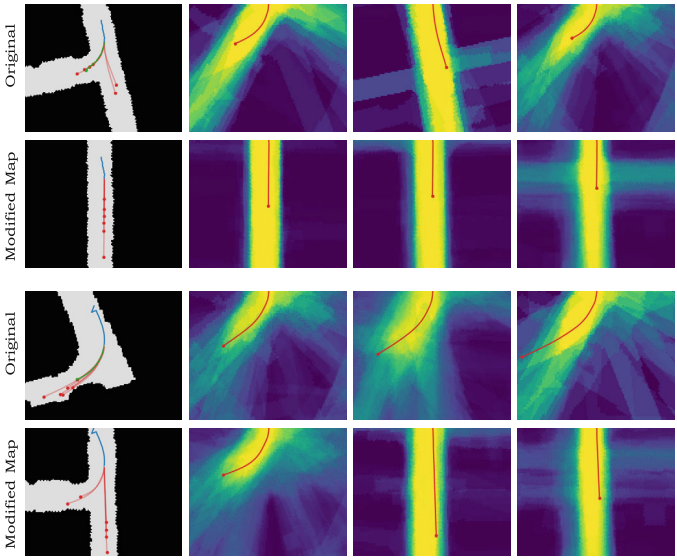

Fig. 10. When changing the observed map, the controller focuses on different memory samples. Past trajectory is in blue, ground-truth in green and the predictions in red. (Color figure online)

A similar analysis can be done for MemoNet+ESA, which stores intentions as endpoint coordinates instead of future trajectories and maps. In Fig. 9, we show for each prediction the intentions retrieved from memory, weighed by attention. The final position of the generated trajectories is always in a neighborhood of the intentions considered to be relevant by the ESA controller.

In addition, to verify the robustness of the model and its explainability, we perform an ablation study on MANTRA-M+ESA. We manually perturb the input and observe how this affects both the predictions and the explainability. In particular, we change the feature of the semantic map, leaving the past unchanged and observe which are the elements in memory that the model focuses on. As we can observe in Fig. 10, different trajectories are generated which are coherent with the new map. The ESA controller also focuses on memory instances that are related to the new semantic map.

6 Conclusions

We proposed ESA a novel reading controller based on explainable sparse attention for Memory-based Trajectory Predictors. Differently from the prior work, ESA allows to generate predictions based on different combinations of the elements in memory, leading to better generalization and robustness. Furthermore, thanks to the sparsemax activation function, it is possible to identify a small subset of samples relevant to generate the output. We tested ESA on top of state

of the art Memory-based Trajectory Predictors obtaining considerable improvements and demonstrated the explainability of the predictions.

Acknowledgements. This work was supported by the European Commission under European Horizon 2020 Programme, grant number 951911 - AI4Media.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971 (2016)
2. Berlincioni, L., Becattini, F., Galteri, L., Seidenari, L., Bimbo, A.D.: Road layout understanding by generative adversarial inpainting. In: Escalera, S., Ayache, S., Wan, J., Madadi, M., Güçlü, U., Baró, X. (eds.) Inpainting and Denoising Challenges. TSSCML, pp. 111–128. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25614-2_10
3. Berlincioni, L., Becattini, F., Seidenari, L., Del Bimbo, A.: Multiple future prediction leveraging synthetic trajectories (2020)
4. Bhattacharyya, A., Hanselmann, M., Fritz, M., Schiele, B., Straehle, C.N.: Conditional flow variational autoencoders for structured sequence prediction (2020)
5. Caesar, H., et al.: nuScenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11621–11631 (2020)
6. Chang, M.F., et al.: Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8748–8757 (2019)
7. De Divitiis, L., Becattini, F., Baecchi, C., Bimbo, A.D.: Disentangling features for fashion recommendation. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) (2022)
8. De Divitiis, L., Becattini, F., Baecchi, C., Del Bimbo, A.: Style-based outfit recommendation. In: 2021 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–4. IEEE (2021)
9. Dendorfer, P., Osep, A., Leal-Taixe, L.: Goal-GAN: multimodal trajectory prediction based on goal position estimation. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2020)
10. Deo, N., Trivedi, M.M.: Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMS. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1179–1184. IEEE (2018)
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. arXiv preprint arXiv:1810.04805 (2018)
13. De Divitiis, L., Becattini, F., Baecchi, C., Del Bimbo, A.: Garment recommendation with memory augmented neural networks. In: Del Bimbo, A., et al. (eds.) ICPR 2021. LNCS, vol. 12662, pp. 282–295. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68790-8_23

14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE (2012)
15. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 10335–10342. IEEE (2021)
16. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)
17. Graves, A., et al.: Hybrid computing using a neural network with dynamic external memory. Nature **538**(7626), 471–476 (2016)
18. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018)
19. He, Z., Wildes, R.P.: Where are you heading? Dynamic trajectory prediction with expert goal examples. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
20. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
21. 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)
22. Kosaraju, V., Sadeghian, A., Martin-Martin, 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, vol. 32. Curran Associates, Inc. (2019)
23. Kothari, P., Kreiss, S., Alahi, A.: Human trajectory forecasting in crowds: a deep learning perspective. arXiv preprint arXiv:2007.03639 (2020)
24. Kumar, A., et al.: Ask me anything: dynamic memory networks for natural language processing. In: International Conference on Machine Learning, pp. 1378–1387 (2016)
25. Lee, N., et al.: Desire: distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 336–345 (2017)
26. Li, J., Yang, F., Tomizuka, M., Choi, C.: Evolvegraph: multi-agent trajectory prediction with dynamic relational reasoning. In: Proceedings of the Neural Information Processing Systems (NeurIPS) (2020)
27. Liang, J., Jiang, L., Hauptmann, A.: *SimAug*: learning robust representations from simulation for trajectory prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12358, pp. 275–292. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58601-0_17
28. Lisotto, M., Coscia, P., Ballan, L.: Social and scene-aware trajectory prediction in crowded spaces. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)
29. Ma, C., et al.: Visual question answering with memory-augmented networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6975–6984 (2018)
30. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D.: Trafficpredict: trajectory prediction for heterogeneous traffic-agents. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6120–6127 (2019)

31. Mangalam, K., et al.: It is not the journey but the destination: endpoint conditioned trajectory prediction. arXiv preprint arXiv:2004.02025 (2020)
32. Marchetti, F., Becattini, F., Seidenari, L., Del Bimbo, A.: Mantra: memory augmented networks for multiple trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
33. Marchetti, F., Becattini, F., Seidenari, L., Del Bimbo, A.: Multiple trajectory prediction of moving agents with memory augmented networks. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
34. Marchetti, F., Becattini, F., Seidenari, L., Del Bimbo, A.: Smemo: social memory for trajectory forecasting. arXiv preprint arXiv:2203.12446 (2022)
35. Martins, A., Astudillo, R.: From softmax to sparsemax: a sparse model of attention and multi-label classification. In: International Conference on Machine Learning, pp. 1614–1623. PMLR (2016)
36. Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C.: Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14424–14432 (2020)
37. Pang, B., Zhao, T., Xie, X., Wu, Y.N.: Trajectory prediction with latent belief energy-based model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11814–11824 (2021)
38. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 261–268. IEEE (2009)
39. Pernici, F., Bruni, M., Del Bimbo, A.: Self-supervised on-line cumulative learning from video streams. Comput. Vis. Image Underst. **197**, 102983 (2020)
40. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: ICARL: incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2001–2010 (2017)
41. Ridel, D., Deo, N., Wolf, D., Trivedi, M.: Scene compliant trajectory forecast with agent-centric spatio-temporal grids. IEEE Robot. Autom. Lett. **5**(2), 2816–2823 (2020). https://doi.org/10.1109/LRA.2020.2974393
42. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: human trajectory understanding in crowded scenes. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 549–565. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_33
43. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: towards a benchmark for human trajectory prediction. arXiv preprint (2018)
44. 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)
45. 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)
46. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning, pp. 1842–1850 (2016)
47. Shafiee, N., Padir, T., Elhamifar, E.: Introvert: human trajectory prediction via conditional 3D attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16815–16825 (2021)

48. Shi, L., et al.: SGCN: sparse graph convolution network for pedestrian trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8994–9003 (2021)
49. Srikanth, S., Ansari, J.A., Sharma, S., et al.: INFER: intermediate representations for future prediction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019) (2019)
50. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in Neural Information Processing Systems, pp. 2440–2448 (2015)
51. Sun, J., Li, Y., Fang, H.S., Lu, C.: Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. arXiv preprint arXiv:2103.07854 (2021)
52. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
53. Weston, J., Chopra, S., Bordes, A.: Memory networks. arXiv preprint arXiv:1410.3916 (2014)
54. Xu, C., Mao, W., Zhang, W., Chen, S.: Remember intentions: retrospective-memory-based trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6488–6497 (2022)
55. Yuan, Y., Weng, X., Ou, Y., Kitani, K.: Agentformer: agent-aware transformers for socio-temporal multi-agent forecasting. arXiv preprint arXiv:2103.14023 (2021)
56. Zhao, H., et al.: TNT: target-driven trajectory prediction. arXiv abs/2008.08294 (2020)