Remember Intentions: Retrospective-Memory-based Trajectory Prediction

Chenxin Xu[1*], Weibo Mao[1*], Wenjun Zhang[1], Siheng Chen[1,2†],
[1]Shanghai Jiao Tong University, [2]Shanghai AI Laboratory
{xcxwakaka,kirino.mao,zhangwenjun,sihengc}@sjtu.edu.cn

Abstract

To realize trajectory prediction, most previous methods adopt the parameter-based approach, which encodes all the seen past-future instance pairs into model parameters. However, in this way, the model parameters come from all seen instances, which means a huge amount of irrelevant seen instances might also involve in predicting the current situation, disturbing the performance. To provide a more explicit link between the current situation and the seen instances, we imitate the mechanism of retrospective memory in neuropsychology and propose MemoNet, an instance-based approach that predicts the movement intentions of agents by looking for similar scenarios in the training data. In MemoNet, we design a pair of memory banks to explicitly store representative instances in the training set, acting as prefrontal cortex in the neural system, and a trainable memory addresser to adaptively search a current situation with similar instances in the memory bank, acting like basal ganglia. During prediction, MemoNet recalls previous memory by using the memory addresser to index related instances in the memory bank. We further propose a two-step trajectory prediction system, where the first step is to leverage MemoNet to predict the destination and the second step is to fulfill the whole trajectory according to the predicted destinations. Experiments show that the proposed MemoNet improves the FDE by 20.3%/10.2%/28.3% from the previous best method on SDD/ETH-UCY/NBA datasets. Experiments also show that our MemoNet has the ability to trace back to specific instances during prediction, promoting more interpretability.

1. Introduction

Trajectory prediction aims to predict the future movements for one or multiple interacting agents given the past trajectories. On the one hand, this task has broad practical applications to autonomous driving [24], drones [6], surveillance systems [42] and interactive robotics [18]; on the other hand, this is a fundamental scientific question about linking the past to the future. The overall strategy is to summarize useful experiences from a large amount of seen past-future pairs and then leverage those experiences to predict possible future intentions for the current situation.

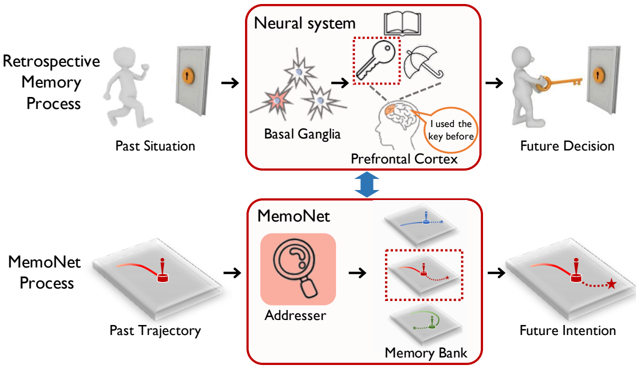


Figure 1. MemoNet mimics retrospective memory process. We use the memory bank to explicitly store representative instances, acting like prefrontal cortex; and the memory addresser to search similar memory instances with current situation, acting like basal ganglia.

To obtain useful experiences, previous works consider a parameter-based approach, which uses training data to optimize model parameters. In this way, all the experiences are implicitly summarized and stored in a model as a whole during the optimization process. For example, [16, 35, 46] use encoder-decoder architectures and [13, 15] consider generator-discriminator architectures to regress future trajectory predictions. [17,23,32,38,47] use conditional variational autoencoders to sample multiple future trajectory embedding from latent distributions. [10,26] rely on a bivariate Gaussian Mixture Model to output position distributions. However, the parameter-based approach is not optimal for two reasons. First, it lacks interpretability because all model parameters do not have clear semantic meaning in the physical world. This is critical in safety-sensitive applications, such as autonomous driving. Second, since the model parameters are trained from all seen instances, a huge amount of irrelevant seen past-future pairs might also involve in predicting the current situation, disturbing the performance.

To promote more interpretability and provide a more explicit link between the current situation and the seen instances, we propose MemoNet whose working mechanism is

[*]Equal contribution. [†]Corresponding author.
Code is available at: https://github.com/MediaBrain-SJTU/MemoNet

inspired by human's retrospective memory in neuropsychology [3, 9], the process that human learns intended future actions by recalling information learned before. The proposed MemoNet achieves the intention prediction by searching for similar instances stored during training. In MemoNet, we use a pair of past and intention memory banks to store the features of past-future instance pairs and a memory addresser to search relevant instances with the new prediction case in the memory bank. The memory bank simulates the prefrontal cortex in the neural system, which records the human reaction when performing a task. The memory addresser simulates the basal ganglia in the neural system, which activates the related memory records in the prefrontal cortex. Fig.1 shows an analogy between the retrospective memory process and our MemoNet process.

The proposed MemoNet includes four key designs. First, we propose a joint-reconstruction-based feature-learning architecture to initialize the pair of past and intention memory banks. The architecture contains two encoders and follows a joint-reconstruction structure to obtain compatible past trajectory and future intention features. Second, we propose a memory filter algorithm to erase the redundant memory instances in the memory banks. The filter algorithm is training-free and invariant to the permutation of training samples, providing high efficiency and robustness for the memory banks. Third, we propose a trainable memory addresser to search similar memory instances. The addresser contains a learnable attention network to compute similarity scores. To train such an addresser, we propose a pseudo-label generation to guide the addresser to correctly search most similar memory instances. Fourth, we propose an intention clustering to produce diverse intention predictions. Through the clustering algorithm, intentions with low-frequency occurrences are captured to promote the prediction diversity and intentions with high-frequency occurrences are merged to improve the prediction robustness.

We build a two-step trajectory prediction system, where the first step is to leverage MemoNet to predict the intentions and the second step is to fulfill the whole trajectory according to the predicted intentions. Note that MemoNet only predicts the destination to represent the intention because the destination carries most of the modality information in a trajectory. This two-step prediction disassembles a complex problem into two relatively simple problems, promoting a more accurate prediction. To evaluate the effectiveness of our method, we conduct experiments on three datasets: Stanford Drones (SDD), ETH-UCY and NBA. The quantitative result shows we outperform the previous state-of-the-art method 20.3%/10.2%/28.3% on FDE representing we achieve an accurate intention prediction with the MemoNet. The qualitative results also reflect that our MemoNet has the ability to trace back to specific memorized samples during the prediction, promoting more interpretability.

The main contributions of this paper are:

• We propose MemoNet, a novel instance-based framework to achieve future intention prediction. The working mechanism of MemoNet is based on a more explicit link between the current situation and seen instances, imitating retrospective memory studied in neuropsychology.

• We propose four novel designs in MemoNet, including 1) reconstruction-based feature-learning architecture, which initializes the memory banks, 2) memory filtering, which reduces the redundancy in memory banks, 3) memory addresser, which searches similar memory instances with the incoming prediction case in memory banks, and 4) intention clustering, which promotes prediction diversity.

• We conduct experiments to evaluate our method on several real-world datasets. Our approach achieves the state-of-the-art on well-established pedestrian trajectory prediction datasets by reducing the FDE 20.3%/10.2%/28.3% on SDD/ETH-UCY/NBA datasets. Our approach also equips with the ability to trace back to specific memorized instances during the prediction, promoting more interpretability.

2. Related Work

Trajectory prediction. Early work on trajectory prediction adopts a deterministic approach using models such as social forces [14, 34], Markov process [21, 44], and RNNs [1, 36, 43]. Recently, researchers begin to propose frameworks to predict multi-model trajectories, which can be mainly categorized into two types: regression, generation. Regression frameworks mainly utilize encode-decode structures [7, 16, 27, 35, 46], or reinforcement learning-based structure [25], or generator-discriminator structures [13, 15] with adding noise [13, 15, 16, 35, 46], using random initialization [28], or using multi-head output [29, 41] to regress multiple future trajectories. Generation frameworks estimate the distribution of future trajectory or its embedding with deep generative models [19]. [10, 26] utilize a Gaussian mixture distribution to model the future trajectory distribution and the model estimates its mean and covariance. The mainly used framework is conditional variational autoencoders [17, 23, 32, 38, 47], which achieve the prediction by estimating the parameters of an intermediate distribution and sampling future trajectory features from such a distribution.

Both the regression and generation frameworks are parameter-based, utilizing training data to optimize model parameters. In such frameworks, learned experience is a hidden representation stored implicitly in model parameters as a whole, lacking the ability to address an individual instance of experience. In this work, we propose a new instance-based framework based on retrospective memory which memorizes various past trajectories and corresponding intentions. In predicting, the framework recalls similar previous memory instances for guiding future prediction. Compared with previous methods, our method provides a more explicit link between the current prediction and seen data, which promotes more interpretability and higher performance.

Memory Networks. The first proposed memory network is called Neural Turing Machines (NTM) [11] which is analogous to a Von Neumann architecture consisting of a neural network controller and a memory bank. The NTM architecture is extended in meta-learning [39] which implements a Least Recently Used memory access strategy to make predictions using few samples. [12] proposes a differentiable neural computer that can read from and write to an external memory matrix. Memory network is also proved its effectiveness on question-answering tasks [45] where the model stores the question-answering pair into a long-term memory as a knowledge base and outputs a textual response. [40] proposes an end-to-end memory network for question-answering with a recurrent attention model in which the recurrence reads from a large external memory. [22, 30] apply memory networks further into visual question-answering tasks [2]. [31] applies a generative memory for continual trajectory prediction.

A close related work with ours is [33], which leverages the memory mechanism to achieve single-agent trajectory prediction. However, the differences include four aspects: i) the previous work only considers single-agent trajectory prediction; while the proposed MemoNet is able to handle multi-agent trajectory prediction with social influence; ii) the memory bank in the previous work stores the entire trajectories; while MemoNet focuses on intention, which is more efficient in memorizing possible movement patterns; iii) the previous work uses fixed cosine similarity to search related memories; while MemoNet uses a trainable addresser to learn a similarity metric, leading to better memory searching; and iv) the previous work is hard to both ensure diversity and preserve precision while MemoNet adopts intention clustering to promote multi-modality prediction with robustness. Overall, the proposed MemoNet outperforms [33] by 28.7%/46.2% in FDE on SDD/ETH-UCY datasets.

3. Problem Formulation

Trajectory prediction is to predict an agent's future trajectory from its past trajectory and neighboring agents' past trajectories. Mathematically, for a to-be-predicted agent, let $\mathbf{x}^t \in \mathbb{R}^2$ be its spatial coordinate at timestamp t and $\mathbf{X} = [\mathbf{x}^{-T_\mathrm{p}+1}, \mathbf{x}^{-T_\mathrm{p}+2}, \cdots, \mathbf{x}^0] \in \mathbb{R}^{T_\mathrm{p}\times 2}$ be its past trajectory over T_p timestamps. Let $\mathcal{N}$ be the neighbouring agent set and $\mathbb{X}_\mathcal{N} = [\mathbf{X}_{\mathcal{N}_1}, \mathbf{X}_{\mathcal{N}_2}, \cdots, \mathbf{X}_{\mathcal{N}_N}] \in \mathbb{R}^{N\times T_p\times 2}$ be the past trajectories of neighbours, where $\mathbf{X}_{\mathcal{N}_\ell} \in \mathbb{R}^{T_\mathrm{p}\times 2}$ is the trajectory of the ℓth neighbour. The future trajectory of the to-be-predicted agent is $\mathbf{Y} = [\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^{T_\mathrm{f}}] \in \mathbb{R}^{T_\mathrm{f}\times 2}$ where $\mathbf{y}^t \in \mathbb{R}^2$ is the spatial coordinate of at future timestamp t. The overall goal is to train a prediction model $g(\cdot)$, so that the predicted future trajectory $\widehat{\mathbf{Y}} = g(\mathbf{X}, \mathbb{X}_\mathcal{N})$ is as close to the ground-truth $\mathbf{Y}$ as possible.

To reach this goal, we consider a two-step strategy, where we first predict the agent's intention and then fulfill the complete trajectory based on the predicted intention. The intuition behind is to disassemble a complex problem into two relatively simple problems, promoting a more accurate prediction. Here we represent the agent's intention by its destination as the destination could reflect most of the movement patterns. Mathematically, we target to learn an intention prediction model $g_\text{int}(\cdot)$ that predicts a intention $\widehat{\mathbf{y}}^{T_f} = g_\text{int}(\mathbf{X}, \mathbb{X}_\mathcal{N})$. We next target to train the trajectory fulfilling model $g_\text{full}(\cdot)$ based on the predicted intention $\widehat{\mathbf{Y}} = g_\text{full}(\mathbf{X}, \mathbb{X}_\mathcal{N}, \widehat{\mathbf{y}}^{T_f})$. In this spirit, we propose MemoNet for intention prediction; see Sec.4; we then build the overall prediction model based on MemoNet; see Sec.5.

4. MemoNet: Intention Prediction

MemoNet exploits retrospective memory from similar scenarios of previous experience to obtain the possible multimodal future movement intentions. The core of MemoNet is to store representative instances in the memory bank and then use a memory addresser to search relevant seen instances with the current situation in the memory bank. Sec. 4.1 proposes the memory bank and Sec. 4.2 proposes the memory addresser. To enable diverse intention prediction, we propose intention clustering in Sec. 4.3. Finally, we summarize the inference process of MemoNet in Sec. 4.4.

4.1. Memory bank

Memory bank initialization. We consider a pair of correlated memory banks: a past memory bank and an intention memory bank. The past memory bank stores a set of past trajectory features and the intention memory bank stores a set of corresponding future intention features. They together associate the past with the future. Mathematically, let $\mathcal{M}_\text{past} = \{\mathbf{k}_i | i = 1, 2, \cdots, M\}$ be the past memory bank, where $\mathbf{k}_i$ is the instance at the ith memory address, recording the features extracted from the past trajectory with social influence in the ith training sample. Correspondingly, $\mathcal{M}_\text{int} = \{\mathbf{v}_i | i = 1, 2, \cdots, M\}$ be the intention memory bank, where $\mathbf{v}_i$ is the instance at the ith address, recording the features extracted from the future intention (destination) in the ith training sample. Both the past and the intention memory banks share the same size M.

To obtain the features in the memory bank pair $\mathbf{k}_i, \mathbf{v}_i$, we propose a joint-reconstruction-based feature learning architecture; see Fig 3(a). The social encoder extracts the past feature with social influence of the past trajectory. The intention encoder extract the intention feature from the future intention (destination). The decoder receives the concatenated past-and-intention features and reconstructs the past trajectory and the future intention jointly. Mathematically, let $\mathcal{E}_\text{social}(\cdot)$ and $\mathcal{E}_\text{int}(\cdot)$ be the social encoder and intention encoder, $\mathcal{D}(\cdot)$ be the decoder, given an agent's trajectory $\mathbf{X}$, its neighbouring agents' trajectories $\mathbb{X}_\mathcal{N}$, and its future

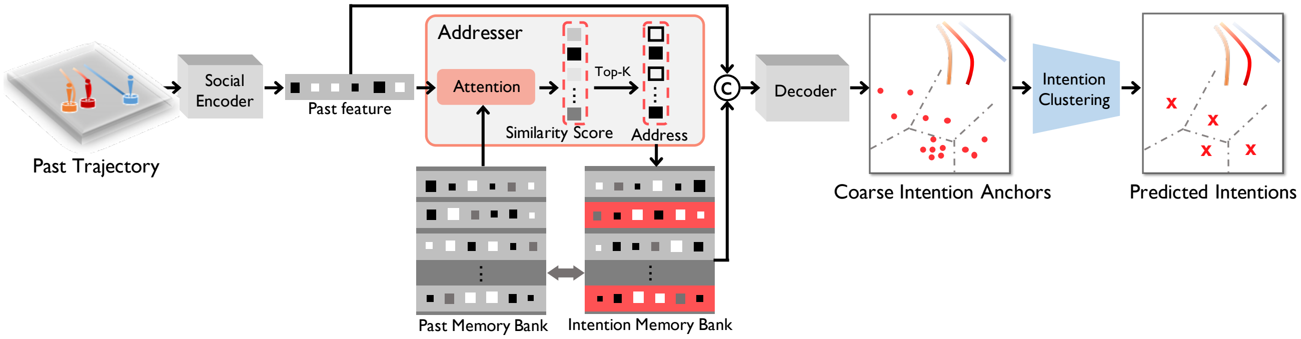


Figure 2. Inference phase of MemoNet. The red agent is to-be-predicted and the blue/orange agents are neighbours. According to the past feature obtained by the social encoder, we address related memory instances in the past memory bank through an attention network, producing similarity scores. The intention memory bank outputs future intention features for decoding coarse intention anchors according to the memory addresses with top similarity scores. At last, we utilize a clustering algorithm to obtain diverse and robust intention predictions.

intention $\mathbf{y}^{T_f}$, the joint-reconstruction process is:

$$\mathbf{k} = \mathcal{E}_{\text{social}}(\mathbf{X}, \mathbb{X}_{\mathcal{N}}),\ \mathbf{v} = \mathcal{E}_{\text{int}}(\mathbf{y}^{T_f}),\ \widehat{\mathbf{X}}, \widehat{\mathbf{y}}^{T_f} = \mathcal{D}([\mathbf{k}; \mathbf{v}]),$$

where $[\cdot;\cdot]$ represents the concatenate operation and $\widehat{\mathbf{X}}, \widehat{\mathbf{y}}^{T_f}$ denote the reconstructed past trajectory and future intention.

To optimize the feature learning architecture, we use a joint-reconstruction loss function:

$$\mathcal{L}_{\text{rec}} = \|\widehat{\mathbf{X}} - \mathbf{X}\|_2^2 + \alpha \left\|\widehat{\mathbf{y}}^{T_f} - \mathbf{y}^{T_f}\right\|_2^2,$$

where α is a weight hyperparameter. Through the proposed feature learning architecture, we obtain respective feature of the past and the intention. Their features are compatible because of the joint-reconstruction process.

Once we finish the feature learning architecture, we fix the past and the intention encoders and enumerate over all the past-intention samples in training data to initialize the past memory bank $\mathcal{M}_{\text{past}}^{(0)}$ and the intention memory bank $\mathcal{M}_{\text{int}}^{(0)}$. Specifically, for the ith past/intention sample, we use the social encoder/intention encoder to get the past feature $\mathbf{k}_i$/intention feature $\mathbf{v}_i$ storing at the ith address of the past/intention memory bank; see Fig.3(b).

Memory bank filtering. When we write all the past and intention features into the memory bank pair, many instances could be redundant, which wastes the storage. We thus propose a filtering algorithm to erase redundant memory instances and preserve representative memory instances.

For features $\mathbf{k}_i$,$\mathbf{v}_i$ at ith address in initial memory bank pair $\mathcal{M}_{\text{past}}^{(0)}$ and $\mathcal{M}_{\text{int}}^{(0)}$, we use its corresponding starting position and intention $\mathbf{x}_i^{-T_p+1}, \mathbf{y}_i^{T_f}$ to filter similar memory instances. For the ith and the jth addresses, if their memory instances have close past starting positions and future intentions, this pair of addresses is redundant and one should be removed. Mathematically, for memory instances in the ith address with its starting position $\mathbf{x}_i^{-T_p+1}$ and intention $\mathbf{y}_i^{T_f}$ and the jth address with its starting position $\mathbf{x}_j^{-T_p+1}$ and intention $\mathbf{y}_j^{T_f}$, they are redundant when:

$$\|\mathbf{x}_i^{-T_p+1} - \mathbf{x}_j^{-T_p+1}\|_2 \le \theta_{\text{past}},\ \|\mathbf{y}_i^{T_f} - \mathbf{y}_j^{T_f}\|_2 \le \theta_{\text{int}},\ (1)$$

Algorithm 1 Memory bank filtering

```
Input:  Initial memory banks $\mathcal{M}_{\text{past}}^{(0)}$, $\mathcal{M}_{\text{int}}^{(0)}$
Output: Filtered memory banks $\mathcal{M}_{\text{past}}$,$\mathcal{M}_{\text{int}}$
Initialize $\mathcal{M}_{\text{past}} = \varnothing$,$\mathcal{M}_{\text{int}} = \varnothing$
while $\mathcal{M}_{\text{past}}^{(0)} \neq \varnothing$ and $\mathcal{M}_{\text{int}}^{(0)} \neq \varnothing$ do
    Randomly pick address $i$ in $\mathcal{M}_{\text{past}}$, $\mathcal{M}_{\text{int}}$
    for all address $j$ in current $\mathcal{M}_{\text{past}}$,$\mathcal{M}_{\text{past}}$
    if Eq.(1) not satisfied for all addresses $j$ then
        Add $\mathbf{k}_i$,$\mathbf{v}_i$ into $\mathcal{M}_{\text{past}}$,$\mathcal{M}_{\text{int}}$
        Delete $\mathbf{k}_i$,$\mathbf{v}_i$ from $\mathcal{M}_{\text{past}}^{(0)}$,$\mathcal{M}_{\text{int}}^{(0)}$
    end if
end while
return $\mathcal{M}_{\text{past}}$,$\mathcal{M}_{\text{int}}$
```

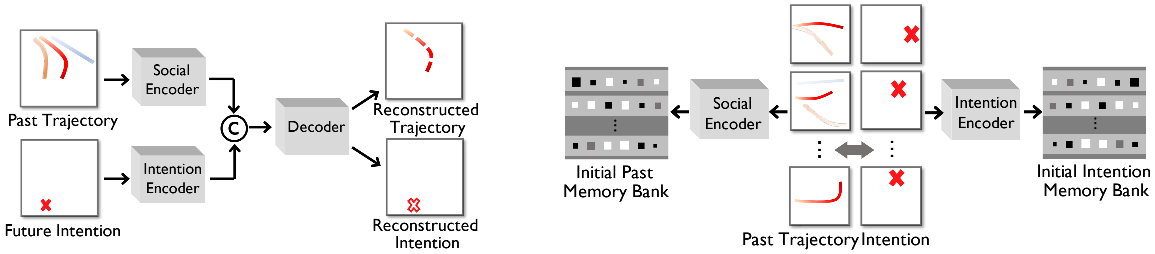
where θ_{past} and θ_{int} are two thresholds for tuning. We use this rule to filter the past and the intention memory bank; see Algorithm 1. Briefly, $\theta_{\text{past}}/\theta_{\text{int}}$ will control the memory size of the final past/intention memory banks.

Compared to previous method that uses a controller to reduce redundancy [33], our filtering has two advantages. First, our memory bank is invariant to the permutation of training samples; while in the previous method, various orderings of training samples would cause unstable memory banks. Second, our memory filter is training-free, which is more efficient; while the previous method needs to train the controller for multiple epochs.

Relations to previous methods. The proposed memory bank is similar to dictionary learning as both aim to infer a few representatives from input data to approximate incoming data, but differences include: i) a dictionary usually requires a fixed and predefine size; while the size of a memory bank is flexible and adaptive to the complexity of input data; ii) to make a prediction, the dictionary usually combines several atoms by weighted averaging; while the memory bank directly searches a single memory instance that allows an explicit link between the inference data and the training data.

4.2. Memory addresser

The functionality of a memory addresser is to search the addresses of similar past memory instances in the memory



(a) Feature learning architecture via joint reconstruction. (b) Initialization process via enumerating past-intention samples.

Figure 3. Memory bank initialization. We train a feature learning architecture by a joint-reconstruction process and initialize the memory bank by enumerating all past-intention samples using two encoders.

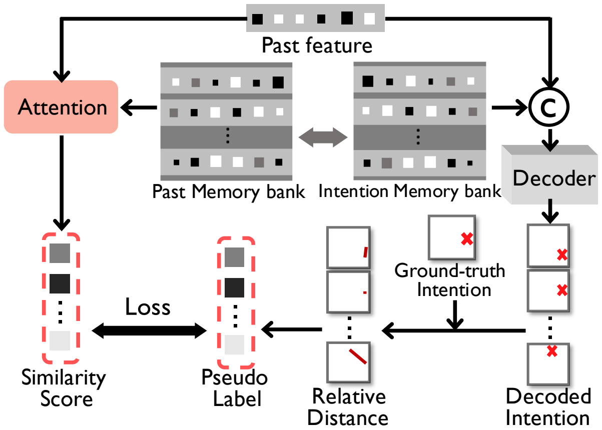


Figure 4. The addresser learning process. To train the attention network, we generate a pseudo label based on the relative distance between the decoded intentions and the ground-truth intentions.

bank for an input past trajectory feature. The key is to find an appropriate similarity metric. The previous memory addressing mechanisms leverage the cosine distance between two features as the similarity metric [11, 33]. However, any pre-defined function, including the cosine distance, might not be capable of fully reflecting the similarity between two feature vectors. To solve this issue, we propose a trainable addresser, which contains a shallow attention network to learn a similarity metric. Mathematically, given the input past feature $\mathbf{q}$ and the past memory bank $\mathcal{M}_{\text{past}} = \{\mathbf{k}_i | i = 1, 2, \cdots, |\mathcal{M}|\}$, we calculate the similarity scores across all the memory instances, which is formulated as:

$$s_i = \mathcal{F}_{\text{ATT}}(\mathbf{q}, \mathbf{k}_i) = \frac{\mathcal{F}_{\text{q}}(\mathbf{q})\mathcal{F}_{\text{k}}^{\text{T}}(\mathbf{k}_i)}{\|\mathcal{F}_{\text{q}}(\mathbf{q})\|_2 \|\mathcal{F}_{\text{k}}(\mathbf{k}_i)\|_2}, i = 1, 2, \cdots, M,$$

where $\mathcal{F}_q(\cdot)$ and $\mathcal{F}_k(\cdot)$ are two individual MLPs that transform features to a space for more appropriate distance measuring, s_i is the similarity score between the input feature and the ith memory instance. We then select the largest similarity scores and return their memory addresses.

To train such an addresser, we need to determine the "ground-truth" similarity score. Intuitively, the similarity score measured in the feature space should reflect the prediction error in the physical space. We thus consider a pseudo label which is related to the relative distance between the ground-truth intention of the input and the predicted intentions. Mathematically, let $\mathbf{y}^{T_f}$ be the ground-truth intention of the input trajectory and $\widehat{\mathbf{y}}_i^{T_f} = \mathcal{D}([\mathbf{k}_i; \mathbf{v}_i])$ be the predicted intention of the ith memory instance produced by the aforementioned intention decoder $\mathcal{D}(\cdot)$. The pseudo label the ith memory instance is defined as $\max(0, \frac{d_{\text{T}} - d_{\text{i}}}{d_{\text{T}}}) \in [0, 1]$, where $d_i = \|\mathbf{y}^{T_f} - \widehat{\mathbf{y}}_i^{T_f}\|_2$ is the relative distance between two intentions and d_T is a distance threshold. Based on this pseudo label, we train the addresser with the following loss:

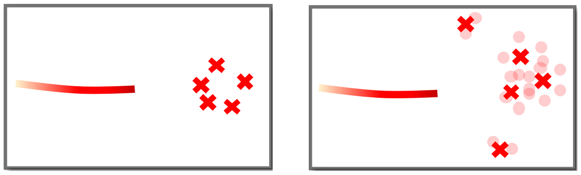
$$\mathcal{L}_{\text{Addr}} = \sum_{i=1}^{M} (s_i - \max(0, \frac{d_{\text{T}} - d_{\text{i}}}{d_T}))^2;$$

see the training process of a memory addresser in Figure 4.

4.3. Intention diversity

Fig. 5(a) illustrates a scenario that the top few searched memory instances might fall into the same modality and cannot provide sufficient diversity. The reason is that the memory bank might recall numerous seen instances like the agent will move straight in various ways, but miss other movement modalities, such as sharp left turn or right turn. Note that although simply using memory bank filtering with a large $\theta_{\text{past}}/\theta_{\text{int}}$ could promote diversity, it would remove too many memory instances and make it harder to search relevant memory instances, deteriorating the performance. To achieve a diverse prediction preserving precision, we propose an intention clustering method.

Suppose that we need to predict K possible trajectories. Here we first find L ($L \gg K$) memory instances based on the L largest similarity scores and then decode them into L intentions, which are called coarse intention anchors. We then use K-means clustering method to produce K possible intentions from the L coarse intention anchors. On one hand, since $L \gg K$, the coarse intention anchors are more likely to capture more agents' movement patterns and the clustering operation is capable to preserves these patterns to produce a more diverse prediction. On the other hand, intention clustering preserves the enrichment of the memory bank and considers multiple memory instances to cluster a predicted intention, leading to a more precise and confident intention prediction, see the example in Fig.5(b).



(a) No intention clustering (b) With intention clustering

Figure 5. Examples of intention prediction. With intention clustering, MemoNet produces a more diverse prediction.

4.4. Inference Phase

During inference, MemoNet involves four steps to obtain possible future intentions: past trajectory encoding, memory addressing, intention decoding and intention clustering; see Fig.2. First, we input the past trajectory and its neighbouring past trajectories to the social encoder $\mathcal{E}_{\text{social}}(\cdot)$ to obtain the past trajectory feature. Second, we search L most related memory instances in the past memory bank through the proposed memory addresser and return their addresses. According to the memory addresses, the intention memory bank outputs L corresponding future intention features. Third, the decoder $\mathcal{D}(\cdot)$ decodes each of L intention features into L intention anchors. Fourth, we use the proposed clustering algorithm to refine L intention anchors to K final intentions.

5. Trajectory prediction system

5.1. Trajectory fulfilling

After obtaining an agent's trajectory intentions (destinations), we fulfill the whole trajectory through an encoding-decoding process conditioned on predicted intentions; see Fig.6. Mathematically, given a predicted intention $\widehat{\mathbf{y}}^{T_f}$ of the agent's past trajectory $\mathbf{X}$ with its neighbours' past trajectories $\mathbb{X}_{\mathcal{N}}$, the trajectory fulfilling process is:

$$\mathbf{h}_{\text{x}} = \mathcal{E}_{\text{full}}(\mathbf{X},\ \mathbb{X}_{\mathcal{N}}),\ \mathbf{h}'_x = [\mathbf{h}_{\text{x}};\ \mathcal{F}_{\text{d}}(\widehat{\mathbf{y}}^{T_f})],$$
$$\widehat{\mathbf{Y}},\ \widehat{\mathbf{X}}_{\text{full}} = \mathcal{D}_{\text{full}}(\mathbf{h}'_x),$$

where $\mathcal{E}_{\text{full}}(\cdot)$ and $\mathcal{D}_{\text{full}}(\cdot)$ represent the trajectory fulfillment encoder and decoder, which share a same structure with $\mathcal{E}_{\text{social}}(\cdot)$ and $\mathcal{D}(\cdot)$, respectively. We concatenate the trajectory feature $\mathbf{h}_{\text{x}}$ with intention feature encoded by a MLP function $\mathcal{F}_{\text{d}}(\cdot)$ for whole trajectory $\widehat{\mathbf{Y}}$ decoding. To keep most past information, the fulfillment decoder also aim to reconstruct the past trajectory $\widehat{\mathbf{X}}_{\text{full}}$. To train the fulfillment encoder and decoder, we use the ℓ_2 loss:

$$\mathcal{L}_{\text{traj}} = \|\widehat{\mathbf{X}}_{\text{full}} - \mathbf{X}\|_2^2 + \beta\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_2^2,$$

where β is a weight hyperparamter.

5.2. Overall training pipeline

To train the overall system with MemoNet, we design the following training pipeline:

1. Train two encoders $\mathcal{E}_{\text{social}}(\cdot)$, $\mathcal{E}_{\text{int}}(\cdot)$ and the decoder $\mathcal{D}(\cdot)$ using the feature learning architecture with the joint-reconstruction loss $\mathcal{L}_{\text{rec}}$.

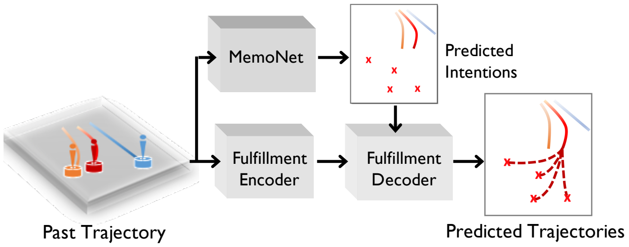


Figure 6. The inference of trajectory prediction system with MemoNet. The red color represents the to-be-predict agent and the blue/red color represent neighbours. We fulfill the whole trajectory conditioned on the prediction intentions from MemoNet.

2. Freeze two encoders $\mathcal{E}_{\text{social}}(\cdot)$, $\mathcal{E}_{\text{int}}(\cdot)$. Create the pair of initial past and intention memory banks $\mathcal{M}_{\text{past}}^{(0)}$ and $\mathcal{M}_{\text{int}}^{(0)}$ by using $\mathcal{E}_{\text{social}}(\cdot)$, $\mathcal{E}_{\text{int}}(\cdot)$. Apply memory filtering to obtain the final past and intention memory banks $\mathcal{M}_{\text{past}}$ and $\mathcal{M}_{\text{int}}$.

3. Freeze the memory banks $\mathcal{M}_{\text{past}}$, $\mathcal{M}_{\text{int}}$, the past trajectory encoders $\mathcal{E}_{\text{social}}(\cdot)$, $\mathcal{E}_{\text{int}}(\cdot)$ and the decoder $\mathcal{D}(\cdot)$. Train the memory addresser with the loss $\mathcal{L}_{\text{Addr}}$.

4. Freeze the MemoNet and train the trajectory fulfillment encoder $\mathcal{E}_{\text{full}}(\cdot)$ and decoder $\mathcal{D}_{\text{full}}(\cdot)$ with the loss $\mathcal{L}_{\text{traj}}$.

6. Experiments

6.1. Datasets

Stanford Drone Dataset (SDD): SDD is a large-scale dataset collected from campus in bird's eye view. Following [13, 32], we use the standard train-test split and predict the future 4.8s (12 frames) using past 3.2s (8 frames).

ETH-UCY: The ETH-UCY dataset contains 5 subsets, including ETH, HOTEL, UNIV, ZARA1 and ZARA2 containing various scenes captured at 2.5Hz. We use the same segment length of 8s as SDD following previous works [15, 32] and use the leave-one-out approach with 4 sets for training and the remaining set for testing.

NBA: The NBA trajectory dataset is collected by NBA using the SportVU tracking system, which reports the trajectories of the ten players and the ball in real basketball games. We randomly sample 50k samples for training and testing.

6.2. Implementation details

For MemoNet, the feature dimensions of the past memory bank and the intention memory bank are 128 and 64, respectively. On SDD, we filter the initial memory banks with $\theta_{\text{past}} = 1$, $\theta_{\text{int}} = 1$ and the coarse intention anchor number L is 120. On ETH-UCY, we filter the initial memory banks with $\theta_{\text{past}} = 0.02$, $\theta_{\text{int}} = 0.02$ and the coarse intention anchor number L is 320. The coefficients α and β in loss functions are set to 1. We train the entire framework with SGD optimizer [5]. We use an initial learning rate of 10^{-3} to train the feature learning framework, 10^{-4} to train the memory addresser, and 10^{-3} to train the trajectory fulfillment. All these modules are finetuned with a learning rate of 10^{-6}. See more details in the supplementary material.

Table 1. minADE$_{20}$ / minFDE$_{20}$ (pixels) of trajectory prediction (SDD dataset). Lower is better. The bold/underlined font represent the best/second best result. Our method achieves a **20.3%** FDE improvement compared to PECNet.

Time	Social -GAN [13]	Social-STGCNN [35]	Trajectron++ [38]	SOPHIE [37]	NMMP [15]	EvolveGraph [26]	CF-VAE [4]	MANTRA [33]	PECNet [32]	**Ours**
4.8s	27.23/41.44	20.60/33.10	19.30/32.70	16.27/29.38	14.67/26.72	13.90/22.90	12.60/22.30	8.96/17.76	9.96/15.88	**8.56/12.66**

Table 2. minADE$_{20}$ / minFDE$_{20}$ (meters) of trajectory prediction (ETH-UCY dataset). Lower is better. The bold/underlined font represent the best/second best result. Our method achieves a **10.2%** FDE improvement compared to Agentformer.

Subset	Social-GAN [13]	STGAT [16]	NMMP [15]	MANTRA [33]	Transformer -TF [8]	STAR [46]	PECNet [32]	Trajectron++ [38]	Agentformer [47]	**Ours**
ETH	0.87/1.62	0.65/1.12	0.61/1.08	0.48/0.88	0.61/1.12	**0.36**/0.65	0.54/0.87	0.39/0.83	0.45/0.75	0.40/**0.61**
HOTEL	0.67/1.37	0.35/0.66	0.33/0.63	0.17/0.33	0.18/0.30	0.17/0.36	0.18/0.24	0.12/0.21	0.14/0.22	**0.11/0.17**
UNIV	0.76/1.52	0.52/1.10	0.52/1.11	0.37/0.81	0.35/0.65	0.31/0.62	0.35/0.60	**0.20**/0.44	0.25/0.45	0.24/**0.43**
ZARA1	0.35/0.68	0.34/0.69	0.32/0.66	0.27/0.58	0.22/0.38	0.29/0.52	0.22/0.39	**0.15**/0.33	0.18/**0.30**	0.18/0.32
ZARA2	0.42/0.84	0.29/0.60	0.43/0.85	0.30/0.67	0.17/0.32	0.22/0.46	0.17/0.30	**0.11**/0.25	0.14/**0.24**	0.14/**0.24**
AVG	0.61/1.21	0.43/0.83	0.41/0.82	0.32/0.65	0.31/0.55	0.26/0.53	0.29/0.48	**0.19**/0.41	0.23/0.39	0.21/**0.35**

Table 3. minADE$_{20}$ / minFDE$_{20}$ (meters) of trajectory prediction (NBA dataset). Lower is better. The bold/underlined font represent the best/second best result. Our method achieves a **28.3%** FDE improvement compared to NMMP.

Time	Social-LSTM [1]	Social-GAN [13]	Social-STGCNN [35]	STGAT [16]	NRI [20]	STAR [46]	PECNet [32]	NMMP [15]	**Ours**
4.0s	1.79/3.16	1.62/2.51	1.59/2.37	1.41/2.22	2.06/3.74	1.26/2.04	1.83/3.41	1.33/2.05	**1.25/1.47**

6.3. Quantitative results

Two used evaluation metrics are the minimum average displacement error ($\min\text{ADE}_K$), which is the minimum among K time-averaged distances of predicted trajectories compared to the ground-truths, and the minimum final displacement error ($\min\text{FDE}_K$), which is the minimum distance among K predicted endpoints to the ground-truth endpoints.

On SDD dataset, we compare our method with current 9 state-of-the-art prediction methods; see Table 1. We see that i) our MemoNet significantly outperforms all baselines in intention prediction measured by FDE. Our method reduces FDE from 15.88 to 12.66 compared to the current state-of-the-art method, PECNet, achieving **20.3%** improvement; ii) with a more precise intention prediction, our method predicts the whole trajectory more accurately. Our method outperforms PECNet by **14.1%** in ADE.

On ETH-UCY dataset, we compare our method with 9 prediction methods; see Table 2. We see that i) MemoNet outperforms competitive methods in predicting intention measured by FDE. Specifically, our method reduces the average FDE from 0.39 to 0.35 compared to the previous state-of-the-art method, AgentFormer, achieving **10.2%** improvement; and ii) our method achieves the best or close to the best performance in ADE over all the five subsets.

On NBA dataset, we compare our proposed method with 8 prediction methods; see Table 3. We see that MemoNet reduces FDE from 2.05 to 1.47 compare to the current state-of-the-art method, NMMP, achieving **28.3%** improvement.

6.4. Qualitative results

Visualization of diverse intention. Fig.7 illustrates the diverse intention prediction with MemoNet, where the pink dots are coarse intention anchors. We see that with the help of intention clustering, MemoNet can provide diverse and accurate intention predictions.

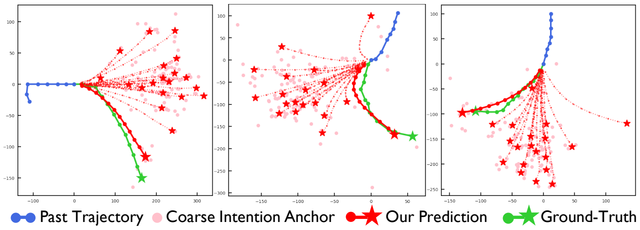


Figure 7. Diverse intention prediction by MemoNet on SDD, where 20 final intentions are clustered from 120 coarse intention anchors. MemoNet can provide diverse and accurate intention predictions.

Visualization of predicted trajectory. Fig.8 compares the best-of-20 predicted trajectories produced by our MemoNet and previous state-of-the-art method PECNet and MANTRA. We see that our predictions (red) are closer to the ground-truth (green) than other two methods. Especially, for challenging direction-turning cases (third column), previous methods fail to capture the right direction; while our MemoNet still provides precise prediction.

Visualization of explicit link. Fig.9 shows prediction cases with their seen past-future trajectory instances traced by the addressed memory instances. We see that seen similar scenarios provide instance-level experience to obtain multi-modal future intentions and reflects that our model can trace back to specific memorized samples during the prediction.

6.5. Ablation studies

Effect of components in MemoNet. We explore the effect of each of four proposed key components in MemoNet, including memory bank, memory filtering, memory addresser and intention clustering. Table 4 presents the results. We see that i) the proposed memory bank can sig-

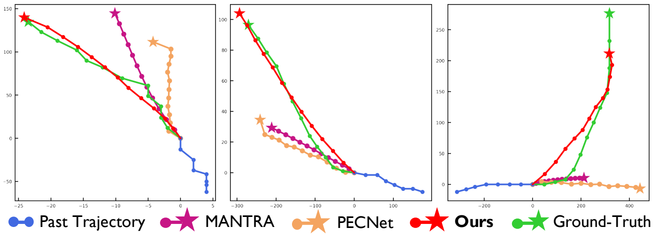


Figure 8. We compare the best-of-20 predicted trajectories produced by our method and two previous methods on SDD. Our method achieves a more precise trajectory prediction.

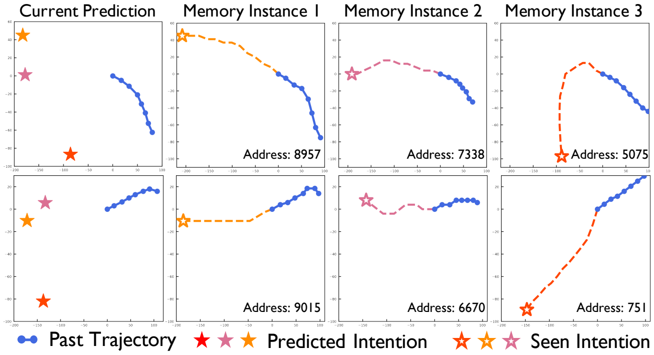


Figure 9. Prediction cases with corresponding past-future trajectories traced by memory addresser. Our model promotes a more explicit link between the current situation and seen instances.

Table 4. Ablation study of each component in MemoNet on the SDD and ETH dataset. ○/✓ represent using cosine distance/learnable addresser. Each component is beneficial.

Memory Bank	Memory Filtering	Memory Addresser	Intention Clustering	SDD	ETH
				14.16/27.76	0.78/1.44
✓		○		9.64/15.25	0.55/0.94
✓	✓	○		9.59/15.08	0.55/0.93
✓	✓	✓		9.50/14.78	0.53/0.89
✓	✓	✓	✓	**8.56/12.66**	**0.40/0.61**

nificantly improve the prediction performance; and ii) the memory filtering, learnable addresser and intention clustering all contribute to promoting accurate prediction.

Effect of the number of coarse intention anchors. Fig.10 illustrates the influence of coarse intention anchor numbers L. We find that either too small or too large L causes performance degeneration as i) when L is small, the model tends to miss intention modality, causing insufficient diversity and worse prediction performance; and ii) when L is too large, the prediction involves too many irrelevant instances, also resulting in worse prediction performance.

Table 5. Ablation study of thresholds $\theta_{\text{past}}/\theta_{\text{int}}$ in memory filtering on SDD. $\theta_{\text{past}} = \theta_{\text{int}} = 1$ achieves the best performance.

$\theta_{\text{past}}/\theta_{\text{int}}$	$\text{minADE}_{20}/\text{minFDE}_{20}$	Memory Size	Storage
0	8.65/12.84	17970 (100.0%)	13.8MB
0.5	8.59/12.70	15442 (85.9%)	11.9MB
1	**8.56/12.66**	14652 (81.5%)	11.2MB
5	9.22/14.29	10698 (59.5%)	8.2MB
10	9.64/15.57	6635 (36.9%)	5.1MB
20	10.41/17.32	2692 (15.0%)	2.1MB
50	13.77/25.86	604 (3.4%)	465KB

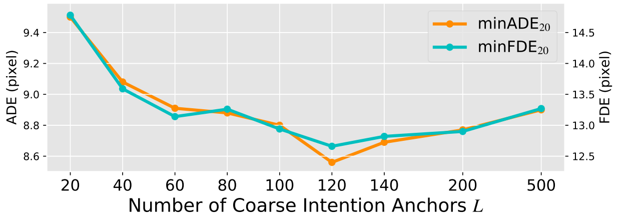


Figure 10. ADE/FDE as a function of the number of coarse intention anchors L on SDD. $L = 120$ provides the best performance.

Effect of thresholds in memories filtering. Table 5 reports the prediction errors with various thresholds $\theta_{\text{past}}/\theta_{\text{int}}$ in memory filtering. We see that i) an appropriate $\theta_{\text{past}}/\theta_{\text{int}}$ leads to a remarkable performance and lightweight storage; ii) when $\theta_{\text{past}}/\theta_{\text{int}}$ are too small, the model tends to preserve redundant information and decrease the intention diversity, wasting the storage and affecting the performance; and iii) when $\theta_{\text{past}}/\theta_{\text{int}}$ are too large, a large amount of useful information are filtered out, which makes it harder to find relevant instances, deteriorating the performance.

Real-time inference speed. We run the whole inference model for 10 times on SDD dataset using one RTX-3090 GPU. The average prediction time is 18.03ms per sample, with a real-time predictions FPS=55.5, much faster than the common sampling rate of data collection.

7. Conclusion

This paper proposes MemoNet, an instance-based approach that is designed based on the retrospective memory mechanism, where the seen instances are stored into a memory bank pair during training and could be used for relevant movement pattern matching during inference. The proposed MemoNet includes four key designs: a joint-reconstruction-based feature-learning architecture, a memory filtering algorithm, a learnable addresser, and an intention clustering method. Experiments show that our method significantly improves the state-of-the-art performance on trajectory prediction datasets and has the ability to trace back to specific instances during prediction, promoting more interpretability.

Limitation and future work. In this paper, we focus on memorizing past-intention pairs. However, it is a challenge to predict some special actions only using past trajectories, such as a sharp turn. A future work is to utilize the map information to generate environment conditioned prediction.

Acknowledgements

This research is partially supported by the National Key R&D Program of China under Grant 2021ZD0112801, National Natural Science Foundation of China under Grant 62171276, the Science and Technology Commission of Shanghai Municipal under Grant 21511100900 and CCF-DiDi GAIA Research Collaboration Plan 202112.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 2, 7

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 3

[3] Alan Baddeley. *Essentials of human memory (classic edition)*. Psychology Press, 2013. 2

[4] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019. 7

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*, pages 177–186, 2010. 6

[6] Dario Floreano and Robert J Wood. Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460–466, 2015. 1

[7] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 2

[8] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342. IEEE, 2021. 7

[9] Patricia S Goldman-Rakic. Cellular basis of working memory. *Neuron*, 14(3):477–485, 1995. 2

[10] Colin Graber and Alexander G Schwing. Dynamic neural relational inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8513–8522, 2020. 1, 2

[11] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 3, 5

[12] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016. 3

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 1, 2, 6, 7

[14] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[15] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6319–6328, 2020. 1, 2, 6, 7

[16] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6272–6281, 2019. 1, 2, 7

[17] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2375–2384, 2019. 1, 2

[18] Takayuki Kanda, Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, and Ryohei Nakatsu. Development and evaluation of an interactive humanoid robot" robovie". In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1848–1855. IEEE, 2002. 1

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

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

[21] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European conference on computer vision*, pages 201–214. Springer, 2012. 2

[22] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR, 2016. 3

[23] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 1, 2

[24] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 163–168. IEEE, 2011. 1

[25] Jiachen Li, Fan Yang, Hengbo Ma, Srikanth Malla, Masayoshi Tomizuka, and Chiho Choi. RAIN: reinforced hybrid attention inference network for motion forecasting. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16076–16086, 2021. 2

[26] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 7

[27] Maosen Li, Siheng Chen, Yanning Shen, Genjia Liu, Ivor W Tsang, and Ya Zhang. Online multi-agent forecasting with in-

terpretable collaborative graph neural network. *arXiv preprint arXiv:2107.00894*, 2021. 2

[28] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019. 2

[29] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, pages 541–556. Springer, 2020. 2

[30] Chao Ma, Chunhua Shen, Anthony Dick, Qi Wu, Peng Wang, Anton van den Hengel, and Ian Reid. Visual question answering with memory-augmented networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6975–6984, 2018. 3

[31] Hengbo Ma, Yaofeng Sun, Jiachen Li, Masayoshi Tomizuka, and Chiho Choi. Continual multi-agent interaction behavior prediction with conditional generative memory. *IEEE Robotics Autom. Lett.*, 2021. 3

[32] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pages 759–776. Springer, 2020. 1, 2, 6, 7

[33] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7143–7152, 2020. 3, 4, 5, 7

[34] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942. IEEE, 2009. 2

[35] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020. 1, 2, 7

[36] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1289–1298, 2016. 2

[37] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 7

[38] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. 2020. 1, 2, 7

[39] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016. 3

[40] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015. 3

[41] Bohan Tang, Yiqi Zhong, Ulrich Neumann, Gang Wang, Ya Zhang, and Siheng Chen. Collaborative uncertainty in multi-agent trajectory forecasting. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[42] Maria Valera and Sergio A Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings-Vision, Image and Signal Processing*, 152(2):192–204, 2005. 1

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

[44] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. 2

[45] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 3

[46] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020. 1, 2, 7

[47] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021. 1, 2, 7