2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

S^3-Net: A Single-stage Spectrum-domain Network for Trajectory Prediction

Beihao Xia, Qinmu Peng, and Xinge You (✉)

***Abstract*— Trajectory prediction is a fundamental yet challenging task in intelligent systems. Existing methods are mainly categorized as single-stage time-domain, two-stage time-domain, or two-stage spectrum-domain approaches, while single-stage spectrum-domain methods have been relatively underexplored. In the frequency domain, low-frequency components reflect global motion trends, while high-frequency components capture fine-grained local variations. Most existing spectrum-domain approaches process these components independently, overlooking their intrinsic complementarity. Inspired by the success of bilinear models in explicitly capturing cross-factor interactions, we propose S^3-Net, a single-stage spectrum-domain trajectory prediction network with a bilinear fusion module that integrates low- and high-frequency dynamics. This design yields richer spectral representations and enables accurate, socially compliant, and multimodal predictions. Experiments on the ETH-UCY and Stanford Drone Datasets demonstrate that S^3-Net achieves up to 16.8%/15.1% ADE/FDE reduction over spectrum-domain baselines while maintaining a compact model size and low inference latency, making it suitable for real-time scenarios.**

I. Introduction

Understanding, analyzing, and predicting the future behaviors of agents are fundamental to building safe and reliable intelligent systems [1]. Trajectory prediction aims to forecast plausible future motion paths of agents based on historical observations and interactions [2], [3]. It has been widely applied in autonomous driving [4], [5], robot navigation [6], [7], motion planning [8], [9], activity perception [10], [11], and multi-target tracking [12], [13].

With continuous efforts from the research community, trajectory prediction has achieved remarkable progress. Existing deep learning approaches can generally be categorized into four groups, as illustrated in Fig. 1:

(a) *Single-stage Time-domain Methods* ("S & T"). The pioneering work Social-LSTM [2] formulates trajectory prediction as a time-series generation problem. Following this paradigm, a variety of architectures such as Recurrent Neural Networks (RNNs) [14]–[16], Generative Adversarial Networks (GANs) [17]–[19], and Transformers [20]–[22] have been employed to forecast future trajectories. However, these methods struggle to capture multi-scale motion dynamics.

(b) *Two-stage Time-domain Methods* ("T & T"). Inspired by humans' goal-driven behavior, another line of research [23]–[25] reformulates trajectory prediction into a two-stage process of "goal estimation" followed by "trajectory reconstruction". For instance, Mangalam *et al.* [23] first predict potential destinations and then refine the trajectories through social interaction modeling. However, such methods often suffer from error propagation across stages.

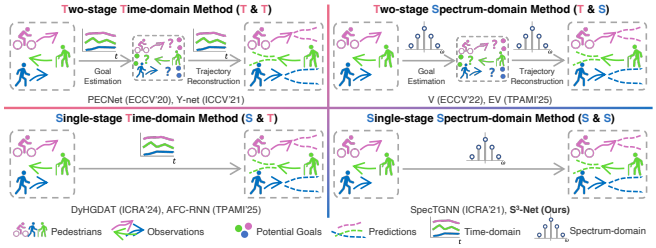

Fig. 1. Simplified pipelines of four types of trajectory prediction methods.

(c) *Two-stage Spectrum-domain Methods* ("T & S"). To capture motion variations at multiple temporal scales, researchers [26], [27] have applied the Discrete Fourier Transform (DFT) to represent trajectories in the frequency domain. Although the spectral view reveals distinct frequency-response characteristics, the multi-stage design inevitably increases inference latency and deployment complexity.

(d) *Single-stage Spectrum-domain Methods* ("S & S"). Recent studies attempt to simplify the spectral pipeline [28], [29]. Cao *et al.* [30], [31] construct a spectral temporal graph network to model interactive behaviors in trajectory prediction. However, these methods primarily focus on modeling social interaction in the frequency domain while neglecting latent behavioral information across frequency components.

Despite these advances, spectrum-domain approaches typically treat frequency components independently, overlooking their inherent complementarity. In practice, trajectory prediction involves two coupled factors: the *global motion trend* captured by low-frequency components and the *fine-grained local variations* encoded by high-frequency components [26]. Modeling these two factors jointly is crucial, as agents usually follow a long-term goal while making subtle short-term adjustments in response to interactions.

To address this limitation, we draw inspiration from *bilinear models* [32], [33], which have shown strong representational power in fine-grained recognition [34], visual question answering [35], and multi-modal learning [36]. The key advantage of bilinear formulations lies in their outer-product representation, which explicitly models cross-factor interactions and provides a richer feature space than simple concatenation or addition. Motivated by this, we introduce a bilinear fusion module to integrate low- and

Corresponding author: Xinge You. This work was supported in part by the National Natural Science Foundation of China under Grant 62575116.

All authors are with the School of Electronic Information and Communications, National Anti-Counterfeit Engineering Research Center, Huazhong University of Science and Technology, Wuhan 430074, China. (✉: {xbh_hust, pengqinmu, youxg}@hust.edu.cn).

high-frequency dynamics in a single-stage spectrum-domain framework, thereby enabling more expressive representations for accurate and socially compliant trajectory prediction.

Therefore, we propose S^3-Net, a single-stage spectrum-domain trajectory prediction model. Specifically, we apply the Discrete Fourier Transform to decompose trajectories into low- and high-frequency components. To capture their interrelations, we incorporate a bilinear module that fuses frequency features into richer behavioral representations. S^3-Net generates multiple plausible trajectories with compact model size and low inference latency, while explicitly considering social interactions and scene constraints to ensure socially compliant and physically feasible predictions.

The main contributions are summarized as follows: (a) We propose S^3-Net, a single-stage spectrum-domain model with frequency interaction modeling, which achieves a favorable balance between accuracy, efficiency, and model size. (b) We introduce a bilinear fusion module to capture the complementary dynamics between low- and high-frequency components, enhancing the representational capacity of spectrum-domain approaches. (c) Extensive experiments on ETH-UCY and SDD show the effectiveness and competitiveness of S^3-Net.

II. Related Work

Time-domain Trajectory Prediction. Most existing approaches [37]–[41] formulate trajectory prediction as a sequence generation problem, adopting architectures such as RNNs, LSTMs, GANs, CVAEs and Transformers to model motion patterns [42], social interactions [43], and multimodal behaviors [44]. Although effective, time-domain methods often emphasize temporal correlations, while struggling to capture multi-scale motion dynamics and subtle behavioral variations. This limitation motivates the exploration of frequency-domain representations, which naturally decompose trajectories into multiple scales.

Hierarchical Trajectory Prediction. Another line of research leverages human goal-driven behavior by decomposing prediction into two stages, namely "goal estimation" and "trajectory reconstruction" [23], [25], [45], [46]. These hierarchical methods provide interpretability but are prone to cumulative errors and incur high inference latency, making them less suitable for real-time applications.

Bilinear Models. Bilinear models [27], [32], [33], [47], [48] explicitly capture interactions between two factors through outer-product representations. They have been widely applied in fine-grained recognition [49], visual question answering [50], and re-identification [34], demonstrating strong representational power. However, their potential for modeling interactions between frequency components in trajectory prediction remains unexplored, which provides the key insight motivating our bilinear fusion design.

In this work, we view frequency-domain trajectory prediction as a two-factor problem, where low-frequency components encode global trends and high-frequency components capture local variations. Unlike prior spectrum-based approaches that process these independently, our method introduces a bilinear fusion module to integrate them, yielding richer spectral representations and enabling more accurate and efficient predictions.

III. Method

A. Vanilla Bilinear Model

The vanilla bilinear model is used to capture two-factor variations such as "location" and "appearance" [32], [48]. As mentioned in [48], the bilinear structure $\mathcal{B}$ for feature refinement consists of four components. Formally,

$$\mathcal{B} = (f_A, f_B, \mathcal{P}, \mathcal{E}), \tag{1}$$

where f_A and f_B are feature functions, $\mathcal{P}$ is the max pooling function, and $\mathcal{E}$ represents an encoding function.

In a two-factor task, let $\mathcal{U}$ and $\mathcal{V}$ denote the feature spaces of the two factors, respectively. We define two feature functions $f_A : \mathcal{U} \times \mathcal{V} \to \mathbb{R}^K$ and $f_B : \mathcal{U} \times \mathcal{V} \to \mathbb{R}^D$. Given two representations $u \in \mathcal{U}$ and $v \in \mathcal{V}$, the bilinear feature $f_b \in \mathbb{R}^{K \times D}$ is computed as the outer product:

$$f_b = f_A(u, v) \otimes f_B(u, v) = f_A(u, v)\, f_B(u, v)^\top \in \mathbb{R}^{K \times D}. \tag{2}$$

Then, we obtain the refined feature f_R through max pooling and encoding operations, *i.e.*,

$$f_R = \mathcal{E}\left(\mathcal{P}(f_b)\right). \tag{3}$$

B. The Architecture of S^3-Net

As illustrated in Fig. 2, we design S^3-Net with an encoder-decoder architecture. First, the Discrete Fourier Transform is applied to obtain trajectory spectra consisting of low- and high-frequency components. Second, a bilinear module is introduced to fuse these frequency features into richer representations. Leveraging these bilinear features, S^3-Net generates multiple socially compliant trajectory predictions.

Problem Formulation. Suppose M agents' observed and predicted trajectories are represented by $X = \{X^i\}_{i=1}^M$ and $\hat{Y} = \{\hat{Y}^i\}_{i=1}^M$, respectively. Moreover, $X^i = \{(x_t^i, y_t^i)\}_{t=1}^{t_{obs}}$ and $\hat{Y}^i = \{(\hat{x}_t^i, \hat{y}_t^i)\}_{t=t_{obs}+1}^{t_{obs}+t_{pre}}$ denote the i-th agent's coordinates at the observed time $t = 1, 2, \ldots, t_{obs}$ and at the predicted time $t = t_{obs}+1, t_{obs}+2, \ldots, t_{obs}+t_{pre}$. In this work, we aim to find a predictor to generate future trajectories $\hat{Y}$ based on the observed trajectories X and the scene images I. For clarity, we omit the symbol $i \in [1, M]$, which is used to indicate the agent index.

(a) Trajectory Spectra. We first apply DFT to the observed trajectories to obtain their corresponding trajectory spectra. Specifically, we apply 1D-DFT on different dimensions of the observed trajectory $X = \{(x_t, y_t)\}_{t=1}^{t_{obs}}$ to obtain their spectra, including amplitudes $A = \{a_x, a_y\}$ and phases $\Phi = \{\phi_x, \phi_y\}$. Formally,

$$\begin{aligned}
\mathcal{X} &= \mathrm{DFT}[(x_1, x_2, \ldots, x_{t_{obs}})] \in \mathbb{C}^{t_{obs}}, \\
\mathcal{Y} &= \mathrm{DFT}[(y_1, y_2, \ldots, y_{t_{obs}})] \in \mathbb{C}^{t_{obs}}, \\
A &= \{a_x, a_y\} = \{|\mathcal{X}|, |\mathcal{Y}|\} \in \mathbb{R}^{t_{obs} \times 2}, \\
\Phi &= \{\phi_x, \phi_y\} = \{\arg \mathcal{X}, \arg \mathcal{Y}\} \in \mathbb{R}^{t_{obs} \times 2}.
\end{aligned} \tag{4}$$

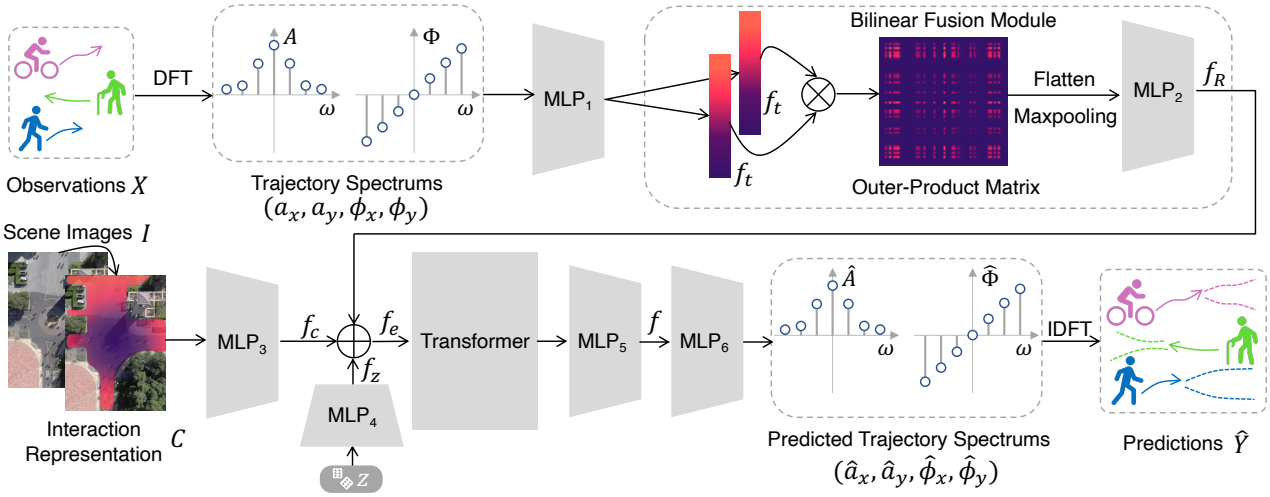

Fig. 2. Overview of S^3-Net. Detailed layer connections, output units, and activations of the whole network architecture are listed in Tab. I.

Then we employ a multi-layer perceptron (MLP) (denoted as MLP_1 in Tab. I) to embed the observed trajectory spectra into the high-dimensional feature f_t. Formally,

$$f_t = \mathrm{MLP}_1([A, \Phi]) \in \mathbb{R}^{t_{obs} \times 64}. \tag{5}$$

(b) Bilinear Fusion Module. Following the workflow of bilinear models, the bilinear fusion module operates on the trajectory spectra to construct relations between different frequency components. Meanwhile, the feature functions share all their computations. Then, we apply max pooling and the encoder MLP (denoted MLP_2) to obtain the refined feature f_R. Formally,

$$f_R = \mathrm{MLP}_2\left(\mathrm{Flatten}\left(\mathrm{MaxPool}(f_t \otimes f_t)\right)\right) \in \mathbb{R}^{t_{obs} \times 64}. \tag{6}$$

(c) Interaction Modeling. Moreover, interactions should also be considered, although they are not the main focus of this work. In detail, we employ the interaction representation C [3][1] to achieve this goal. Then, we adopt an MLP (denoted MLP_3) to obtain the context feature $f_c = \mathrm{MLP}_3(C)$, and compute the joint feature $f_j = [f_R, f_c]$ by concatenating the refined and context features. To model the multimodality of agents' potential future trajectories, we sample the random noise vector $z \sim \mathcal{N}(0, I)$ for N times during inference, and concatenate the corresponding random representation f_z with f_j. Specifically, we use another MLP (denoted MLP_4) with the same structure as MLP_1 to encode the noise vector z:

$$\begin{aligned} f_z &= \mathrm{MLP}_4(z) \in \mathbb{R}^{t_{obs} \times 64}, \\ f_e &= [f_j, f_z] \in \mathbb{R}^{t_{obs} \times 128}. \end{aligned} \tag{7}$$

Here, $[a, b]$ represents the concatenation for vectors $\{a, b\}$ on the last dimension.

(d) Trajectory Decoding. We introduce a Transformer (denoted "Trans") to obtain agents' behavior representations. The Transformer serves as a feature extractor without the final output layer. The embedded vector f_e is passed to the Transformer. We employ another MLP encoder (denoted MLP_5) to aggregate features at different frequency nodes, inferring the behavior feature f. Formally,

$$f = \mathrm{MLP}_5\left(\mathrm{Trans}(f_e)\right) \in \mathbb{R}^{t_{obs} \times 128}. \tag{8}$$

A decoder MLP (denoted MLP_6) is adopted to predict trajectory spectra $[\hat{A}, \hat{\Phi}] = (\hat{a}_x, \hat{a}_y, \hat{\phi}_x, \hat{\phi}_y)$. Meanwhile, the Inverse DFT (IDFT) is applied to obtain the predicted spatial trajectory $\hat{Y}$. Formally,

$$\begin{aligned} &(\hat{a}_x, \hat{a}_y, \hat{\phi}_x, \hat{\phi}_y) = \mathrm{MLP}_6(f) \in \mathbb{R}^{t_{pre} \times 4}, \\ &\hat{Y} = (\mathrm{IDFT}[\hat{a}_x \exp(j\,\hat{\phi}_x)], \mathrm{IDFT}[\hat{a}_y \exp(j\,\hat{\phi}_y)]) \in \mathbb{R}^{t_{pre} \times 2}. \end{aligned} \tag{9}$$

(e) Loss Function. When training the network end to end, the entire ground-truth future trajectories Y will be used as the supervision. The network parameters are optimized by minimizing the average Euclidean distance between Y and predictions $\hat{Y}$, thus learning to predict the corresponding trajectory spectra. Formally,

$$\mathcal{L} = \frac{1}{t_{pre}} \sum_{t=t_{obs}+1}^{t_{obs}+t_{pre}} \|(\hat{x}_t, \hat{y}_t) - (x_t, y_t)\|_2. \tag{10}$$

IV. Experiments

A. Experimental Setup

Datasets. Following previous works [2], [27], [51], [52], we choose two widely used datasets, ETH-UCY [12] and the Stanford Drone Dataset (SDD) [53], to evaluate performance. (a) *ETH-UCY Benchmark* consists of five subsets: eth, hotel, univ, zara1, zara2. It contains trajectories from 1,536 pedestrians, including thousands of nonlinear paths. Annotations provide pedestrians' coordinates in meters. (b) *Stanford Drone Dataset* has 60 bird's-eye-view videos captured by drones. More than 11,000 different agents are annotated with bounding boxes in pixels, including over 185,000 agent-to-agent interactions and 40,000 agent-to-scene interactions.

[1]First, available trajectories are used to obtain the activity probability for each region of the images I as supervision. Second, a CNN is trained under this supervision to directly infer the activity probability from scene images.

TABLE I

ARCHITECTURE DETAILS OF THE PROPOSED S^3-NET.

Layers	Network Architecture
MLP_1	$(a_x, a_y, \phi_x, \phi_y) \rightarrow \text{FC}(64, \text{ReLU})$ $\rightarrow \text{FC}(64, \tanh) \rightarrow f_t$
MLP_2	$f_t \otimes f_t \rightarrow \text{MaxPool}(2 \times 2) \rightarrow \text{Flatten}$ $\rightarrow \text{FC}(64\, t_{obs}, \tanh) \rightarrow \text{Reshape}(t_{obs}, 64) \rightarrow f_R$
MLP_3	$C \rightarrow \text{MaxPool}(5 \times 5) \rightarrow \text{Flatten}$ $\rightarrow \text{FC}(64\, t_{obs}, \tanh) \rightarrow f_c$
MLP_4	$z \rightarrow \text{FC}(64, \text{ReLU}) \rightarrow \text{FC}(64, \tanh) \rightarrow f_z$
Trans	$f_e \rightarrow \text{TransformerEncoder}(128) \rightarrow f'$ $\rightarrow \text{TransformerDecoder}(128) \rightarrow f''$
MLP_5	$f'' \rightarrow \text{FC}(128, \tanh) \rightarrow \text{FC}(128) \rightarrow f$
MLP_6	$f \rightarrow \text{FC}(128, \tanh) \rightarrow \text{FC}(128, \text{ReLU}) \rightarrow \text{FC}(4\, t_{pre})$ $\rightarrow \text{Reshape}(t_{pre}, 4) \rightarrow (\hat{a}_x, \hat{a}_y, \hat{\phi}_x, \hat{\phi}_y)$

Metrics. We employ two standard metrics to evaluate prediction accuracy, namely the Average Displacement Error (ADE) and the Final Displacement Error (FDE) [2], [54]. (a) *ADE* is the average point-wise Euclidean distance between the ground truth and predictions of all steps. (b) *FDE* is the Euclidean distance between the last point's prediction and the ground truth. Following [17], [26], we use a "Best-of-N" strategy to compute ADE and FDE with $N = 20$.

Baselines. We compare against four categories of methods for a comprehensive evaluation. (a) *Single-stage Time-domain methods* ("S & T"): SHENet [55], MID [56], SEEM [57], Flowchain [58], Eqmotion [59], IMP [60], LED [61], DyHGDAT [62], MS-TIP [63], SMEMO [64], LMTraj-SUP [65], Dyset [66], TrajCLIP [67], AFC-RNN [68]. (b) *Two-stage Time-domain methods* ("T & T"): PECNet [23], Y-net [24], LB-EBM [45], GSMNet [46], MSN-SC [52], PPT [69]. (c) *Two-stage Spectrum-domain methods* ("T & S"): V [26]. (d) *Single-stage Spectrum-domain methods* ("S & S"): SpecTGNN [31].

Implementation Details. We predict future trajectories over $t_{pre} = 12$ frames (4.8 seconds) given $t_{obs} = 8$ observed frames (3.2 seconds). The frame rate is set to 2.5 frames per second when sampling trajectories. We train the entire S^3-Net with the Adam optimizer with a learning rate of 0.0003 on a single NVIDIA GeForce GTX 1080Ti GPU. S^3-Net is trained with a batch size of 2500 for 1000 epochs on ETH-UCY and 800 epochs on SDD. We employ a 4-layer encoder-decoder Transformer structure with 8 attention heads. The output dimension of fully connected layers in multi-head attention blocks is set to 128. The leave-one-out strategy [2] is adopted to train the network end-to-end.

B. Comparison to State-of-the-Art Methods

ETH-UCY. As reported in Tab. II, the proposed S^3-Net exhibits strong competitiveness against various categories of methods. S^3-Net (0.17/0.27) achieves performance comparable to several top-performing models, TrajCLIP (0.18/0.33) [67], Y-net (0.18/0.27) [24], and V (0.18/0.28) [26]. Although the average gain is relatively modest, S^3-Net ranks in the top two across three subsets (eth, hotel, and univ), highlighting its consistent effectiveness across diverse scenarios within ETH-UCY. To further validate its superiority, we extend the evaluation to the more challenging SDD benchmark.

SDD. On the SDD, S^3-Net achieves notable improvements over all categories of baselines, as shown in Tab. III. Compared to the current state-of-the-art "S & T" approach AFC-RNN [68], S^3-Net reduces ADE/FDE by 4.7%/7.9%. It also surpasses the "T & T" method PPT [69] and the "T & S" approach V [26], with gains of 2.8%/1.0% and 4.1%/7.0%, respectively. Moreover, against the "S & S" baseline SpecT-GNN [31], S^3-Net achieves substantial improvements of 16.8%/15.1% in ADE/FDE. These results demonstrate the robustness and adaptability of S^3-Net in handling complex and diverse real-world scenarios.

C. Quantitative Analysis

Effect of the Bilinear Fusion Module. Tab. IV demonstrates that S^3-Net consistently outperforms its variant without the bilinear fusion module (S^3-Net w/o BFM). Specifically, S^3-Net achieves significant improvements of 15.00%/12.90% on ETH-UCY and 13.76%/12.75% on SDD in terms of ADE/FDE, respectively. These consistent gains across two distinct datasets validate the effectiveness of the bilinear fusion module. To further analyze its role, we introduce V (linear) (Variation c in [26]). V (linear) decomposes trajectories into high- and low-frequency components, predicts potential destinations in the first stage, and applies linear interpolation in the second stage. Viewed from this perspective, V (linear) can be regarded as an "S & S" method without a bilinear module. Its quantitative results are 0.19/0.29 on ETH-UCY and 7.43/11.15 on SDD. By comparing V (linear), S^3-Net w/o BFM, and S^3-Net, we conclude that the bilinear fusion module significantly enhances the model's representation capacity.

Single-stage vs Two-stage. As shown in Tab. II, different categories of methods exhibit their respective strengths. In this section, our goal is not to revalidate the effectiveness of spectrum-based approaches, which has already been demonstrated in [26], [27], [31]. For a fair comparison, we select V as the baseline since both our proposed S^3-Net and V are spectrum-domain methods built upon Transformer backbones. Compared with V, S^3-Net achieves moderate improvements of 5.6%/3.6% on ETH-UCY and 4.1%/7.5% on SDD in terms of ADE/FDE. Although the gains are not particularly significant, the results highlight the advantage of adopting a single-stage design. Moreover, it may be insufficient to evaluate performance solely from the perspective of "Accuracy." Therefore, we proceed to conduct a deeper analysis focusing on "Model Size and Efficiency."

Model Size and Efficiency. Trajectory prediction models should not only ensure accuracy but also reduce model size and inference time to enhance real-world applicability. As reported in Tab. V, we compare the number of parameters and inference latency of the proposed S^3-Net with several popular baselines. Following the setting in [72], inference

TABLE II

QUANTITATIVE RESULTS ON ETH-UCY USING THE *best-of-20* STRATEGY, REPORTED AS "ADE/FDE" IN METERS. "S & T", "T & T", "T & S", AND "S & S" DENOTE SINGLE-STAGE TIME-DOMAIN, TWO-STAGE TIME-DOMAIN, TWO-STAGE SPECTRUM-DOMAIN, AND SINGLE-STAGE SPECTRUM-DOMAIN METHODS, RESPECTIVELY. LOWER IS BETTER. "TOP-2 BEST RESULTS" ON EACH SET ARE BOLDED.

Category	Model	Source	eth ↓	hotel ↓	univ ↓	zara1 ↓	zara2 ↓	Average ↓
S & T	SHENet [55]	NeurIPS'22	0.41/0.61	0.13/0.20	0.25/0.43	0.21/0.32	0.15/0.26	0.23/0.36
	MID [56]	CVPR'22	0.39/0.66	0.13/0.22	0.22/0.45	0.17/0.30	0.13/0.27	0.21/0.38
	SEEM [57]	TPAMI'23	0.62/1.20	0.61/1.21	0.50/1.04	0.31/0.61	0.36/0.68	0.48/0.95
	Flowchain [58]	ICCV'23	0.55/0.99	0.20/0.35	0.29/0.54	0.22/0.40	0.20/0.34	0.29/0.52
	Eqmotion [59]	CVPR'23	0.40/0.61	0.12/0.18	0.23/0.43	0.18/0.32	0.13/0.26	0.21/0.35
	IMP [60]	TPAMI'23	0.29/0.47	0.12/0.18	0.29/0.51	0.20/0.35	0.15/0.27	0.21/0.35
	LED [61]	CVPR'23	0.39/0.58	0.11/0.17	0.26/0.43	0.18/**0.26**	0.13/0.22	0.21/0.33
	DyHGDAT [62]	ICRA'24	0.41/0.61	0.13/0.20	0.22/0.42	0.17/0.32	0.13/0.24	0.21/0.36
	MS-TIP [63]	ICML'24	0.39/0.57	0.13/0.22	0.24/0.40	0.20/0.34	0.17/0.29	0.22/0.36
	SMEMO [64]	TPAMI'24	0.39/0.59	0.14/0.20	0.23/0.41	0.19/0.32	0.15/0.26	0.22/0.35
	LMTraj-SUP [65]	CVPR'24	0.41/0.51	0.12/0.16	0.22/**0.34**	0.20/0.32	0.17/0.27	0.22/0.32
	Dyset [66]	ECCV'24	0.32/0.46	0.14/0.21	0.24/0.45	0.17/0.28	**0.12**/0.25	0.20/0.33
	TrajCLIP [67]	NeurIPS'24	0.36/0.57	**0.10**/0.17	**0.19**/0.41	**0.16**/0.28	**0.11/0.20**	**0.18**/0.33
	AFC-RNN [68]	TPAMI'25	0.37/0.56	0.12/0.19	**0.20**/0.36	**0.16**/0.30	**0.12/0.21**	0.19/0.32
T & T	PECNet [23]	ECCV'20	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
	LB-EBM [45]	CVPR'21	0.30/0.52	0.13/0.20	0.27/0.52	0.20/0.37	0.15/0.29	0.21/0.38
	Y-net [24]	ICCV'21	0.28/**0.33**	**0.10/0.14**	0.24/0.41	0.17/**0.27**	0.13/0.22	**0.18/0.27**
	GSMNet [46]	ACCV'24	0.32/0.39	0.11/**0.13**	0.25/0.43	0.18/0.28	0.15/0.26	0.20/0.30
	MSN-SC [52]	CVPR'24	0.27/0.39	0.13/0.18	0.22/0.45	0.18/0.34	0.15/0.27	0.19/0.33
	PPT [69]	ECCV'24	0.36/0.51	0.11/0.15	0.22/0.40	0.17/0.30	**0.12/0.21**	0.20/0.31
T & S	V [26]	ECCV'22	**0.23**/0.37	0.11/0.16	0.21/**0.35**	0.19/0.30	0.14/0.24	**0.18/0.28**
S & S	S^3-Net	Ours	**0.22/0.34**	**0.10/0.14**	**0.20/0.35**	0.18/0.30	0.13/0.23	**0.17/0.27**

TABLE III

QUANTITATIVE RESULTS ON SDD WITH THE *best-of-20* STRATEGY AS "ADE/FDE" IN PIXELS. "TOP-2 BEST RESULTS" ARE BOLDED.

Category	Model	Source	SDD ↓
S & T	SHENet	NeurIPS'22	9.01/13.24
	MID	CVPR'22	7.91/14.50
	Flowchain	ICCV'23	9.93/17.17
	IMP	TPAMI'23	8.98/15.54
	DyHGDAT	ICRA'24	7.88/13.21
	SMEMO	TPAMI'24	8.11/13.06
	LMTraj-SUP	CVPR'24	7.80/**10.10**
	AFC-RNN	TPAMI'25	7.17/11.44
T & T	PECNet	ECCV'20	9.96/15.88
	LB-EBM	CVPR'21	8.87/15.61
	Y-net	ICCV'21	7.85/11.85
	GSMNet	ACCV'24	8.30/12.70
	MSN-SC	CVPR'24	7.49/12.12
	PPT	ECCV'24	**7.03**/10.65
T & S	V	ECCV'22	7.12/11.39
S & S	SpecTGNN [31]	ICRA'21	8.21/12.41
	S^3-Net	Ours	**6.83/10.54**

TABLE IV

QUANTITATIVE ABLATION STUDY ON ETH-UCY AND SDD. "BFM" IS SHORT FOR THE BILINEAR FUSION MODULE.

Model	S^3-Net w/o BFM	S^3-Net	Gain ↑
eth ↓	0.27/0.40	0.22/0.34	18.52%/15.00%
hotel ↓	0.11/0.15	0.10/0.14	9.09%/6.67%
univ ↓	0.23/0.37	0.20/0.35	13.04%/5.41%
zara1 ↓	0.22/0.36	0.18/0.30	18.18%/16.67%
zara2 ↓	0.16/0.27	0.13/0.23	18.75%/14.81%
Average ↓	0.20/0.31	0.17/0.27	15.00%/12.90%
SDD ↓	7.92/12.08	6.83/10.54	13.76%/12.75%

TABLE V

MODEL SIZE AND EFFICIENCY. A SINGLE 1080TI GPU IS USED FOR INFERENCE. "TOP 2 BEST RESULTS" ARE BOLDED.

Category	Model	Source	Params ↓	Inf Time ↓
S & T	S-GAN [17]	CVPR'18	46.3K	0.0968s
	SR-LSTM [70]	CVPR'19	64.9K	1.1789s
	STGAT [71]	ICCV'19	**44.6K**	1.3497s
	STC-Net [72]	ICCV'21	**0.7K**	**0.0013s**
	LMTraj-SUP	CVPR'24	1401M	0.1830s
	TrajCLIP	NeurIPS'24	14.9M	0.2108s
T & T	PECNet	ECCV'20	2.1M	0.6070s
T & S	V	ECCV'22	4.0M	0.0427s
S & S	S^3-Net	Ours	1.9M	**0.0201s**

time is measured on a single 1080Ti GPU for a fair comparison. The classical STC-Net was specifically designed for lightweight and low-latency prediction. Low latency requires inference time to remain below the frame sampling interval. In practice, trajectories are typically sampled at 2.5 frames per second [38], corresponding to a threshold of 0.4s per frame. Under this setting, our model also satisfies this low-latency requirement. Although not as efficient as STC-Net, S^3-Net achieves faster inference than two other methods, PECNet and V. Additionally, we test S^3-Net with different batch sizes of $\{1, 5, 10, 100\}$ and find that the inference time remains basically unchanged.

As shown in Fig. 3, we observe that S^3-Net contains a parameter count comparable to PECNet, yet delivers substantial performance gains of 37.9% in ADE on ETH-UCY. Moreover, we also test the inference cost of S^3-Net on

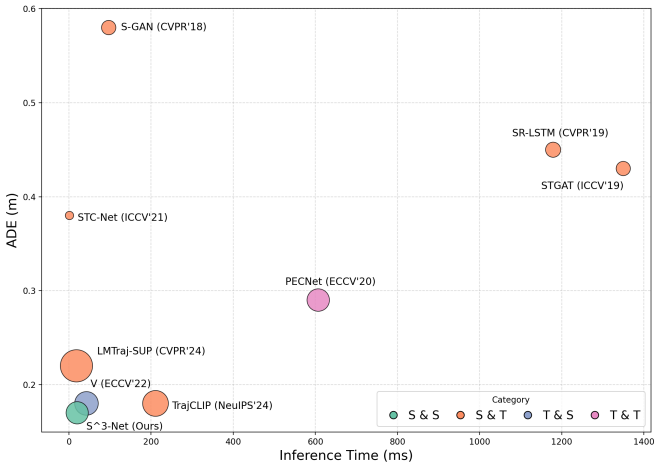

Fig. 3. Accuracy (ADE), model size, and inference time of different models on ETH-UCY. Bubble size represents the number of parameters, where larger bubbles correspond to more complex models. For fair visualization, parameter counts are log-scaled before being mapped to bubble sizes.

a CPU (3.1 GHz Intel Core i5), which is approximately 150ms, further demonstrating its suitability for low-budget devices. These results confirm that S^3-Net strikes a favorable balance among accuracy, model size, and efficiency, making it suitable for deployment in computationally constrained scenarios. Returning to the discussion of "Single-stage vs Two-stage," S^3-Net (1.9M) requires less than half the parameters of V (4.0M). In terms of inference speed, S^3-Net (0.0201s) is 2.12× faster than V (0.0427s). The above results further highlight the effectiveness and competitiveness of single-stage spectrum-domain methods.

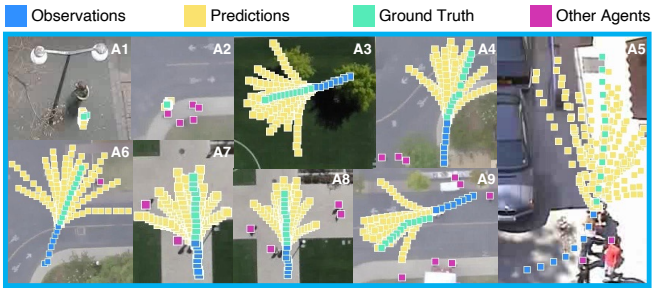

Fig. 4. Visualization of S^3-Net on ETH-UCY and SDD. Each sample illustrates 20 random predictions. Please note that observations are sometimes not visible since they are covered by predictions and ground truth.

D. Qualitative Analysis

Visualization. Fig. 4 presents qualitative results across diverse scenarios. Cases (A1) and (A2) illustrate that our model excels at predicting stationary pedestrians, regardless of the presence of nearby individuals. In (A3) and (A4), S^3-Net demonstrates strong multimodal prediction capabilities in open environments when guided solely by scene constraints. Moreover, in cases (A5)-(A9), S^3-Net effectively accounts for both social interactions and scene constraints, thereby generating multiple plausible trajectories.

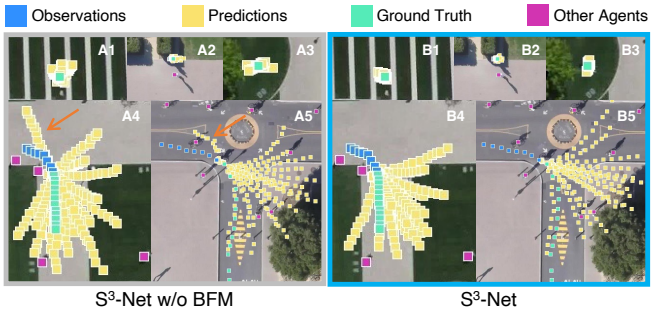

Fig. 5. Visualization of S^3-Net w/o BFM and S^3-Net in different scenarios.

Effect of the Bilinear Fusion Module. Fig. 5 compares the qualitative results of S^3-Net with S^3-Net w/o BFM. While quantitative results have already confirmed the superiority of S^3-Net, qualitative analysis provides additional insights. Two challenging scenarios are considered: predicting a stationary pedestrian and a pedestrian making a turn. (a) In (A1)-(A3) and (B1)-(B3), S^3-Net produces predictions that are closer to the ground truth. By contrast, S^3-Net w/o BFM tends to forecast short-range displacements for stationary pedestrians. This does not mean that the predictions of S^3-Net w/o BFM are entirely unreasonable, since the future motion of stationary pedestrians is inherently uncertain (*i.e.*, they may either remain still or suddenly start moving). Nevertheless, S^3-Net produces predictions that are more consistent with common sense, demonstrating its advantage in handling uncertain behaviors. (b) In (A4)-(A5) and (B4)-(B5), S^3-Net w/o BFM frequently outputs an opposite-direction trajectory (see orange arrows). Although this might be interpreted as predictive diversity, it is counterintuitive because pedestrians rarely reverse direction abruptly without external cues. Furthermore, in (B5), S^3-Net generates two plausible left-turn trajectories, whereas S^3-Net w/o BFM predicts only one. Notably, the bilinear fusion module enables S^3-Net to produce smoother and more realistic predictions, such as the left turn in (B5). These findings further demonstrate the effectiveness of the bilinear fusion module.

V. Conclusion

In this work, we propose S^3-Net, a single-stage spectrum-domain trajectory prediction network. By introducing a bilinear fusion module, S^3-Net explicitly models the interactions between low- and high-frequency components, yielding richer spectral representations. Extensive experiments on ETH-UCY and SDD demonstrate that S^3-Net achieves competitive accuracy, a smaller model size, and faster inference speed compared with baseline methods. These results highlight the practicality of frequency interaction modeling for real-time intelligent systems. In contrast to the widely studied single-stage and two-stage time-domain paradigms and two-stage spectrum-domain methods, our work advances an underexplored direction of single-stage spectrum-domain trajectory prediction. In future work, we plan to explore adaptive frequency selection and extend the framework to multi-agent collaboration and planning-oriented applications.

REFERENCES

[1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[3] B. Xia, C. Wong, Q. Peng, W. Yuan, and X. You, "Cscnet: Contextual semantic consistency network for trajectory prediction in crowded spaces," *Pattern Recognition*, p. 108552, 2022.

[4] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 244–253.

[5] T. Phong, H. Wu, C. Yu, P. Cai, S. Zheng, and D. Hsu, "What truly matters in trajectory prediction for autonomous driving?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[6] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 797–803.

[7] Z. Zhang, D. Guo, S. Zhou, J. Zhang, and Y. Lin, "Flight trajectory prediction enabled by time-frequency wavelet transform," *Nature Communications*, vol. 14, no. 1, p. 5258, 2023.

[8] Y. Chen, B. Ivanovic, and M. Pavone, "Scept: Scene-consistent, policy-based trajectory predictions for planning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 103–17 112.

[9] J. Tang, J.-F. Hu, T. Liang, X. Lin, J. Sun, W.-S. Zheng, and J. Lai, "Human motion prediction via continual prior compensation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[10] H. Ergezer and K. Leblebicioğlu, "Anomaly detection and activity perception using covariance descriptor for trajectories," in *European Conference on Computer Vision*. Springer, 2016, pp. 728–742.

[11] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hard-wired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018.

[12] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 261–268.

[13] F. Saleh, S. Aliakbarian, M. Salzmann, and S. Gould, "Artist: Autoregressive trajectory inpainting and scoring for tracking," *arXiv preprint arXiv:2004.07482*, 2020.

[14] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2891–2900.

[15] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 399–404.

[16] S. Zamboni, Z. T. Kefato, S. Girdzijauskas, C. Norén, and L. Dal Col, "Pedestrian trajectory prediction with convolutional neural networks," *Pattern Recognition*, vol. 121, p. 108252, 2022.

[17] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[18] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "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*, 2019, pp. 1349–1358.

[19] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 137–146.

[20] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *European Conference on Computer Vision*. Springer, 2020, pp. 507–523.

[21] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9813–9823.

[22] S. Lee, J. Lee, Y. Yu, T. Kim, and K. Lee, "Mart: Multiscale relational transformer networks for multi-agent trajectory prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 89–107.

[23] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *European Conference on Computer Vision*, 2020, pp. 759–776.

[24] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.

[25] C. Wong, B. Xia, Q. Peng, W. Yuan, and X. You, "Msn: multi-style network for trajectory prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 9751 – 9766, 2023.

[26] C. Wong, B. Xia, Z. Hong, Q. Peng, W. Yuan, Q. Cao, Y. Yang, and X. You, "View vertically: A hierarchical network for trajectory prediction via fourier spectrums," in *European Conference on Computer Vision*. Springer, 2022, pp. 682–700.

[27] B. Xia, C. Wong, D. Xu, Q. Peng, and X. You, "Another vertical view: A hierarchical network for heterogeneous trajectory prediction via spectrums," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[28] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9489–9497.

[29] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–489.

[30] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 766–17 778, 2020.

[31] D. Cao, J. Li, H. Ma, and M. Tomizuka, "Spectral temporal graph neural network for trajectory prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1839–1845.

[32] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.

[33] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 317–326.

[34] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 402–419.

[35] D. Guo, C. Xu, and D. Tao, "Bilinear graph networks for visual question answering," *IEEE Transactions on neural networks and learning systems*, 2021.

[36] Q. Xu, Y. Mei, J. Liu, and C. Li, "Multimodal cross-layer bilinear pooling for rgbt tracking," *IEEE Transactions on Multimedia*, vol. 24, pp. 567–580, 2021.

[37] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.

[38] P. Zhang, J. Xue, P. Zhang, N. Zheng, and W. Ouyang, "Social-aware pedestrian trajectory prediction via states refinement lstm," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 5, pp. 2742–2759, 2022.

[39] L. Rossi, M. Paolanti, R. Pierdicca, and E. Frontoni, "Human trajectory prediction and generation using lstm models and gans," *Pattern Recognition*, vol. 120, p. 108136, 2021.

[40] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.

[41] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," pp. 10 335–10 342, 2021.

[42] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7386–7400, 2021.

[43] W. Xiang, Y. Haoteng, H. Wang, and X. Jin, "Socialcvae: predicting pedestrian trajectory via interaction conditioned latents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 6, 2024, pp. 6216–6224.

[44] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[45] B. Pang, T. Zhao, X. Xie, and Y. N. Wu, "Trajectory prediction with latent belief energy-based model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 814–11 824.

[46] S. Liu, Y. Wang, Y. Zhu, P. Yao, T. Mao, and Z. Wang, "Gsmnet: Towards long-term trajectory prediction by integrating multi-scale information," in *Proceedings of the Asian Conference on Computer Vision*, 2024, pp. 2954–2969.

[47] Y. Li, N. Wang, J. Liu, and X. Hou, "Factorized bilinear models for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2079–2087.

[48] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1309–1322, 2017.

[49] J.-F. Hu, W.-S. Zheng, J. Pan, J. Lai, and J. Zhang, "Deep bilinear learning for rgb-d action recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 335–351.

[50] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1821–1830.

[51] J. Yue, D. Manocha, and H. Wang, "Human trajectory prediction via neural social physics," in *European Conference on Computer Vision*. Springer, 2022, pp. 376–394.

[52] C. Wong, B. Xia, Z. Zou, Y. Wang, and X. You, "Socialcircle: Learning the angle-based social interaction representation for pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 005–19 015.

[53] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision*. Springer, 2016, pp. 549–565.

[54] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi, "Unitraj: A unified framework for scalable vehicle trajectory prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 106–123.

[55] M. Meng, Z. Wu, T. Chen, X. Cai, X. Zhou, F. Yang, and D. Shen, "Forecasting human trajectory from scene history," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 920–24 933, 2022.

[56] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 113–17 122.

[57] D. Wang, H. Liu, N. Wang, Y. Wang, H. Wang, and S. Mcloone, "Seem: a sequence entropy energy-based model for pedestrian trajectory all-then-one prediction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 1070–1086, 2023.

[58] T. Maeda and N. Ukita, "Fast inference and update of probabilistic density estimation on trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9795–9805.

[59] C. Xu, R. T. Tan, Y. Tan, S. Chen, Y. G. Wang, X. Wang, and Y. Wang, "Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1410–1420.

[60] L. Shi, L. Wang, C. Long, S. Zhou, W. Tang, N. Zheng, and G. Hua, "Representing multimodal behaviors with mean location for pedestrian trajectory prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[61] W. Mao, C. Xu, Q. Zhu, S. Chen, and Y. Wang, "Leapfrog diffusion model for stochastic trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5517–5526.

[62] H. Guo, Y. Peng, Z. Fan, H. Zhu, and X. Song, "Hhgnn: heterogeneous hypergraph neural network for traffic agents trajectory prediction in grouping scenarios," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 101–14 108.

[63] P. S. Chib, A. Nath, P. Kabra, I. Gupta, and P. Singh, "Ms-tip: Imputation aware pedestrian trajectory prediction," in *International Conference on Machine Learning*. PMLR, 2024, pp. 8389–8402.

[64] F. Marchetti, F. Becattini, L. Seidenari, and A. Del Bimbo, "Smemo: social memory for trajectory forecasting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[65] I. Bae, J. Lee, and H.-G. Jeon, "Can language beat numerical regression? language-based multimodal trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 753–766.

[66] M. Pourkeshavarz, J. Zhang, and A. Rasouli, "Dyset: A dynamic masked self-distillation approach for robust trajectory prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 324–342.

[67] P. Yao, Y. Zhu, H. Bi, T. Mao, and Z. Wang, "Trajclip: Pedestrian trajectory prediction method using contrastive learning and idempotent networks," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[68] Y. Dong, L. Wang, S. Zhou, W. Tang, G. Hua, and C. Sun, "Afc-rnn: Adaptive forgetting-controlled recurrent neural network for pedestrian trajectory prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[69] X. Lin, T. Liang, J. Lai, and J.-F. Hu, "Progressive pretext task learning for human trajectory prediction," in *European Conference on Computer Vision*. Springer, 2024, pp. 197–214.

[70] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 085–12 094.

[71] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6272–6281.

[72] S. Li, Y. Zhou, J. Yi, and J. Gall, "Spatial-temporal consistency network for low-latency trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 1940–1949.