TrajFine: Predicted Trajectory Refinement for Pedestrian Trajectory Forecasting

Kuan-Lin Wang[1], Li-Wu Tsao[1], Jhih-Ciang Wu[2], Hong-Han Shuai[1], and Wen-Huang Cheng[2]
[1]National Yang Ming Chiao Tung University, Taiwan
[2]National Taiwan University, Taiwan

Abstract

Trajectory prediction, aiming to forecast future trajectories based on past ones, encounters two pivotal issues: insufficient interactions *and* scene incompetence. *The former signifies a lack of consideration for the interactions of predicted future trajectories among agents, resulting in a potential collision, while the latter indicates the incapacity for learning complex social interactions from simple data. To establish an interaction-aware approach, we propose a diffusion-based model named* TrajFine *to extract social relationships among agents and refine predictions by considering past predictions and future interactive dynamics. Additionally, we introduce Scene Mixup to facilitate the augmentation via integrating agents from distinct scenes under the Curriculum Learning strategy, progressively increasing the task difficulty during training. Extensive experiments demonstrate the effectiveness of TrajFine for trajectory forecasting by outperforming current SOTAs with significant improvements on the benchmarks.*

1. Introduction

Pedestrian trajectory prediction, the task of forecasting the future paths of pedestrians based on their past behaviors, has become increasingly pivotal. This surge in importance is attributed to the rapid advancements in autonomous vehicles [24], human-robot interaction systems [7], smart city planning [17], and even video surveillance systems [11]. Considering complex interactions among individuals, such as co-navigation, collision avoidance, and hesitation, adds intricacy to the task, especially when dealing with dynamic real-world scenarios. These numerous interactions significantly contribute to the complexity of the prediction task.

Over the years, abundant methodologies have emerged to address pedestrian trajectory prediction challenges by adopting generative techniques such as GAN [2, 13, 35] and CVAE [45]. More recently, with the emergence of diffusion models, remarkable achievements have been made in the generation tasks of computer vision [15, 38]. The quality, diversity, and controllable nature of outcomes generated by diffusion models have paved the way for various applications. For trajectory generation, a novel approach [33] incorporated controllable guidance factors such as goals, avoidance, and social groups into the diffusion model, leading to the generation of trajectory data that rivals real-world scenarios while maintaining rationality. Moreover, MID [12] leveraged social-temporal information from past trajectories as conditioning factors to enable the diffusion model to predict a range of diverse future trajectories.

While the diffusion-based model exhibits impressive performance in trajectory forecasting, two overlooked issues, which we termed as *insufficient interactions* and *scene*

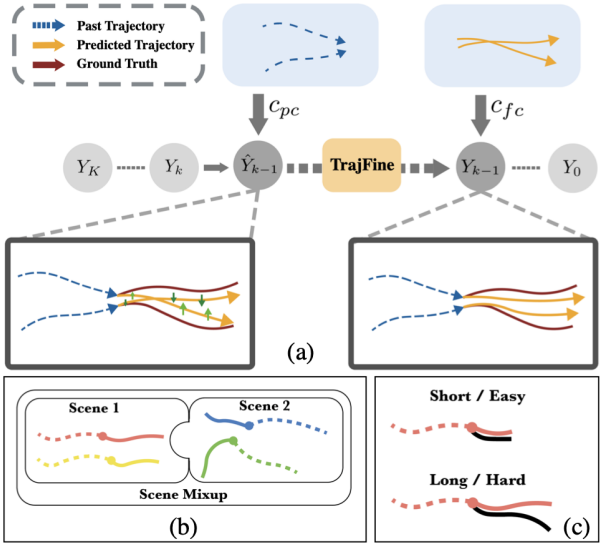

Figure 1. Illustration of main concepts in TrajFine. (a) General diffusion models generate the denoised future trajectory $\hat{Y}_{k-1}$ from Y_k, which consider the condition of past context c_{pc} only. In contrast, our refinement process further conditioned on the denoised future context c_{fc}, which allows for appropriate consideration of future interactions to generate more reasonable Y_{k-1}. (b) Scene Mixup augments additional social agents across wide range of scenarios, which learns the capability to navigate through various social situations while interacting with others. (c) Curriculum Learning presents an easy-to-hard strategy on a variety of lengths.

incompetence, warrant further exploration. For instance, MID allows the model to predict future trajectories according to past trajectories. It operates only for single-agent prediction, where the path of each agent is predicted separately. This setup misses the simultaneous consideration of the rationality of multiple agents' prediction. Thus, it could lead to predictions that deviate from plausible real-world scenarios, *e.g.*, collision. Besides, the quality of trajectories generated by the diffusion model primarily depends on the diversity and number of training samples, which are bottlenecks while using existing *incomplete* benchmarks.

Regarding insufficient interactions, the most intuitive solution is to enable the model to interact in the feature space, exchanging information about the future trajectories of all agents within the same scenario. This interaction would subsequently facilitate the concurrent prediction of their future paths. However, implementing this "Parallel" approach, *i.e.*, simultaneously handling future trajectory prediction and extracting future interaction, is challenging within the sole diffusion mode. The diffusion model is designed to learn the objective of the appropriate amount of noise to be denoised from the current noisy trajectory during each diffusion step. Hence, these features represent noise in the feature space, meaning they lack any information related to trajectories. Therefore, if we direct interaction and simultaneous prediction of these features from various agents, the model would encounter difficulties while effectively utilizing the information from interactions.

In this paper, in contrast to the intuitive parallel concept mentioned earlier, we introduce a novel "Series" approach, *i.e.*, the process involves initially generating preliminary predictions based on past trajectories and refining these predictions by allowing them to interact and extracting features from each other. As shown in Figure 1, our formulation mainly focuses on tackling the mentioned issues in designing a compelling trajectory refine module. In detail, we proposed *TrajFine*, which first denoises the future trajectories within the same scenario and then extracts interactive relationship features between trajectories. Subsequently, the refined trajectories are made using these comprehended features that capture the future trajectories of all agents. This strategy enables the model to generate predictions that also consider the future trajectories of other agents and yield rational outcomes. To mitigate the scene incompetence issue, we propose the Scene Mixup, which involves merging agents from diverse scenes, thereby augmenting data into more complicated scenarios. The model attains enhanced robustness through this augmentation process, enabling it to outperform across monotonous and complex environments. With such augmentative scenes, we adopt the Curriculum Learning (CL) strategy to progressively increase the lengths of trajectories during the training process, moving from simple to complicated tasks. In summary, the main contributions of this paper are as follows.

- We introduced a novel framework for refining predicted trajectories. By extracting future trajectory relationships, TrajFine refines predictions more reasonably by considering the interactions among agents while achieving SOTA performance to validate effectiveness.
- We proposed an augmentation technique named Scene Mixup, which combines distinct scenes to facilitate the model's understanding of varied social relationships.
- We utilized Curriculum Learning to progressively predict trajectories from short to long lengths, resulting in enhanced outcomes of the refinement process.

2. Related Work

Trajectory Prediction. The dynamics of social forces are centered on the internal motivations of agents, dictating specific motion actions for pedestrians [14]. These forces encompass various dynamics, including maintaining interpersonal distances, attraction, and repulsion between individuals. Consequently, numerous studies [1, 2, 4, 13, 35, 42] concentrate on modeling social forces. Recent efforts explore multiple plausible avenues from a stochastic perspective, acknowledging the inherent uncertainty and augmenting diversity in pedestrian trajectory prediction. For instance, studies like [1, 4] integrate social dynamics inherent in historical trajectories using Long Short-Term Memory (LSTM) models to predict and researches in [16, 19, 44], employ GNN-based models to connect relationships among different agents. Another category involves GAN-based [2, 10, 13, 19] techniques to strike a delicate balance between diversity and authenticity in trajectory prediction. Additionally, [45] introduced Transformers with Conditional Variational Autoencoders (CVAE) [5, 21, 43] to capture long-range dependencies within trajectories. These diverse approaches aim to improve the fidelity and diversity of predicted trajectories by considering various aspects of social dynamics.

Based on the remarkable effects of diffusion models, recent research explores the potential and utilizes the diffusion process to generate diverse predictions covering plausible trajectories. For example, the Leapfrog [27] employs an initializer to plan trajectories based on potential destinations and variance, allowing some steps to be skipped during denoising. In the TRACE [33], controllable guidance is incorporated to enable the diffusion model to generate trajectories to real-world scenarios and possess rationality. Additionally, MID [12] employs spatial-temporal information from past trajectories as a condition, allowing the diffusion model to predict future trajectories based on the prior.

In contrast to previous works, the primary objective for TrajFine is to enhance the learned predictive distribution of the model by extracting social-temporal interactions among predicted trajectories. This refinement process concentrates

the predictive distribution more accurately around actual trajectories while preventing overlap with other agents.

Curriculum Learning. Motivated by mimicking the learning procedures like humans, CL designs a progressive flow that is typically simple to complex. This learning style has been demonstrated in various tasks to accelerate convergence and discover superior local optima. The CL can be carried out from distinct standpoints, such as data, task, and model. From the data perspective, the early work [3] involves predefined difficulty levels according to samples, while Spitkovsky *et al*. [40] formulates the various tasks by predicting different sequence lengths. Subsequent works [20, 22] focus on assessing sample difficulty by iteratively evaluating model performance during training. In addition to facilitating the model's progression from easy to hard samples, the studies [39, 47] emphasize the need to ensure diversity among the samples, employing additional constraints to achieve balance. From the model view, Curriculum Dropout [29] employs a time-based schedule to determine the probability of retaining neurons in the network, where the approach effectively gradually increases the difficulty for optimization. Another investigation [18] tackles model capacity by progressively adding layers to the generator and discriminator during training, intensifying the adversarial learning process between them.

We employed CL at the data and task perspectives for our approach. More precisely, we treated the instances subjected to Scene Mixup augmentation as hard examples. We initially focused on the original samples during training and then gradually increased the augmented sample ratio. From the task aspect, the model also transits different level tasks by learning to predict trajectories from short to long.

3. Method

The overall architecture is illustrated in Figure 2. We first elaborate on the notations and then focus on the proposed *TrajFine*. Furthermore, we present the innovative Scene Mixup by augmenting additional social agents within the scene, to enhance social diversity. Finally, we incorporate the easy-to-hard learning pipeline, *i.e*., the CL framework, which gradually captures the importance of refinement.

3.1. Preliminary

We consider a scene containing multiple agents for data processing. The past trajectories are denoted by $X = (x_1^{-p}, \cdots, x_1^0, \cdots, x_N^{-p}, ..., x_N^0)$, where p and N stand for past timestamp and number of agents, respectively. The future trajectories are denoted by $Y_0 = (y_1^1, \cdots, y_1^f, \cdots, y_N^1, ..., y_N^f)$, where f represents future timestamp. Since our framework is based on the diffusion model, it is also essential to define the notations in our diffusion and denoise process.

Diffusion Process. To prevent any confusion between trajectory timestamps and diffusion timestamps, we represent the diffusion process as $(Y_0, Y_1, ..., Y_K)$, with K being the maximum diffusion timestamps. In a single step of diffusion process, the procedure can be formulated as:

$$q(Y_k|Y_{k-1}) = N(Y_k; \sqrt{1-\beta_k}Y_{k-1}, \beta_k\mathbf{I}), \quad (1)$$

where β_1,...,β_K is the predefined variance for separated timestamps. While considering multiple diffusion steps from the original trajectory Y_0 to the sampling of Y_k, the definition turned into another form:

$$q(Y_k|Y_0) = N(Y_k; \sqrt{\bar{\alpha}_k}Y_0, (1-\bar{\alpha}_k)\mathbf{I}), \quad (2)$$

where $\alpha_k = 1 - \beta_k$ and $\bar{\alpha}_k = \prod_{s=1}^{k} \alpha_s$. To enable the fast sampling speed of noisy data Y_k, the noise ϵ is linearly combined with the clean trajectory Y_0, represented as:

$$Y_k = \sqrt{\bar{\alpha}_k}Y_0 + \sqrt{1-\bar{\alpha}_k}\epsilon. \quad (3)$$

Denoise Process. Based on conditional diffusion model [9, 15], we can sample trajectories from the condition of past context c_{pc} as shown in (4). The transition of Gaussian process $p_\theta(\hat{Y}_{k-1}|Y_k, c_p)$ formulates a step-by-step prediction to get closer to the ground truth trajectory, which is constructed by learning the mean μ_θ with the predefined variance σ_k for each denoising timestamp in the model.

$$\begin{aligned} p_\theta(Y_0|c_{pc}) &= \int p_\theta(Y_{0:K}|c_{pc})dY_{1:K}. \\ p_\theta(Y_{0:K}|c_{pc}) &= p(Y_K)\prod_{k=1}^{K} p_\theta(\hat{Y}_{k-1}|Y_k, c_{pc}). \end{aligned} \quad (4)$$

$$p_\theta(\hat{Y}_{k-1}|Y_k, c_{pc}) = N(\hat{Y}_{k-1}; \mu_\theta(Y_k, k, c_{pc}), \sigma_k^2\mathbf{I}). \quad (5)$$

Specifically, the decomposition of μ_θ is a linear combination between Y_k and the noise approximator ϵ_θ. Thus, the Gaussian formulation of $\hat{Y}_{k-1}$ can be represented as:

$$\hat{Y}_{k-1} = \frac{1}{\sqrt{\alpha_k}}(Y_k - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}}\epsilon_\theta(Y_k, k, c_{pc})) + \sigma_k\mathbf{z}, \quad (6)$$

where $\mathbf{z}$ implies the stochastic sampling from the normal distribution $\mathcal{N}(0, \mathbf{I})$.

3.2. Training

This section discusses estimating coarse prediction by intra-trajectory noise ϵ_θ and refining future sample dependency on social aspects by inter-trajectory noise ϵ_ϕ. We split the discussion into estimation phase and refinement phase.

Estimation Phase. The main consideration in this phase is to realize the spatial-temporal information of each agent. To reach this goal, we employ the past context encoder $\mathcal{F}_{pc}$ [45]. This encoder leverages attention mechanisms of

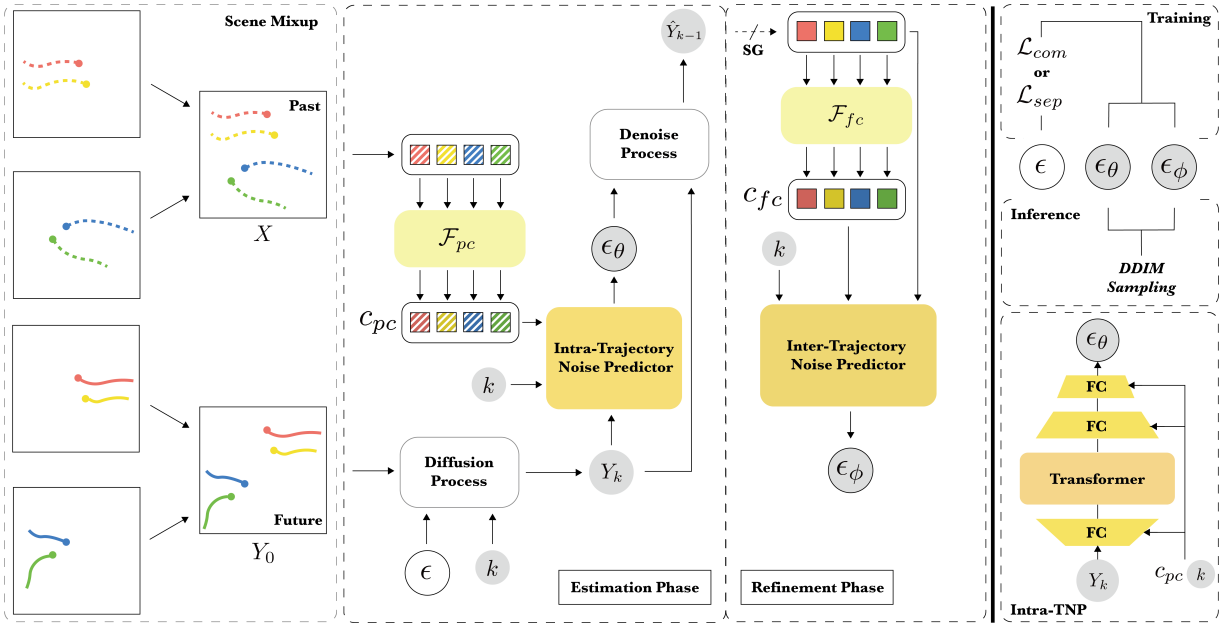

Figure 2. Overview of our TrajFine framework and Scene Mixup. In Estimation Phase, the past context encoder $\mathcal{F}_{pc}$ extracts the feature c_{pc} according to the past trajectories X. The feature then combines with the noisy trajectory Y_k at diffusion timestamp k, which is constructed by adding the sampled noise ϵ to the future trajectory Y_0. The prediction of ϵ_θ by Intra-Trajectory Noise Predictor (Intra-TNP) represents the extraction of spatial-temporal information, which is used to generate the coarse prediction $\hat{Y}_{k-1}$ by denoise process. Subsequently, in Refinement Phase, the future context encoder $\mathcal{F}_{fc}$ extracts the feature c_{fc} according to the predicted trajectory $\hat{Y}_{k-1}$. Since the prediction of each agent in $\hat{Y}_{k-1}$ are sampled independently, the overall target of ϵ_ϕ is to enable further refinement on the future interactions, which is generated by the Inter-Trajectory Noise Predictor (Inter-TNP). From Estimation Phase to Refinement Phase, we use the abbreviation SG to denote the stop gradient. The use of SG can reduce the interference of these two independent targets, ϵ_θ and ϵ_ϕ.

transformer to analyze interactions among agents and across various timestamps within the scene, enabling us to extract past information from X, represented as:

$$c_{pc} = \mathcal{F}_{pc}(X). \tag{7}$$

This extracted information c_{pc} is then utilized as the condition for our intra-trajectory noise predictor (Intra-TNP), which is composed of fully connected layers for dimension adjustment and a transformer encoder to realize the future behaviors from different timestamps. However, if we directly utilize (8) as the training objective, the predicted future trajectories $\hat{Y}_{k-1}$ may implicitly include outcomes that do not adhere to social forces, due to the independence of stochastic sampling strategy in estimation phase.

$$\mathcal{L}_{intra} = \mathbb{E}_{\epsilon,Y_0,k}\|\epsilon - \epsilon_\theta(Y_k, k, c_{pc})\|. \tag{8}$$

To address this issue, we introduce TrajFine to consider the social relationships on the predicted trajectories among different agents. Thus, constructing inter-trajectory relationships through the design of our refinement phase.

Refinement Phase. Different from the estimation phase that connects the understanding from past to future, the main target in this phase only focuses on the mutual information among future agent trajectories. We adopt the future context encoder $\mathcal{F}_{fc}$ [45] on the coarse prediction $\hat{Y}_{k-1}$ to extract c_{fc} as the condition of social understanding, *i.e.*,

$$c_{fc} = \mathcal{F}_{fc}(\hat{Y}_{k-1}). \tag{9}$$

Algorithm 2 endeavors to alleviate the imprecise noise $(\epsilon - \epsilon_\theta)$ in predictions through conditioning on c_{fc}. The inter-trajectory noise predictor (Inter-TNP) is similar to Intra-TNP in model structure. However, its weights are distinct from those of Intra-TNP. Leveraging c_{fc} and the pre-

Algorithm 1 Estimation Phase Training

Input: X, Y_0
Output: ϵ_θ, $\hat{Y}_{k-1}$, $\mathcal{L}_{intra}$

1: $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
2: $k \sim \text{Uniform}(1, ..., K)$
3: $Y_k \leftarrow$ add noise on Y_0 by ϵ and k ▷ Eq. 3
4: $c_{pc} \leftarrow \mathcal{F}_{pc}(X)$ extract past context ▷ Eq. 7
5: $\epsilon_\theta \leftarrow$ Intra-TNP(Y_k, k, c_{pc})
6: Sample $\hat{Y}_{k-1}$ by Y_k and ϵ_θ ▷ Eq. 6
7: Calculate loss $\mathcal{L}_{intra}(\epsilon, \epsilon_\theta)$ ▷ Eq. 8

Algorithm 2 Refinement Phase Training

Input: $\hat{Y}_{k-1}$
Output: ϵ_ϕ, $\mathcal{L}_{inter}$
1: $c_{fc} \leftarrow \mathcal{F}_{pc}(\hat{Y}_{k-1})$ extract future context ▷ Eq. 9
2: $\epsilon_\phi \leftarrow$ Inter-TNP($\hat{Y}_{k-1}$, k, c_{fc})
3: Calculate loss $\mathcal{L}_{inter}(\epsilon, \epsilon_\theta, \epsilon_\phi)$ ▷ Eq. 10

dicted trajectory $\hat{Y}_{k-1}$, Inter-TNP generates ϵ_ϕ by encapsulating social information from future trajectories of different agents. During this phase, our compensatory training objective can be expressed as follows:

$$\mathcal{L}_{inter} = \mathbb{E}_{\epsilon,Y_0,k}\|[\epsilon - \epsilon_\theta(Y_k, k, c_{pc})] - \epsilon_\phi(\hat{Y}_{k-1}, k, c_{fc})\|. \tag{10}$$

As a result, our separated loss can be expressed as:

$$\mathcal{L}_{sep} = \mathcal{L}_{intra} + \mathcal{L}_{inter}. \tag{11}$$

To better match our inference, we reformulate $\mathcal{L}_{sep}$ by omitting $\mathcal{L}_{intra}$ and define the combined loss as follows:

$$\mathcal{L}_{com} = \mathbb{E}_{\epsilon,Y_0,k}\|\epsilon - [\epsilon_\theta(Y_k, k, c_{pc}) + \epsilon_\phi(\hat{Y}_{k-1}, k, c_{fc})]\|, \tag{12}$$

which is the loss we used as $\mathcal{L}_{inter}$. Comparing the differences between $\mathcal{L}_{sep}$ and $\mathcal{L}_{com}$, the simple social cases can better fit $\mathcal{L}_{sep}$, which put more weights on the intra-trajectory noise ϵ_θ. In contrast, complex scenarios are closely related to real-world situations, and thus $\mathcal{L}_{com}$ is a balanced way to put equal emphasis on ϵ_θ and ϵ_ψ.

3.3. Inference

During inference, the overall noise $\hat{\epsilon}$ incorporates both historical and forthcoming social information insights, which is derived from ϵ_θ and ϵ_ϕ as:

$$\hat{\epsilon} = \epsilon_\theta + \epsilon_\phi. \tag{13}$$

The original approach to DDPM [15] sampling proves excessively protracted and time-consuming. Based on the experimental study in DDIM [38], excluding the stochastic factor σ_τ can contribute to enhancements when the number of sampled timestamps is small. Thus, we adopt the DDIM [38] method to omit the stochastic mechanism in the denoising process, which implies that the process becomes deterministic once Y_K is established. By doing this, the number of diffusion timestamps can be decreased. We have condensed the timestamp schedule resulting from predefined reduction into the shorten timestamp τ, represented by the following equation:

$$Y_{\tau-1} = \sqrt{\alpha_{\tau-1}}\left(\frac{Y_\tau - \sqrt{1-\alpha_\tau}\hat{\epsilon}}{\sqrt{\alpha_\tau}}\right) + \sqrt{1-\alpha_{\tau-1}}\hat{\epsilon}. \tag{14}$$

3.4. Scene Mixup

Since TrajFine generates the final results by extracting social information from both past observed trajectories and future predicted trajectories. Due to the scarcity of agents in some scenarios, which poses challenges for the model in learning complex social interactions, we propose the application of Scene Mixup, which is elaborated upon in Figure 2. Mixup [46, 48] is a data augmentation technique that enhances a model's representational capabilities, thus improving generalization. This concept has been extended to various applications in several previous works [28, 30]. While a straightforward approach might entail randomly selecting agents from different scenes for blending, such a method has the potential to disrupt the intrinsic social dynamics within each scene. To tackle this challenge, we propose the approach of combining agents from just two entire scenes. By doing so, we not only involve more intricate social relationships inherent in both scenes but also preserve authenticity by decreasing the likelihood of unrealistic social situations when augmenting more scenes. This concept is formulated as follows:

$$\begin{aligned} \tilde{X} &= X^i \oplus X^j, \\ \tilde{Y}_0 &= Y_0^i \oplus Y_0^j, \end{aligned} \tag{15}$$

$\forall\, i \neq j$, where i, j represent different scenes. Operation $\oplus$ denotes concatenation of past trajectories and ground truth from scene i, j. After Scene Mixup, we will get augmented past trajectories $\tilde{X}$ and ground truth of new scene $\tilde{Y}_0$.

3.5. Curriculum Learning

Due to the potentially long training process and the need for diverse data in training the diffusion model, we employ the concept of Curriculum Learning for two aspects: predicted length and data. It is intuitive that as the model predicts longer future trajectories, the difficulty of task increases. This phenomenon has been demonstrated in experiments conducted by Gupta *et al.* [13], where the average displacement error (ADE) tends to grow as the prediction length extends. Based on these observations, we introduce the concept of length-aware curriculum learning as training objective, raising the task complexity during training by gradually extending the predicted length of future trajectories. We modify the equation (12) into following:

$$\mathcal{L}^{'}_{com} = \mathbb{E}_{\epsilon,Y_0,k}\|\sum_{t=1}^{\tilde{f}} \epsilon^t - [\epsilon^t_\theta(Y_k, k, c_{pc}) + \epsilon^t_\phi(\hat{Y}_{k-1}, k, c_{fc})]\|, \tag{16}$$

where $\tilde{f}$ represents the number of length-aware future timestamps, gradually increasing towards its maximum value f as determined by the training process. This training strategy can lead to faster and more stable convergence of the model. Furthermore, according to Choi *et al.* [8],

		ADE / FDE ↓, Best of $S = 20$ samples					
Method	Venue	**ETH**	**HOTEL**	**UNIV**	**ZARA1**	**ZARA2**	**AVG**
Social-GAN [13]	CVPR 2018	0.81 / 1.52	0.72 / 1.61	0.60 / 1.26	0.34 / 0.69	0.42 / 0.84	0.58 / 1.18
PECNet [25]	ECCV 2020	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
Trajectron++ [36]	ECCV 2020	0.61 / 1.02	0.19 / 0.28	0.30 / 0.54	0.24 / 0.42	0.18 / 0.32	0.30 / 0.51
AgentFormer [45]	ICCV 2021	0.45 / 0.75	0.14 / 0.22	0.25 / 0.45	0.18 / **0.30**	0.14 / **0.24**	0.23 / 0.39
MID [12]	CVPR 2022	0.39 / 0.66	0.13 / 0.22	0.22 / 0.45	0.17 / **0.30**	0.13 / 0.27	0.21 / 0.38
BOsampler [6]	CVPR 2023	0.52 / 0.95	0.19 / 0.39	0.30 / 0.67	**0.14** / 0.33	0.20 / 0.45	0.27 / 0.56
TrajFine (Ours) with $\mathcal{L}_{sep}$		**0.34 / 0.58**	**0.11** / 0.19	**0.21 / 0.44**	0.17 / 0.35	**0.12** / 0.27	**0.19** / 0.37
TrajFine (Ours) with $\mathcal{L}_{com}$		0.35 / 0.60	**0.11 / 0.18**	0.22 / 0.48	0.15 / **0.30**	**0.12** / 0.25	**0.19 / 0.36**

Table 1. Comparisons on the ETH-UCY dataset with Best of 20 ADE / FDE ↓. The boldface score and underline denote the best and second-best results, respectively.

the samples resulting from Scene Mixup are considered to be more challenging. By employing data-aware curriculum learning, we continuously enhance our training process by gradually increasing the number of Scene Mixup samples. This approach involves the model first learning basic social relationships (original samples) before advancing to more complex social relationships (Scene Mixup samples).

4. Experiment

We present the evaluation datasets, shedding light on their inherent properties. The evaluation metrics commonly used in trajectory prediction are introduced afterward. Later, the comparative analyses with other SOTAs are conducted and discussed. Finally, we visualize results and investigate ablation studies that are explicitly tailored to evaluate the effectiveness of our proposed method.

4.1. Experimental Setup

Datasets. We evaluate our method on two public datasets. The widely recognized ETH-UCY dataset [23, 32] comprises five subsets: ETH, HOTEL, UNIV, ZARA1, and ZARA2, where the first two subsets are from ETH and the remaining belong to UCY. Those scenes encapsulate various levels of complexity for social relationships, such as the splits from ETH, which present simplistic and sparse scenarios with limited numbers of agents, while the UCY parts contain more intricate social relationships and higher agent density. Besides, the Stanford Drone Dataset (SDD) [34] dataset has been recognized as a well-established benchmark. It captures agent trajectories in diverse scenes via drones, offering complicated social relationships insights.

Evaluation Metrics. We follow the prevalent trajectory prediction metrics: Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE computes the mean difference between all the ground truth positions and all the predicted positions throughout the trajectories. On the other hand, FDE quantifies the displacement between the final points of the actual and predicted trajectories, essentially measuring the distance to the final destination.

Evaluation Protocol. The tendency of recent works has shifted towards predicting stochastic trajectories rather than deterministic ones to enrich the diversity and capability of predicting plausible areas. In practice, the trajectory predictor generates S samples and chooses the best one for evaluation. We adopt the leave-one-out strategy to evaluate the ETH-UCY dataset, meaning the model is evaluated on one subset after being trained on the remaining. This strategy also indicates the generalization capabilities in the context of domain adaptation. Our approach involves operating the observed 8 timestamps as input and forecasting the subsequent 12 timestamps as the prediction for all benchmarks. In addition, within the context of the stochastic strategy, we follow previous works by selecting $S = 20$.

Implementation Details. We use a backbone similar to AgentFormer [45] for past/future context encoders to extract c_{pc} and c_{fc}, respectively. The diffusion process is referred to MID [12]. We employ the Adam optimizer with a learning rate 0.0001 and a batch size of 32 for 150 epochs. All phases transfer the 2D positions into 512-dimensional embeddings for learning by transformer. We employ diffusion timestamps $K = 100$ during training, while the skip in timestamps is a trade-off between acceleration and performance during inference, as shown in Table 4.

4.2. Baseline Comparison

In trajectory prediction task, aside from using trajectory information, certain methodologies also utilize map images to enhance performance by comprehending environmental elements such as obstacles and pedestrian pathways. However, in our evaluation, we primarily focus on using trajectory as the only information for fair comparison.

On the ETH-UCY dataset, we compare our approach

with several state-of-the-art (SOTA) methods [12, 13, 25, 36, 45]. In trajectory sampler based method, we compare with BOsampler [6], which makes further improvement on Trajectron++ [36]. As illustrated in Table 1, we compare (11) and (12) mentioned in Section 3.2 with other baseline methods. Particularly on the ETH and HOTEL sub-datasets, our method effectively captures rich social interactions from other sub-datasets, leading to apparent improvements. Furthermore, our Scene Mixup augments the data with enriched social information, enabling our performance on the UNIV, ZARA1, and ZARA2 sub-datasets comparable to other approaches. In UNIV, $\mathcal{L}_{sep}$ performs well by leveraging $\mathcal{L}_{intra}$ effectively for learning from limited and simple data. In contrast, in ZARA1 and ZARA2, where both training and testing data are complex and abundant, $\mathcal{L}_{com}$ exhibits better performance, highlighting the need for refinement functionalities on complex scenarios. Furthermore, due to the complexity of the SDD dataset in Section 4.3, we select $\mathcal{L}_{com}$ for our analysis.

Method	Venue	**ADE**	**FDE**
PECNet [25]	ECCV 2020	9.96	15.88
Trajectron++† [36]	ECCV 2020	8.98	19.02
Expert† [49]	ICCV 2021	10.67	14.38
LB-EBM [31]	CVPR 2021	8.87	15.61
PCCSNET [41]	ICCV 2021	8.62	16.16
MID [12]	CVPR 2022	7.61	14.30
TUTR [37]	ICCV 2023	7.76	**12.69**
TrajFine (Ours) with $\mathcal{L}_{sep}$		<u>7.22</u>	13.79
TrajFine (Ours) with $\mathcal{L}_{com}$		**7.11**	<u>13.28</u>

Table 2. Comparisons on the SDD dataset with Best of 20 ADE / FDE. † means the results are reproduced by Gu *et al*. [12].

Compared to the ETH-UCY dataset, the SDD dataset comprises a greater variety of scenes, which in turn implies a more complex and diverse range of social relationships. We have compared our results with other SOTA [12, 25, 31, 36, 37, 41, 49], as illustrated in Table 2. Correspondingly, our method outperforms other baseline approaches in ADE and demonstrates comparable performance in FDE. Compare to the recent work TUTR [37], which design general motion modes to enhance the diversity of model predictions, making it more likely for the destination to approximate the ground truth. However, this approach needs to select proper hyperparameter on the number of motion modes, which is sensitive to the final results. Thus, TUTR presents the performance trade-off in ADE and FDE, which is improper to select the best number of mode for the actual use case on different scenarios. Compare to the baseline method MID [12] using diffusion model as backbone, TrajFine utilize a fewer number of diffusion timestamps, reducing ADE from 7.61 to 7.11 and FDE from 14.30 to 13.28.

4.3. Ablation Study

Main	**Auxiliary**		**ADE**	**FDE**
TrajFine	**Scene Mixup**	**Length CL**		
			7.59	14.84
✓			7.25	13.51
✓	✓		7.25	<u>13.30</u>
✓		✓	<u>7.13</u>	13.46
✓	✓	✓	**7.11**	**13.28**

Table 3. Component Analysis on SDD dataset. ✓denotes that the component is included. Length CL means length-aware CL.

Component Analysis. We further analyze the effect of each component in our proposed method on SDD dataset. In Table 3, we provide ablation studies on all of model components: TrajFine, Scene Mixup, and length-aware Curriculum Learning. The performance demonstrates a notable improvement with the primary method, TrajFine. Moreover, Scene Mixup particularly enhances the model's understanding of diverse social interactions, focusing on the long-term goal that continually preserves the social behavior, which strongly correlates to its final decision and makes better performance in FDE. On the other hand, CL employs progressive learning, adeptly managing predicted trajectories from short to long, ensuring that prediction of each timestamp is thoroughly trained, thereby resulting in enhanced ADE. In summary, the joint integration of these three introduced components yields the most favorable outcomes.

DDIM inference timestamps	**ADE**	**FDE**
5	7.44	13.90
10	**7.11**	**13.28**
25	<u>7.19</u>	<u>13.38</u>
50	7.25	13.54

Table 4. DDIM timestamps analysis on SDD dataset. We report the different inference timestamps for the diffusion process. All results are under the identical timestamps during training.

DDIM Timestamps Analysis. In this part, we discuss the impact of DDIM timestamps configuration. As observed from the experimental results of Song *et al*. [38]. When utilizing fewer sampling timestamps, the generated outcomes are commendable, yet they slightly fall short in comparison to those produced with larger timestamps. This phenomenon is also reflected in our own experiments shown as Table 4. We can observe that the best performance is achieved at 10 timestamps. Additionally, considering the

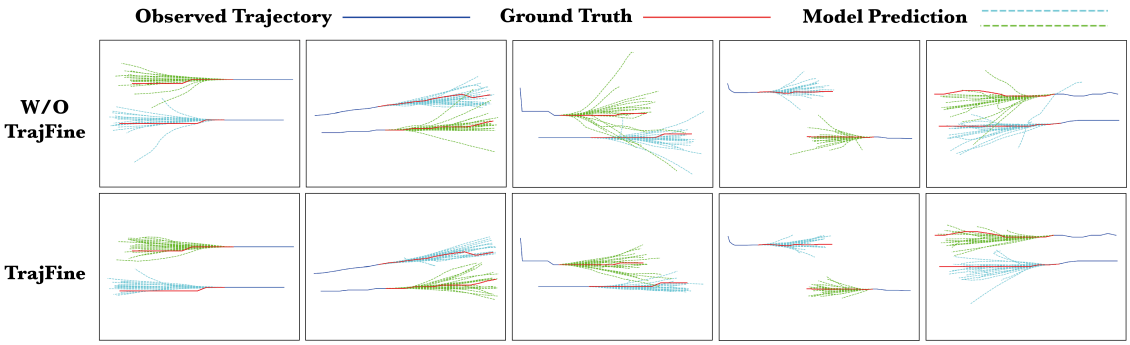

Figure 3. Qualitative results for trajectory forecasting. First row visualize results without TrajFine (only ϵ_θ). Second row visualize the predictions with TrajFine (denoising with $\epsilon_\theta + \epsilon_\phi$)

real-time nature of the trajectory prediction task, decent performance is achieved with just 5 timestamps.

4.4. Visualization Results

In Figure 3, we investigate whether the model's predicted walking area can be reduced by using TrajFine to mitigate collisions. By incorporating our method, which enforces social force principles on future trajectories, the predictions retain a subtle margin and reduce collision between agents, resulting in more rational and purposeful trajectory forecasts. Moreover, it becomes evident that the predicted plausible walking area in the bottom row (with TrajFine) is more reasonable than the top row (without TrajFine). This finding indicates that taking the future trajectories of multiple agents into account can prevent collisions.

Limitations. Our model is trained solely on path data. However, real-world environments often entail obstacles to be avoided or a necessity to walk on pedestrian pathways. As evident from Figure 4, where the model encounters sudden turns due to environmental factors may lead to inaccurate predictions. Therefore, beyond trajectory information, it would be beneficial to incorporate map image data [26] to enhance the model's accuracy in situations that demand sudden movements. This consideration of map image data remains a prospect for future work, as it holds the potential to significantly refine trajectory prediction accuracy.

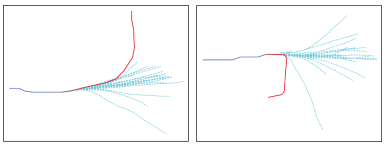

Figure 4. We showcase some failure situations of TrajFine using merely the trajectory information, where the agent may encounter obstacles or areas of the map that are not permitted for pedestrians.

Scene Mixup Samples. As depicted in Figure 5, the predicted trajectories in distinct color schemes (green and blue) correspond to trajectory data originating from different two scenes. Our Scene Mixup method, as demonstrated, facilitates the diversity of trajectory data with more intricate social interactions, while concurrently preserving the inherent social forces from the source scenes.

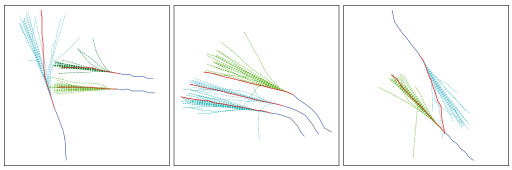

Figure 5. The augmented agents from Scene Mixup contain more complex social interactions, such as walking vertically with trajectory crossing or moving side by side in different directions.

5. Conclusion

We introduce TrajFine to address two common issues in previous diffusion-based trajectory predictors, *i.e.*, insufficient interactions and scene incompetence. To tackle the former, TrajFine extracts social information from predicted future trajectories and refines them, ensuring adherence to social force principles. To address the scene incompetence issue for comprehensively exploring diverse environments, we employ Scene Mixup that breaks the confines of existing datasets, enabling adaptation to intricate scenes. Additionally, our Curriculum Learning strategy progressively increases difficulty from both data and task perspectives, resulting in remarkable performance. The experimental results on the ETH-UCY and SDD datasets show valuable improvements by TrajFine against current SOTA methods, demonstrating that our TrajFine can generate plausible predictions.

Acknowledgment

This work is partially supported by the National Science and Technology Council, Taiwan under Grants, NSTC-112-2628-E-002-033-MY4, NSTC-112-2634-F-002-002-MBK, NSTC-112-2221-E-A49-059-MY3 and NSTC-112-2221-E-A49-094-MY3.

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 *CVPR*, 2016. 2

[2] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 0–0, 2019. 1, 2

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery. 3

[4] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *CVPR*, pages 8483–8492, 2019. 2

[5] Guangyi Chen, Junlong Li, Nuoxing Zhou, Liangliang Ren, and Jiwen Lu. Personalized trajectory prediction via distribution discrimination. In *ICCV*, pages 15580–15589, 2021. 2

[6] Guangyi Chen, Zhenhao Chen, Shunxing Fan, and Kun Zhang. Unsupervised sampling promoting for stochastic human trajectory prediction. In *CVPR*, pages 17874–17884, 2023. 6, 7

[7] Yujiao Cheng, Liting Sun, Changliu Liu, and Masayoshi Tomizuka. Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction. *IEEE Robotics and Automation Letters*, 5(2):2602–2609, 2020. 1

[8] Hyeong Kyu Choi, Joonmyung Choi, and Hyunwoo J. Kim. Tokenmixup: Efficient attention-guided token-level data augmentation for transformers. In *NeurIPS*, 2022. 5

[9] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, pages 14367–14376, 2021. 3

[10] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *ICCV*, 2021. 2

[11] Omar Elharrouss, Noor Almaadeed, and Somaya Al-Maadeed. A review of video surveillance systems. *Journal of Visual Communication and Image Representation*, 77: 103116, 2021. 1

[12] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, pages 17113–17122, 2022. 1, 2, 6, 7

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018. 1, 2, 5, 6, 7

[14] Dirk Helbing and Pé ter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, 1995. 2

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851. Curran Associates, Inc., 2020. 1, 3, 5

[16] 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 (ICCV)*, 2019. 2

[17] Avinash Kumar Jha, Awishkar Ghimire, Surendrabikram Thapa, Aryan Mani Jha, and Ritu Raj. A review of ai for urban planning: Towards building sustainable smart cities. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 937–944, 2021. 1

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 3

[19] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *NeurIPS*. Curran Associates, Inc., 2019. 2

[20] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010. 3

[21] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher Bongsoo Choy, Philip H. S. Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. *CVPR*, pages 2165–2174, 2017. 2

[22] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, pages 1721–1728, 2011. 3

[23] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by Example. *Computer Graphics Forum*, 2007. 6

[24] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, 2011. 1

[25] 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 *ECCV*, 2020. 6, 7

[26] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *ICCV*, pages 15233–15242, 2021. 8

[27] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *CVPR*, pages 5517–5526, 2023. 2

[28] Jitender Maurya, Keyur R. Ranipa, Osamu Yamaguchi, Tomoyuki Shibata, and Daisuke Kobayashi. Domain adaptation using self-training with mixup for one-stage object detection. In *WACV*, pages 4178–4187, 2023. 5

[29] P. Morerio, J. Cavazza, R. Volpi, R. Vidal, and V. Murino. Curriculum dropout. In *ICCV*, pages 3564–3572, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 3

[30] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, pages 1368–1377, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 5

[31] B. Pang, T. Zhao, X. Xie, and Y. Wu. Trajectory prediction with latent belief energy-based model. In *CVPR*, pages 11809–11819, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 7

[32] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009. 6

[33] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, pages 13756–13766, 2023. 1, 2

[34] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565. Springer, 2016. 6

[35] 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 *CVPR*, pages 1349–1358, 2019. 1, 2

[36] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, pages 683–700. Springer, 2020. 6, 7

[37] Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *ICCV*, pages 9675–9684, 2023. 7

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 5, 7

[39] Petru Soviany. Curriculum learning with diversity for supervised computer vision tasks. *arXiv preprint arXiv:2009.10625*, 2020. 3

[40] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Baby steps: How "less is more" in unsupervised dependency parsing. In *NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning*, 2009. 3

[41] J. Sun, Y. Li, H. Fang, and C. Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *ICCV*, pages 13230–13239, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 7

[42] Li-Wu Tsao, Yan-Kai Wang, Hao-Siang Lin, Hong-Han Shuai, Lai-Kuan Wong, and Wen-Huang Cheng. Social-ssl: Self-supervised cross-sequence representation learning based on transformers for multi-agent trajectory prediction. In *ECCV*. Springer, 2022. 2

[43] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *ECCV*, pages 511–528. Springer, 2022. 2

[44] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, 2020. 2

[45] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, 2021. 1, 2, 3, 4, 6, 7

[46] S. Yun, D. Han, S. Chun, S. Oh, Y. Yoo, and J. Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 5

[47] Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. A self-paced multiple-instance learning framework for co-saliency detection. In *ICCV*, pages 594–602, 2015. 3

[48] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 5

[49] He Zhao and Richard P. Wildes. Where are you heading? dynamic trajectory prediction with expert goal examples. In *ICCV*, pages 7609–7618, 2021. 7