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

RealTraj: Towards Real-World Pedestrian Trajectory Forecasting

Ryo Fujii[1,2], Hideo Saito[1,2] and Ryo Hachiuma[3]

Abstract— **This paper jointly addresses three key limitations in conventional pedestrian trajectory forecasting: pedestrian perception errors, real-world data collection costs, and person ID annotation costs. We propose a novel framework, *RealTraj*, that enhances the real-world applicability of trajectory forecasting. Our approach includes two training phases—self-supervised pretraining on synthetic data and weakly-supervised fine-tuning with limited real-world data—to minimize data collection efforts. To improve robustness to real-world errors, we focus on both model design and training objectives. Specifically, we present Det2TrajFormer, a trajectory forecasting model that remains invariant to tracking noise by using past detections as inputs. Additionally, we pretrain the model using multiple pretext tasks, which enhance robustness and improve forecasting performance based solely on detection data. Unlike previous trajectory forecasting methods, our approach fine-tunes the model using only ground-truth detections, reducing the need for costly person ID annotations. In the experiments, we comprehensively verify the effectiveness of the proposed method against the limitations, and the method outperforms state-of-the-art trajectory forecasting methods on multiple datasets.**

I. Introduction

The pedestrian trajectory forecasting task aims to predict the future positions of pedestrians based on their past observed states, a critical capability for understanding motion and behavioral patterns [1]. This task plays a crucial role in real-world applications such as social robotics navigation, autonomous driving, and surveillance systems. Over the years, deep learning models have dominated this domain due to their representational capabilities. However, existing approaches are evaluated under fixed, well-known benchmarks with ideal experimental conditions, such as completely annotated trajectories with the pre-recorded videos. This limits their applicability to real-world scenarios, necessitating further performance enhancements for practical deployment. In this paper, we address three key limitations present in existing approaches, as described below.

Pedestrian Perception Errors. Conventional approaches often assume that each pedestrian's past trajectory is perfectly obtained through upstream perception modules: pedestrian detection and tracking. However, this assumption is unrealistic in real-world scenarios, where obtaining complete trajectories is challenging due to detection errors (*e.g.*, miss-detections and localization errors) and tracking errors (*e.g.*, identity switches). When trajectory forecasting models use imperfect trajectories as input, their performance will degrade significantly [2], [3], [4], [5], [6].

This work was partially supported by JSPS KAKENHI Grant Number 25H01159.

[1]Keio University, [2]Keio AI Research Center, [3]NVIDIA. ryo.fujii0112@keio.jp

Real-World Data Collection Costs. Conventional trajectory forecasting approaches are typically trained on large-scale datasets and evaluated on data domains similar to the training (*e.g.*, consistent camera angles and sensor setups). However, these methods often overlook the substantial costs associated with real-world data collection, including the acquisition of raw sensor data such as point clouds or videos, which requires significant manual effort. Thus, it is crucial to develop prediction models that can be trained effectively while minimizing real-world data collection efforts.

Person ID Annotation Costs. In addition to the challenge of large-scale real-world data collection, most existing trajectory forecasting models rely on fully supervised training. This approach necessitates ground-truth future trajectories, including precise pedestrian positions and consistent person identities across frames.

In this paper, we propose a novel trajectory forecasting framework, RealTraj, designed to enhance the real-world applicability of trajectory forecasting by jointly addressing aforementioned three limitations. Our approach consists of two training phases—self-supervised pretraining and weakly-supervised fine-tuning—followed by an inference phase. First, inspired by advancements in training with synthetic data in image domains [7], we leverage large-scale synthetic trajectory data generated from a simulator [8] to effectively learn trajectory patterns using only synthetic data, reducing the need for extensive real-world data collection (addressing the second limitation). However, relying solely on clean synthetic data weakens robustness against real-world errors.

To mitigate this, we enhance both the model design and training objectives: we introduce *Det2TrajFormer*, a trajectory forecasting model designed to be invariant to tracking noise by only inputs past *detections* (without any person indices across frames). Additionally, inspired by the recent progress of self-supervised learning mechanism [9], [10], we pretrain the model with multiple self-supervised pretext tasks, such as unmasking, denoising, and person identity reconstruction. Unmasking and denoising improve robustness against perception errors (addressing the first limitation), while person identity reconstruction helps enhance forecasting performance using detections alone.

Finally, unlike previous trajectory forecasting approaches, we propose training the model exclusively on ground-truth detections during the fine-tuning phase, thereby reducing person ID annotation costs (addressing the third limitation) on the real data. We further introduce an acceleration regularization term to discourage abrupt changes in acceleration, resulting in smoother and more realistic trajectory forecastings. During inference, Det2TrajFormer takes detection results as

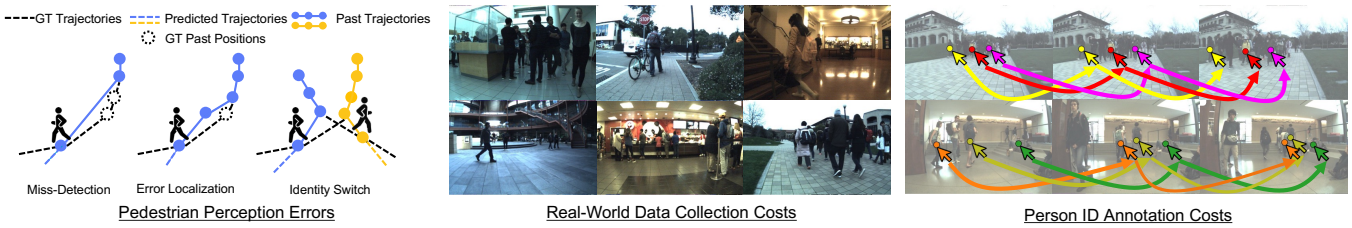


Fig. 1: Our paper addresses the three limitations in the existing pedestrian trajectory forecasting task. (Left) Pedestrian perception errors can significantly degrade trajectory forecasting performance. (Middle) Real-world data collection necessitates substantial manual effort. (Right) Person ID annotations require extensive manual labor.

input to forecast future trajectories.

In summary, our contributions are three-fold: (1) We introduce a novel trajectory forecasting framework, *RealTraj*, with self-supervised pretraining on large-scale synthetic data and weakly-supervised fine-tuning on limited real-world data, enhancing real-world applicability by jointly addressing pedestrian perception errors, data collection costs, and person ID annotation costs within a unified and single framework. (2) We propose three self-supervised pretext tasks to improve robustness against pedestrian perception errors and enhance trajectory forecasting accuracy using only detections. (3) We reduce person ID annotation costs by training the model solely on ground-truth detections via proposed weakly-supervised loss.

II. Related Work

A. Robust Trajectory Forecasting

Pedestrian trajectory forecasting models aim to predict future positions based on observed trajectories. Deep learning methods demonstrating strong performance due to their representational capabilities [1].

Despite the significant advancements, most trajectory forecasting models operate under idealized conditions, relying on ground-truth past trajectories for training and evaluation while ignoring the impact of imperfect inputs. Recent studies have increasingly recognized the challenges posed by perception module errors [5] and adversarial attacks [11]. This work focuses on enhancing robustness against perception module errors, which are unavoidable in real-world deployments and can significantly degrade prediction accuracy [5]. Some approaches address incomplete observations, encompassing missing data [12], momentary visibility [13] caused by occlusions or limited viewpoints (*i.e.*, detection errors). Others focus on mitigating tracking errors [14], [4]. Unlike the previous approaches, which independently tailored the approach for the specific error types, we aim to improve the robustness jointly against both detection (miss and wrong detection) and tracking errors within a single framework.

B. Adapting Trajectory Forecasting

Conventional trajectory forecasting models often heavily depend on specific training data domains, overlooking the substantial costs and manual effort required for real-world data collection, such as collecting the videos to obtain the trajectories. To address this challenge, recent research has focused on lightweight methods for adapting pretrained prediction models to newly captured data. Some approaches target cross-domain transfer [15], while others emphasize online adaptation [16], and continual learning [17]. These methods have demonstrated notable improvements in forecasting performance. However, they often assume access to ground-truth trajectories, which require costly person ID annotations. To reduce this burden, we propose to fine-tune the model using only ground-truth detections, thereby eliminating the extensive need for person ID annotations.

C. Self-Supervised Learning

Self-supervised learning (SSL) is a promising approach that enables models to learn valuable latent features from unlabeled data. Through pretraining on pretext tasks and pseudo-labels derived from data, followed by fine-tuning on downstream tasks, SSL has facilitated significant advancements in computer vision [9] and natural language processing (NLP). However, few studies have explored SSL in trajectory forecasting [18], [19]. Applying SSL to trajectory forecasting poses unique challenges, as it typically requires large-scale annotated data for pretraining. Unlike fields such as computer vision and NLP, which benefit from abundant unlabeled data, trajectory forecasting relies on annotated trajectories—often involving costly sensor setups and extensive human annotation—limiting scalability and the potential of SSL. Inspired by advancements in training with synthetic data in image domains [7], we explore self-supervised pretraining using synthetic data to reduce data collection and annotation costs. Additionally, unlike previous works that focus primarily on learning more extensive and adaptable latent features [18], [19], we introduce multiple pretext tasks designed to enhance model robustness against pedestrian perception errors and improve trajectory forecasting performance from detection inputs.

III. RealTraj

A. Problem Formulation

We aim to forecast the future trajectory of a target pedestrian in the scene based on detections from observed frames, including the non-targeted pedestrians in the environment. Unlike prior work, our approach skips the tracking step and instead uses detections directly, thereby avoiding the propagation of tracking errors into the forecasting task. Formally, let $Y = (y_1, y_2, \ldots, y_{T_{pred}}) \in \mathbb{R}^{T_{pred} \times 2}$ denote

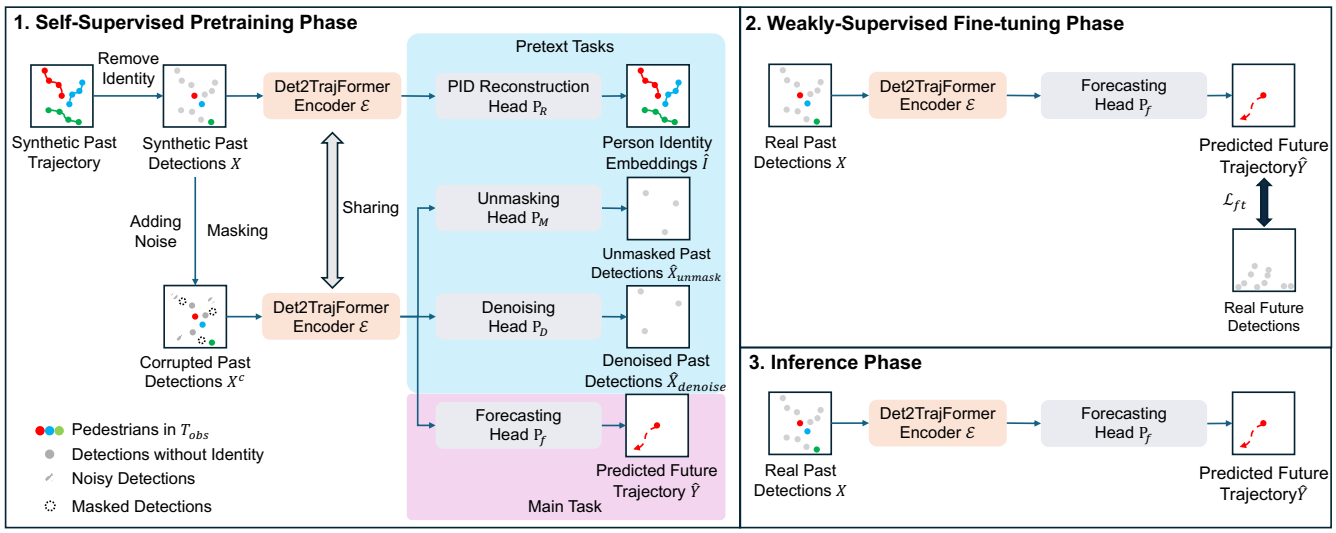


Fig. 2: Our proposed framework consists of two training phases and an inference phase. (1) Self-supervised pretraining on synthetic data using multiple pretext tasks. (2) weakly-supervised fine-tuning on real ground-truth detections. (3) Future trajectory inference based solely on detections.

the future trajectory of the target pedestrian over T_{pred} time steps where each position is represented as a tuple $y_t = (u_t, v_t) \in \mathbb{R}^2$, and let $X = (X_1, X_2, \ldots, X_{T_{obs}}) \in \mathbb{R}^{KT_{obs} \times 2}$ represent the set of past detections of K pedestrians over T_{obs} time steps. The detections at time t are defined as $X_t = (x_t^1, x_t^2, \ldots, x_t^K) \in \mathbb{R}^{K \times 2}$, where each detection is the position of pedestrian $x_t^k = (u_t^k, v_t^k) \in \mathbb{R}^2$ at timestep t[1]. The target pedestrian is one of the K pedestrians in the last observed frame ($t = T_{obs}$). The model can be applied K times to predict K pedestrians' trajectories. The target pedestrian is specified by translating the detections to make one of the detections in the last frame to the origin. The goal is to learn a function $\mathcal{F}(X)$ that maps the input detections X to a predicted future trajectory Y of a target pedestrian, such that $Y = \mathcal{F}(X)$.

B. Overview

Our framework consists of two training phases—self-supervised pretraining and weakly-supervised fine-tuning—along with an inference phase, as illustrated in Figure 2. We leverage large-scale synthetic trajectory data from a simulator [8] to effectively learn patterns while reducing the need for extensive real-world data collection. To ensure real-world robustness, we approach from both the training scheme and the trajectory forecasting model $\mathcal{F}$, Det2TrajFormer. During the pretraining, we leverage multiple pretext tasks to improve robustness and forecasting accuracy using large-scale synthetic trajectory data. During the fine-tuning, the model is fine-tuned solely on real-world ground-truth detections, eliminating the need for costly person ID annotations. During inference, Det2TrajFormer predicts future trajectories from observed detections.

[1]Unlike the conventional trajectory forecasting approaches, our method takes a set of detections as input at each frame. Therefore, it is not guaranteed that x_t^1 and x_{t+1}^1 correspond to detections of the same pedestrian.

a) Det2TrajFormer.: Det2TrajFormer consists of an embedding layer $\mathcal{G} \in \mathbb{R}^{2 \to d}$, Transformer encode $\mathcal{E} \in \mathbb{R}^{d \to d}$, and a forecasting head $\mathcal{P}_f \in \mathbb{R}^{d \to 2}$. Each past detection x is input to the embedding layer to obtain d-dimensional embedding along with sinusoidal time index encoding. We concatenate $\mathcal{G}(X)$ with T_{pred} learnable query tokens ($Q \in \mathbb{R}^{T_{pred} \times d}$) and input them into $\mathcal{E}$ to obtain $\hat{H}$ and $\hat{Q}$. $\hat{H} \in \mathbb{R}^{KT_{obs} \times d}$ are output tokens corresponding to the input tokens, $\mathcal{G}(X)$. Finally, the future trajectories $\hat{Y}$ can be predicted from $\hat{Q}$, as specified as follows:

$$[\hat{H}, \hat{Q}] = \mathcal{E}([\mathcal{G}(X), Q]), \quad \hat{Y} = \mathcal{P}_f(\hat{Q}), \tag{1}$$

where $[\cdot]$ denotes the concatenation operation along the token axis. Following recent works [19], [18], our approach can incorporate multiple forecasting heads ($\mathcal{P}_f^n$) to generate N possible future predictions: $\hat{Y}^n = (\hat{y}_1^n, \hat{y}_2^n, \ldots, \hat{y}_{T_{pred}}^n) \in \mathbb{R}^{T_{pred} \times 2}, \quad n = 1, 2, \ldots, N$. For simplicity, we describe our method using a single future prediction setting.

C. Self-Supervised Pretraining Phase

Using synthetically generated trajectories, we train the model on the primary forecasting task while incorporating pretext tasks to enhance performance and improve robustness against detection errors. From the synthetically generated trajectories, we generate a set of past detections X by removing the identity information. Additionally, we simulate the miss-detection and localization errors to X by randomly masking and adding Gaussian noises, resulting in the corrupted past detections, X^C. We propose three pretext tasks that aim to reconstruct the masked detections from X^C (unmasking), denoise the X^C (denoising), and reconstruct the removed person ID information from X (person ID reconstruction). Only during the pretraining, we additionally employ three heads $\mathcal{P}_M \in \mathbb{R}^{d \to 2}$, $\mathcal{P}_D \in \mathbb{R}^{d \to 2}$, and $\mathcal{P}_R \in \mathbb{R}^{d \to d_I}$, where d_I denotes the dimension of person ID embedding.

Trajectory Forecasting Task. The trajectory forecasting task aims to forecast the future trajectory from detections. To improve robustness against detection errors during inference, we forecast the future trajectory $\hat{Y}$ from corrupted detections X^C using Equation (1).

Unmasking Task. The goal of the unmasking pretext task is to encourage the encoder $\mathcal{E}$ to learn robust features against miss-detections. Specifically, we input $\hat{H}$ obtained from corrupted detection X^C by Equation (1) into the unmasking head $\mathcal{P}_M$, as follows: $\hat{X}_{unmask} = \mathcal{P}_M(\hat{H})$.

Denoising Task. The goal of the denoising pretext task is to encourage the encoder to learn robust features that mitigate localization errors. Similarly, the denoising is formulated as follows: $\hat{X}_{denoise} = \mathcal{P}_D(\hat{H})$.

Person Identity Reconstruction Task. The person identity reconstruction pretext task aims to enhance the model's ability to effectively associate pedestrian instances across observation frames. To achieve this, the model is trained to reconstruct pedestrian identities from detections X by predicting the person identity embeddings. The person identity reconstruction task is formulated as follows: $\hat{I} = \mathcal{P}_R(\hat{H})$, where $\hat{I} \in \mathbb{R}^{KT_{obs} \times d_I}$ are the reconstructed person ID embeddings for past detections.

Loss. The model is trained to minimize the losses for the four tasks mentioned above:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_F(\hat{Y}, Y) + \alpha\mathcal{L}_M(\hat{X}_{unmask}, X) \\ + \beta\mathcal{L}_D(\hat{X}_{denoise}, X) + \gamma\mathcal{L}_I(\hat{I}, I),\end{aligned} \tag{2}$$

where $\mathcal{L}_F$, $\mathcal{L}_M$, $\mathcal{L}_D$, and $\mathcal{L}_I$ represent the trajectory forecasting[2], unmasking, denoising, and person identity reconstruction losses, respectively. We use the mean square error (MSE) loss between the targets and predictions for each task. The terms α, β, and γ are weight hyperparameters.

D. Weakly-Supervised Fine-tuning Phase

We fine-tune the model in a weakly-supervised manner by using future ground-truth detections $X_{T_{obs}+1:T_{obs}+T_{pred}}$, thereby reducing ID annotation costs in real-world data. To achieve this, we propose to employ two losses. The first loss is calculated between the predicted future position $\hat{y}_t$ and the closest ground-truth detection d_t^c at each future timestep, summing over the T_{pred} timesteps as follows:

$$d_t^c = \underset{m \in M}{\operatorname{argmin}} \|\hat{y}_t - d_t^m\|_2, \quad \mathcal{L}_W = \sum_{t=T_{obs}+1}^{T_{obs}+T_{pred}} \|\hat{y}_t - d_t^c\|_2. \tag{3}$$

However, this formulation poses a challenge: the predicted position may sometimes deviate from the true trajectory due to its reliance on the closest detection, resulting in unintended oscillations. To address this issue, we introduce an acceleration regularization term, which discourages abrupt changes in acceleration. This promotes smoother transitions and reduces fluctuations in the predicted trajectories. This results in smoother, more realistic trajectory forecasting that aligns with natural motion dynamics. The acceleration regularization term is formulated as follows:

$$\mathcal{L}_{Reg} = \sum_{t=T_{obs}+1}^{T_{obs}+T_{pred}-1} \|\hat{y}_{t+1} - 2\hat{y}_t + \hat{y}_{t-1}\|_2. \tag{4}$$

The overall loss function for fine-tuning is defined as:

$$\mathcal{L}_{ft} = \mathcal{L}_W + \lambda\mathcal{L}_{Reg}, \tag{5}$$

where the terms λ is weight hyperparameter.

IV. Experiments

A. Experimental Setup

Datasets: We evaluate the performance of RealTraj on six datasets: JRDB [20] (in both image coordinates, JRDB-Image, and world coordinates, JRDB-World), JTA [21], ETH-UCY [22], [23], SDD [24], WildTrack [25] and TrajImpute [26]. For JRDB, JTA, and WildTrack, we predict 12 future timesteps using 9 past timesteps. For ETH-UCY, SDD, and TrajImpute, we predict 12 future timesteps using 8 past timesteps following standard practice.

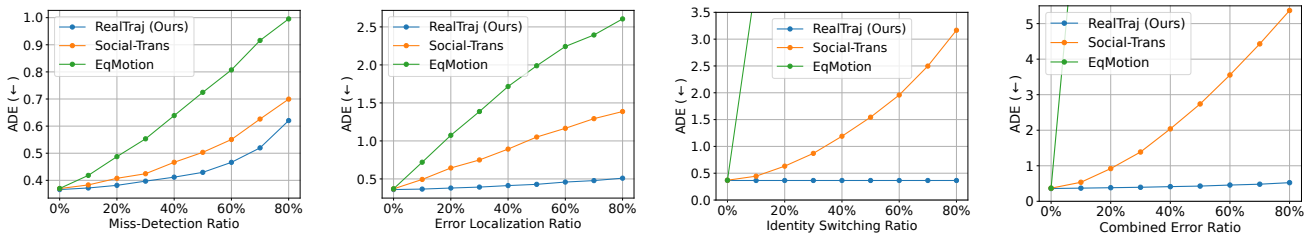
Evaluation Metrics: We use Average Displacement Error (ADE) and Final Displacement Error (FDE), where ADE measures the average error over predicted and ground-truth points, and FDE measures the endpoint error. For multi-future prediction, we also report minADE_N and minFDE_N over $N = 20$ predicted trajectories per pedestrian.

Implementation Details. Our training process is divided into two stages: VTP training and in-context training. In the first stage, we train the model using the AdamW optimizer with a base learning rate of 1×10^{-3} for 100 epochs. We perform a 3-epoch warmup and decay the learning rate to 0 throughout training using the cosine annealing scheduler. In the second stage, we train the model for 400 epochs, with a 12-epoch warmup and the cosine annealing scheduler, following the same setup as in the first stage. The hyperparameters were determined through a standard coarse-to-fine grid search or step-by-step tuning. We set the batch size to 16 and train the model using one NVIDIA RTX A6000 GPU. The model configuration for Predictor consists of three layers and four attention heads, with a model dimension of $d = 128$. We employ Leaky ReLU functions as the activation function. Det2TrajFormer is highly lightweight, requiring only 3.18M parameters and 0.62 GFLOPs per forward pass, which ensures low computational overhead. Data augmentation techniques, including rotation, noise addition, and masking, are applied.

B. Robustness Evaluation

Synthetic Perception Errors. The robustness of the proposed method against perception errors (detection and tracking errors) is compared with that of the transformer-based (Social-Trans [31]) and graph-based (EqMotion [30]) models, which are the well-known state-of-the-art methods across multiple benchmarks (*c.f.*, Table V). Although Social-Trans

[2]For the multi-future prediction, we employ winner-take-all strategy which only optimizes the best prediction with minimal average prediction error to the ground truth, following [18].



(a) Miss-Detection (b) Error Localization (c) Identity Switch (d) Combined Error

Fig. 3: Comparison of model robustness against various error types on the JRDB-World dataset, including detection errors (miss-detections and localization errors), tracking errors (identity switches), and their combined impact.

TABLE I: Comparison with the current state-of-the-art method on the Hard-Impute subset of TrajImpute. We report the minADE_{20} metric. The unit for minADE_{20} is meters. The best results are **bolded**, and the second-best results are underlined. Note that our model takes detections as inputs, while other approaches take the trajectory as inputs.

Method	ETH	HOTEL	UNV	ZARA1	ZARA2	Avg.
GraphTern[27]	0.78	1.68	0.50	0.96	0.37	0.86
LBEBM-ET [28]	0.85	3.31	0.64	0.37	0.27	1.09
SGCN-ET [28]	1.07	3.21	0.77	0.61	0.41	1.21
TUTR [29]	1.12	3.36	0.59	0.50	0.33	1.18
EqMotion [30]	**0.47**	0.72	0.39	**0.28**	0.37	0.45
RealTraj (Ours)	0.48	**0.20**	**0.37**	**0.28**	**0.21**	**0.30**

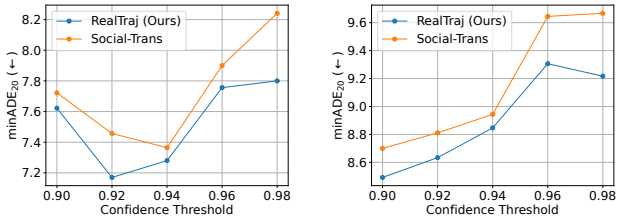
is designed to process multiple modalities, we used only trajectory data as input in all experiments to ensure fair comparisons. We synthetically generated two types of detection errors—miss-detections and localization errors—along with one type of tracking error, specifically identity switches, and combined these errors for evaluation. Miss-detections were simulated by setting past coordinates to zero for our model and linearly interpolated for comparison models, while localization errors were introduced by adding Gaussian noise to the inputs. Identity switches were simulated by swapping identities with nearby pedestrians within a 5-meter radius. Both detection and tracking errors were introduced for combined errors.

Figure 3 (a, b) shows that the performance of comparison methods significantly degrades when detection errors are introduced. In contrast, the proposed method exhibits relatively minor performance degradation, demonstrating robustness against detection errors. Figure 3 (c) presents a comparison of robustness against tracking errors. While the comparison methods show high sensitivity to tracking errors, our proposed method remains unaffected due to its network architecture, Det2TrajFormer, which forecasts future trajectories directly from detection inputs without relying on tracking information. Lastly, in Figure 3 (d), we compare robustness against combined detection and tracking errors, where RealTraj consistently outperforms comparison methods by a significant margin across all error ratios.

Additionally, we evaluate RealTraj's robustness against missing data (*i.e.*, miss-detection) using the TrajImpute [26]. Since the combination of the imputation method SAITS [32] and EqMotion achieved the best performance in the TrajImpute benchmark, we adopt SAITS as the imputation method in this evaluation. The results, presented in Table I, show that RealTraj outperforms all comparison methods across most subsets. This includes significant improvements on the challenging HOTEL subset, where the imputation is especially difficult [26].

TABLE II: Comparison with the current state-of-the-art methods using inputs from the upstream detector and tracker perception modules. The T and D in the Input column denote that the models

Input	Method	JRDB-Image		WildTrack	
		minADE_{20}	minFDE_{20}	minADE_{20}	minFDE_{20}
T	EqMotion [30]	7.82	9.61	7.85	9.30
	Social-Trans [31]	7.36	9.03	7.44	8.74
D	RealTraj	**7.32**	**8.84**	**7.40**	**8.62**



(a) JRDB (b) WildTrack

Fig. 4: We compare the robustness of the models with perception errors. We varied the confidence threshold value of the object detector to generate input data with perception errors.

Realistic Perception Errors. To validate our method's robustness against upstream perception errors, we evaluated its performance using outputs from a dedicated perception module. This module comprises a Faster R-CNN detector with FPN and a BoostTrack++ [33] tracker, pretrained on the MOT17 [34] dataset. We set the detector's confidence threshold to 0.5. The prediction models were then fine-tuned on the perception outputs from the training set and subsequently evaluated on the test set. As shown in table II, our model, RealTraj, consistently outperforms competing methods, demonstrating its resilience to imperfect inputs.

We further investigated our model's performance under severe perception errors. To simulate these conditions, we

TABLE III: Comparisons under a few-shot setting. The ADE metric is reported with different percentages of labeled real data during fine-tuning. The S and WS in the Setting column indicate that the models are trained in a fully-supervised and weakly-supervised manner, respectively. take trajectories and detections as inputs, respectively. The column (w/ Syn.) indicates whether the models are pretrained on the synthetic trajectory data.

(a) JRDB-World

Setting	Input	Method	w/ Syn.	0.1%	0.5%	1%	2%	5%
S	T	EqMotion [30]	✗	0.45	0.48	0.43	0.60	0.43
			✓	0.46	0.45	0.42	0.41	0.41
		Social-Trans [31]	✗	0.68	0.48	0.47	0.43	0.43
			✓	0.61	0.45	0.43	0.42	0.41
	D	RealTraj (Ours)	✗	0.92	0.88	0.88	0.87	0.86
			✓	**0.43**	**0.41**	**0.40**	**0.38**	**0.38**
WS	D	RealTraj (Ours)	✗	1.11	0.89	0.88	0.87	0.86
			✓	**0.43**	**0.41**	**0.40**	0.39	**0.38**

(b) JTA

Setting	Input	Method	w/ Syn.	0.1%	0.5%	1%	2%	5%
S	T	EqMotion [30]	✗	1.76	1.68	1.55	1.55	1.55
			✓	1.75	1.55	1.53	1.46	1.43
		Social-Trans [31]	✗	4.80	3.33	2.91	2.78	2.78
			✓	2.48	1.63	1.49	1.39	1.29
	D	RealTraj (Ours)	✗	4.83	4.74	4.67	1.43	1.37
			✓	**1.38**	**1.25**	**1.22**	**1.20**	**1.07**
WS	D	RealTraj (Ours)	✗	4.98	4.93	4.91	4.88	4.86
			✓	2.13	1.98	1.68	1.46	1.23

(c) ETH-UCY

Setting	Input	Method	w/ Syn.	0.1%	0.5%	1%	2%	5%
S	T	EqMotion [30]	✗	0.70	0.56	0.53	0.53	0.53
			✓	0.81	0.60	0.60	0.61	0.56
		Social-Trans [31]	✗	1.05	0.62	0.55	**0.52**	0.53
			✓	0.57	0.56	**0.52**	**0.52**	**0.50**
	D	RealTraj (Ours)	✗	0.81	0.72	0.60	0.61	0.57
			✓	**0.56**	**0.55**	0.53	**0.52**	0.52
WS	D	RealTraj (Ours)	✗	1.37	1.13	0.81	0.67	0.57
			✓	**0.56**	0.56	0.56	0.53	0.53

(d) SDD

Setting	Input	Method	w/ Syn.	0.1%	0.5%	1%	2%	5%
S	T	EqMotion [30]	✗	24.7	20.0	18.5	19.0	17.6
			✓	25.8	20.9	19.3	19.2	19.8
		Social-Trans [31]	✗	45.0	32.7	20.5	21.8	19.0
			✓	31.5	**18.1**	**17.9**	**17.4**	17.3
	D	RealTraj (Ours)	✗	67.4	41.8	38.3	36.3	18.7
			✓	**22.3**	20.2	19.2	18.5	**17.0**
WS	D	RealTraj (Ours)	✗	70.6	54.4	41.2	38.7	19.5
			✓	23.5	20.3	19.7	19.0	17.9

TABLE IV: Comparison with the current state-of-the-art methods for momentary trajectory prediction.

Input	Method	ETH-UCY		SDD	
		minADE_{20}	minFDE_{20}	minADE_{20}	minFDE_{20}
T	MOE [35]	0.20	0.41	8.40	16.08
	DTO [36]	0.23	0.46	8.32	**11.56**
	BCDiff [13]	**0.19**	**0.39**	8.93	16.92
D	RealTraj	0.24	**0.39**	**8.05**	13.32

increased the detector's confidence threshold from 0.2 to over 0.9, which intentionally filters out less confident detections and thus elevates error rates. As plotted in Figure 4, our method's performance, measured by the minADE_N metric, degrades more gracefully than that of Social-Trans. These results highlight our model's superior robustness to realistic perception errors.

Momentary Trajectory Prediction. We evaluate our model in a momentary trajectory prediction setting, where only two frames serve as input, and the model predicts future trajectories for the next 12 timesteps. The results in table IV show that our model, which integrates three key limitations into a unified framework, achieves performance comparable to specialized momentary prediction approaches.

These findings confirm that the proposed model significantly enhances robustness against perception errors, aligning with our goal of addressing the first robustness limitation described in Section I.

C. Few-shot Evaluation

Next, we demonstrate the effectiveness of RealTraj with limited real-world labeled data. We randomly selected subsets of 0.1%, 0.5%, 1%, 2%, and 5% of the datasets, fine-tuned the model on each subset, and evaluated it on the test set on JRDB-World, JTA, ETH-UCY, and SDD. In this experiment, since no prior work focuses on trajectory prediction with a few-shot setting, we compare our model against the existing approaches, EqMotion [30] and Social-Trans [31]. We prepare the comparison models that are pretrained on the synthetic trajectory data, the same as ours, for a fair comparison.

As shown in Table III, our method consistently outperforms these prior methods on JRDB-World. In the most challenging scenario with only 0.1% of annotations, weakly-supervised RealTraj (last row) achieves improvements of 4.4%, 20%, and 4.9% on JRDB-World, ETH-UCY, and SDD, respectively, compared to the best-performing previous supervised model. In other data settings, RealTraj achieves results comparable to supervised methods despite the detection inputs and requiring only weak detection annotations during fine-tuning. Pretraining on large-scale synthetic data enhances performance across all methods, with our approach showing particular benefits. Unlike baseline models that process trajectories, our detection-based method leverages the person identity reconstruction task during pretraining, yielding substantially greater accuracy improvements (as shown in Section IV-E). These results demonstrate that our model overcomes the second limitation of real-world data collection cost mentioned in Section I. Furthermore, the weakly-supervised RealTraj (last row) achieved the performance on par with the fully-supervised RealTraj on JRDB and SDD datasets, verifying that our approach reduces the person ID annotation cost mentioned in Section I.

D. Comparison with State-of-the-art

We further compare RealTraj with a diverse range of existing approaches under the standard 100% labeled setting, as summarized in Table V. Although achieving state-of-the-art performance in a fully supervised setting with clean trajectory inputs is not the primary focus of our paper, the results show that our weakly supervised framework performs comparably to fully supervised approaches using only detection inputs on the various benchmarks. Additionally, our fully supervised approach surpasses all other methods on the

TABLE V: Comparisons with state-of-the-art methods under a fully-supervised setting.

Setting	Input	Method	JRDB-World		JTA		ETH-UCY				SDD			
			ADE	FDE	ADE	FDE	ADE	FDE	minADE_{20}	minFDE_{20}	ADE	FDE	minADE_{20}	minFDE_{20}
S	T	Trajectron++ [37]	0.40	0.78	1.18	2.53	0.53	1.11	**0.21**	0.41	-	-	-	-
		Transformer [38]	0.56	1.10	1.56	3.54	0.54	1.17	0.31	0.55	-	-	-	-
		Autobots [39]	0.39	0.80	1.20	2.70	0.52	1.05	-	-	-	-	-	-
		TUTR [29]	-	-	-	-	0.53	1.12	**0.21**	0.36	17.4	35.0	7.76	12.7
		EigenTrajectory [28]	-	-	-	-	0.51	1.11	0.22	0.37	20.7	41.9	8.12	13.1
		EqMotion [30]	0.42	0.78	1.13	2.39	**0.49**	**1.03**	**0.21**	0.35	16.8	33.7	8.45	14.1
		Social-Trans† [31]	0.40	0.77	0.99	1.98	0.51	**1.03**	**0.21**	0.41	18.0	38.7	7.21	14.3
		EmLoco [40]	0.37	0.72	0.97	1.91	-	-	-	-	-	-	-	-
	D	RealTraj (Ours)	**0.35**	**0.67**	**0.92**	**1.84**	0.51	**1.03**	0.23	0.42	**16.0**	**31.8**	**7.18**	12.4
WS	D	RealTraj (Ours)	0.36	0.71	1.04	2.11	0.52	1.07	0.26	0.43	16.0	32.0	8.21	13.8

TABLE VI: Ablation study of the proposed pretext tasks. The table reports the ADE metric.

Main Task	Pretext Tasks			Complete	Detection Errors	
F	P	U	D		Miss-Detection	Error Localization
-	-	-	-	0.78	0.91	0.84
✓	-	-	-	0.42	0.53	0.70
-	✓	✓	✓	0.39	0.47	0.51
✓	-	✓	✓	0.42	0.52	0.62
✓	✓	-	-	0.41	0.51	0.54
✓	✓	-	✓	0.38	0.47	0.51
✓	✓	✓	-	0.38	0.44	0.45
✓	✓	✓	✓	**0.36**	**0.42**	**0.43**

JRDB-World, JTA, and SDD datasets in terms of ADE and remains competitive on the ETH-UCY dataset.

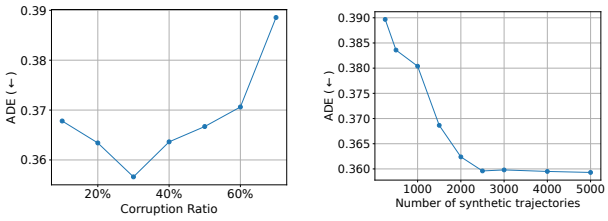
E. Ablation Study

Effectiveness of Pretext Tasks. We first examine the effect of incorporating our proposed pretext tasks during pretraining on the JRDB-World, as shown in Table VI. In addition to using complete inputs, we conducted ablation with inputs containing detection errors to demonstrate the contribution of each pretext task to robustness. Our results show that all pretext tasks enhance trajectory forecasting performance. First, the performance significantly degrades when removing the pretraining stage even with the complete input, verifying the effectiveness of pretraining for enabling the trajectory prediction with detections. Moreover, the person identity reconstruction task improves ADE by 7.9% (from 0.39 to 0.36). Both the unmasking and denoising tasks also contribute to performance gains, each enhancing ADE by 5.3% (from 0.38 to 0.36). Joint training with all three pretext tasks and the main forecasting task improves ADE by 14% (from 0.42 to 0.36).

The pretext tasks also contribute to robustness against detection errors. Specifically, the unmasking pretext task improves ADE by 11% when using inputs with miss-detection errors, while the denoising pretext task enhances ADE by 4.4% when using inputs with localization errors. Combining the unmasking and denoising tasks results in further improvements, boosting ADE by 18% (from 0.51 to 0.42) for inputs with miss-detections and by 20% (from 0.51 to 0.43) for inputs with localization errors. Notably, pretraining is conducted using synthetic trajectories, achieving these results without additional data collection.

TABLE VII: Ablation study of acceleration regularization weight λ on the JTA.

Metric	λ			
	0	1	10	100
ADE	1.16	1.09	**1.04**	3.67



(a) Effect of corruption ratio during training.

(b) Effect of the number of synthetic sequences.

Fig. 5: Effect of corruption ratio during training and number of synthetic trajectories.

Effectiveness Acceleration Regularization. Hyperparameter acceleration regularization λ is used to balance the two terms in Equation (5). We conducted ablations with different values of acceleration regularization λ on JTA in Table VII. The results indicate that setting $\lambda = 10$ provides the best performance for both ADE and FDE metrics, yielding a 10% improvement in ADE (from 1.16 to 1.04) compared to performance without acceleration regularization ($\lambda = 0$).

Effect of Corruption Ratio. We explore the effect of different corruption (adding noise and masking) ratios during pretraining in Figure 5 (a). Setting the corruption ratios either too low or too high results in suboptimal performance, as the pretext tasks become either too easy or too challenging, limiting their effectiveness for model learning. A moderate corruption ratio yielded the best results.

Fig. 6: Visualization of predicted trajectories on the JRDB-Image, WildTrack, and SDD datasets. The primary agent's ground truth path is shown in red and our model's prediction is in orange. Past detections are marked with gray circles.

F. Qualitative Results

Figure 6 visualizes the proposed method's predicted results and the ground-truth future trajectories, highlighting the method's accuracy in predicting future trajectories in complex scenes only using past detections as inputs.

V. Conclusion

In this paper, we introduced *RealTraj*, a novel pedestrian trajectory forecasting framework designed to address three key limitations in conventional works. Our approach combines self-supervised pretraining on synthetic data with weakly-supervised fine-tuning on limited real-world data, reducing data collection requirements while enhancing robustness to perception errors. We proposed Det2TrajFormer, a model that leverages past detections to remain robust against tracking noise. By incorporating multiple pretext tasks during pretraining, we further improved the model's robustness and forecasting accuracy using only detection inputs. Unlike existing methods, our framework fine-tunes only using ground-truth detections, reducing person ID annotation costs. In our experiments, we thoroughly validated that our approach effectively overcomes the key limitations of existing trajectory prediction models, particularly when applied to real-world scenarios.

Limitation Our model currently only predicts trajectories for pedestrians detected in the final observed frame. Future work could extend this to individuals observed in any past frame, even if occluded, using Mean Forecasting Average Precision [5] for evaluation. Additionally, our motion smoothness regularization may limit the prediction of abrupt maneuvers. Addressing these points remains a key direction for future research.

References

[1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *IJRR*, 2019.

[2] R. Yu and Z. Zhou, "Towards Robust Human Trajectory Prediction in Raw Videos," in *IROS*, 2021.

[3] X. Weng, B. Ivanovic, and M. Pavone, "MTP: Multi-hypothesis Tracking and Prediction for Reduced Error Propagation," in *IV*, 2022.

[4] X. Weng, B. Ivanovic, K. Kitani, and M. Pavone, "Whose Track Is It Anyway? Improving Robustness to Tracking Errors with Affinity-based Trajectory Prediction," in *CVPR*, 2022.

[5] Y. Xu, L. Chambon, É. Zablocki, M. Chen, A. Alahi, M. Cord, and P. Pérez, "Towards Motion Forecasting with Real-World Perception Inputs: Are End-to-End Approaches Competitive?" in *ICRA*, 2024.

[6] R. Fujii, R. Hachiuma, and H. Saito, "CrowdMAC: Masked Crowd Density Completion for Robust Crowd Density Forecasting," in *WACV*, 2025.

[7] H. Kataoka, K. Okayasu, A. Matsumoto, E. Yamagata, R. Yamada, N. Inoue, A. Nakamura, and Y. Satoh, "Pre-training without Natural Images," *IJCV*, 2022.

[8] J. P. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *ISRR*, 2011.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," in *CVPR*, 2022.

[10] Q. Wu, H. Ye, Y. Gu, H. Zhang, L. Wang, and D. He, "Denoising masked autoencoders help robust classification," in *ICLR*, 2023.

[11] R. Jiao, X. Liu, T. Sato, Q. A. Chen, and Q. Zhu, "Semi-supervised Semantics-guided Adversarial Training for Robust Trajectory Prediction," in *ICCV*, 2023.

[12] R. Fujii, J. Vongkulbhisal, R. Hachiuma, and H. Saito, "A Two-Block RNN-Based Trajectory Prediction From Incomplete Trajectory," *IEEE Access*, 2021.

[13] R. Li, C. Li, D. Ren, G. Chen, Y. Yuan, and G. Wang, "Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction," in *NeurIPS*, 2023.

[14] C. Feng, H. Zhou, H. Lin, Z. Zhang, Z. Xu, C. Zhang, B. Zhou, and S. Shen, "MacFormer: Map-Agent Coupled Transformer for Real-Time and Robust Trajectory Prediction," *RAL*, 2023.

[15] Y. Xu, L. Wang, Y. Wang, and Y. Fu, "Adaptive trajectory prediction via transferable gnn," in *CVPR*, 2022.

[16] D. Park, J. Jeong, S.-H. Yoon, J. Jeong, and K.-J. Yoon, "T4P: Test-Time Training of Trajectory Prediction via Masked Autoencoder and Actor-specific Token Memory," in *CVPR*, 2024.

[17] F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, "Mantra: Memory augmented networks for multiple trajectory prediction," in *CVPR*, 2020.

[18] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders," *CVPR*, 2023.

[19] Y. Gao, P.-C. Luan, and A. Alahi, "Multi-Transmotion: Pre-trained Model for Human Motion Prediction," in *CoRL*, 2024.

[20] R. Martin-Martin, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese, "Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *TPAMI*, 2021.

[21] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to Detect and Track Visible and Occluded Body Joints in a Virtual World," in *ECCV*, 2018.

[22] S. Pellegrini, A. Ess, and L. Van Gool, "Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings," in *ECCV*, 2010.

[23] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese, "Learning an Image-Based Motion Context for Multiple People Tracking," in *CVPR*, 2014.

[24] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes," in *ECCV*, 2016.

[25] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret, "WILDTRACK: A Multi-Camera HD Dataset for Dense Unscripted Pedestrian Detection," in *CVPR*, 2018.

[26] P. S. Chib and P. Singh, "Pedestrian Trajectory Prediction with Missing Data: Datasets, Imputation, and Benchmarking," in *NeurIPS*, 2024.

[27] I. Bae and H.-G. Jeon, "A Set of Control Points Conditioned Pedestrian Trajectory Prediction," *AAAI*, 2023.

[28] I. Bae, J. Oh, and H.-G. Jeon, "EigenTrajectory: Low-Rank Descriptors for Multi-Modal Trajectory Forecasting," in *ICCV*, 2023.

[29] L. Shi, L. Wang, S. Zhou, and G. Hua, "Trajectory Unified Transformer for Pedestrian Trajectory Prediction," in *ICCV*, 2023.

[30] 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 *CVPR*, 2023.

[31] S. Saadatnejad, Y. Gao, K. Messaoud, and A. Alahi, "Social-Transmotion: Promptable Human Trajectory Prediction," in *ICLR*, 2024.

[32] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, 2023.

[33] V. Stanojević and B. Todorović, "BoostTrack++: using tracklet information to detect more objects in multiple object tracking," 2024.

[34] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *CoRR*, 2016.

[35] J. Sun, Y. Li, L. Chai, H.-S. Fang, Y.-L. Li, and C. Lu, "Human trajectory prediction with momentary observation," in *CVPR*, 2022.

[36] A. Monti, A. Porrello, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara, "How many observations are enough? knowledge distillation for trajectory forecasting," in *CVPR*, 2022.

[37] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data," in *ECCV*, 2020.

[38] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer Networks for Trajectory Forecasting," in *ICPR*, 2021.

[39] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D'Souza, S. E. Kahou, F. Heide, and C. Pal, "Latent Variable Sequential Set Transformers for Joint Multi-Agent Motion Prediction," in *ICLR*, 2022.

[40] H. Taketsugu, T. Oba, T. Maeda, S. Nobuhara, and N. Ukita, "Physical plausibility-aware trajectory prediction via locomotion embodiment," in *CVPR*, 2025.