S2F2: Single-Stage Flow Forecasting for Future Multiple Trajectories Prediction

Yu-Wen Chen, Hsuan-Kung Yang, Chu-Chi Chiu, and Chun-Yi Lee

Elsa Lab, Department of Computer Science, National Tsing Hua University, Taiwan
{carrie, hellochick, chulie9710, cylee}@gapp.nthu.edu.tw

Abstract. In this work, we present a single-stage framework, named **S2F2**, for forecasting multiple human trajectories from raw video images by predicting future optical flows. S2F2 differs from the previous two-stage approaches in that it performs detection, Re-ID, and forecasting of multiple pedestrians at the same time. The architecture of S2F2 consists of two primary parts: (1) a *context feature extractor* responsible for extracting a shared latent feature embedding for performing detection and Re-ID, and (2) a *forecasting module* responsible for extracting a shared latent feature embedding for forecasting. The outputs of the two parts are then processed to generate the final predicted trajectories of pedestrians. Unlike previous approaches, the computational burden of S2F2 remains consistent even if the number of pedestrians grows. In order to fairly compare S2F2 against the other approaches, we designed a StaticMOT dataset that excludes video sequences involving egocentric motions. The experimental results demonstrate that S2F2 is able to outperform two conventional trajectory forecasting algorithms and a recent learning-based two-stage model, while maintaining tracking performance on par with the contemporary MOT models.

Keywords: Multiple trajectory forecasting, optical flow estimation, single-stage forecasting framework, S2F2.

1 Introduction

Multiple pedestrian trajectory forecasting is the task of predicting future locations of pedestrians from video data, and has received increasing attention in recent years across a wide variety of domains. Foreseeing how a scene involving multiple objects will unfold over time is crucial for a number of applications, such as self-driving vehicles, service robots, and advanced surveillance systems.

In the past few years, multiple pedestrian trajectory forecasting from image sequences has been implemented as a two-stage process: (1) *the detection and tracking stage*, where targets in a single video frame are first located (i.e., detection), and then associated to existing trajectories (i.e., tracking) with or without the help of re-identification (Re-ID); and (2) *the forecasting stage*, where the previous trajectory of each person is fed into a forecasting model to predict its potential future locations over a short period of time. This branch of methods is

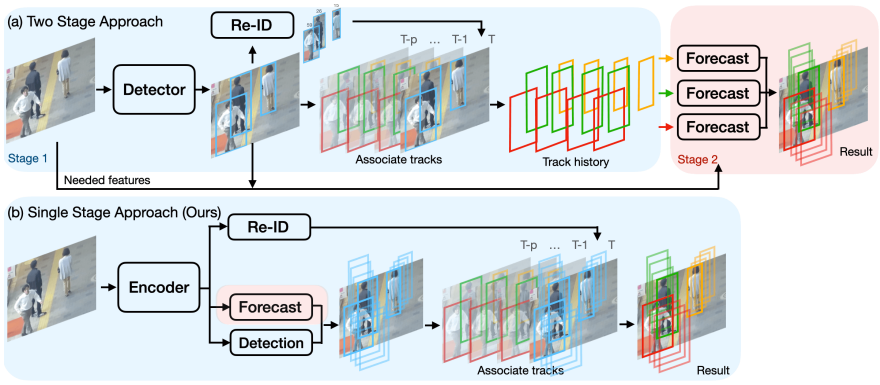

Fig. 1. A comparison between the two-stage approach and our one stage framework.

referred to as the *two-stage approach* in this work, and is illustrated in Fig 1 (a). Among them, previous works concentrated only on the second stage, and utilized pre-processed bounding boxes and tracking histories [21,5,2,28,6,13]. Albeit effective, two-stage approaches inherently suffer from several limitations. First, their forecasting performances are constrained by the quality and correctness of the first stage. Second, despite that the first stage only processes the input in one pass, the second stage usually requires multiple passes of forecasting if the input image sequence contains multiple pedestrians [21,13,28].

In light of these shortcomings, a promising direction to explore is the use of a single-stage architecture. Single-stage architectures often possess favorable properties such as multitasking, fast inference speed, etc., and have recently been investigated in a wide range of other domains [31,24,32,22,19,27]. The advantages of these one-stage approaches usually come from the bottom-up design philosophy, where their feature maps are typically constructed from features of local regions, and optimized to encompass certain hierarchies of different scales if necessary. Such a design philosophy allows them to make multiple predictions in one shot, regardless of the number of target instances in an image. Despite their successes, the previous single-stage approaches are mostly designed for tasks involving only a single image frame. The multiple pedestrian trajectory forecasting task, however, requires temporal information encoded from multiple past frames, making previous single-stage architectures not readily applicable. As this problem setup has not been properly investigated, the challenges to be addressed are twofold. First, it requires various types of information (e.g., detection results, past trajectories, context features, etc.) to be concurrently encrypted to the latent features. Second, it necessitates temporal information to facilitate plausible predictions. Therefore, this multiple pedestrian trajectory forecasting problem can be considered as a unique and complicated multitask learning problem. To this end, we present the first single-stage framework, called **S2F2**, for predicting multiple pedestrian trajectories from raw video images. S2F2 is inspired by the concept of optical flow forecasting, and is constructed atop the

design philosophy of an anchor-free one-stage multiple object tracking (MOT) framework [31]. Fig. 1 highlights the differences between S2F2 and the prior two-stage approaches. S2F2 differs from them in that it performs detection, Re-ID, as well as forecasting of multiple pedestrians at the same time. Unlike two-stage approaches, the computational burden of S2F2 remains consistent even if the number of pedestrians grows. We show that with the same amount of training data, S2F2 is able to outperform two conventional trajectory forecasting algorithms and a recent learning-based two-stage model [21], while maintaining its tracking performance on par with the contemporary MOT models. The main contributions of this work are:

1. We present the first single-stage framework that jointly accomplishes tracking and forecasting of multiple pedestrians from raw video image frames.
2. We introduce a future flow decoder and a special loss function to enable the predictions of future optical flows without any additional labeled data.
3. We propose to leverage the predicted optical flow maps to assist in forecasting the trajectories of multiple pedestrians concurrently, within a consistent computational burden even if the number of pedestrians increases.

2 Related Work

The task of forecasting the trajectories of multiple pedestrians typically requires their track histories. To date, the existing methods [1,5,21,20,2,28,6,13,26] are all carried out in a two-stage fashion: (1) object detection and tracking, and (2) forecasting. The former stage is responsible for extracting features and associating bounding boxes, while the latter utilizes the information from the former to forecast their potential future locations. These two stages have been treated by these methods separately, instead of being integrated as a single model. In these methods, detection is usually based on the ground truths provided by the datasets [21,26,17,10,18,4], which offer continuously tracked bounding boxes or centers. On the other hand, forecasting is performed either based on the track histories alone [2,1,28,5,6], or with a combination of additional extracted context features [13,26,21,11,20,15]. Social-LSTM [1] introduces the concept of social interactions by proposing a technique called social pooling, which encodes the latent features of multiple trajectories through LSTMs for sharing the information of interactions among pedestrians in a scene. In order to make reasonable forecasts, some researchers proposed to further incorporate content or context features into their architectures [21,20,11], such as semantic segmentation [13], optical flow [21,20], human pose [26], ego motion [26,15], etc., and have demonstrated the effectiveness of them. Such additional features are extracted separately by distinct deep neural networks. Albeit effective, these two-stage methods suffer from the issues discussed in Section 1. Although single-stage forecasting has been attempted for point cloud based input data captured by lidars [12], there is no single-stage forecasting method that makes predictions based on raw RGB image frames.

3 Methodology

In this section, we first describe the problem formulation of this work. Then, we introduce the proposed S2F2 framework, followed by a detailed description of its various task modules.

3.1 Problem Formulation

Consider a sequence of raw RGB images from a static scene $\{I_0, I_1, ..., I_t\}$, where t represents the current timestep, our objective is to estimate and track the current and future locations of all pedestrians. Given the tracking information encoded from the previous images, the task of multiple pedestrian trajectory forecasting aims to infer a set of bounding boxes $B_t^i = \{b_t^i, b_{t+1}^i, b_{t+2}^i, ..., b_{t+n}^i\}$ for each identifiable person i in the current and the subsequent n image frames, where b_i^t denotes the bounding box of person i at timestep t.

3.2 Overview of the S2F2 Framework

Fig. 2 illustrates an overview of the S2F2 framework. To accomplish trajectory forecasting for multiple pedestrians within a single stage, S2F2 employs two distinct modules in its architecture: (a) a *context feature extractor* for processing and encrypting the input RGB image frame of the current timestep t, and (b) a *forecasting module* for recurrently encoding the latent features and predicting the future optical flows, which are later exploited for deriving the future trajectories of the pedestrians in the image. Given a raw input image I_t, it is first processed by the backbone K of the context feature extractor to generate a feature embedding $\mathcal{X}_t$, which is used for three purposes: detection, Re-ID, and forecasting. To derive the future flow maps, the forecasting module takes $\mathcal{X}_t$ as its input, and leverages a set of gated recurrent units (GRUs) to generate a series of optical flow maps $\{f_{t+1}, f_{t+2}, ..., f_{t+n}\}$ for the subsequent n timesteps. These optical flow maps represent the estimated offsets of each pixel from I_t to I_{t+n}, and thus can be utilized to perform forward warping of the detection results to derive the future bounding boxes B_t^i for each identifiable person i in the scene, as depicted in Fig. 2 (highlighted as the blue bounding boxes). Finally, all the bounding boxes are processed by a tracking algorithm, and are associated into distinct tracks.

3.3 Context Feature Extractor

The context feature extractor of S2F2 inherits the design from FairMOT [31], in which an enhanced version of Deep Layer Aggregation (DLA) [32] is used as the backbone to generate $\mathcal{X}_t$. Except for forwarding it to the forecasting module, the other two objectives of the context feature extractor is to utilize $\mathcal{X}_t$ to extract necessary features for the detection and Re-ID tasks. These two tasks are accomplished by four heads, including a heatmap head, an offset head, a size head, and a Re-ID head. These heads ensure that $\mathcal{X}_t$ can serve as an adequate

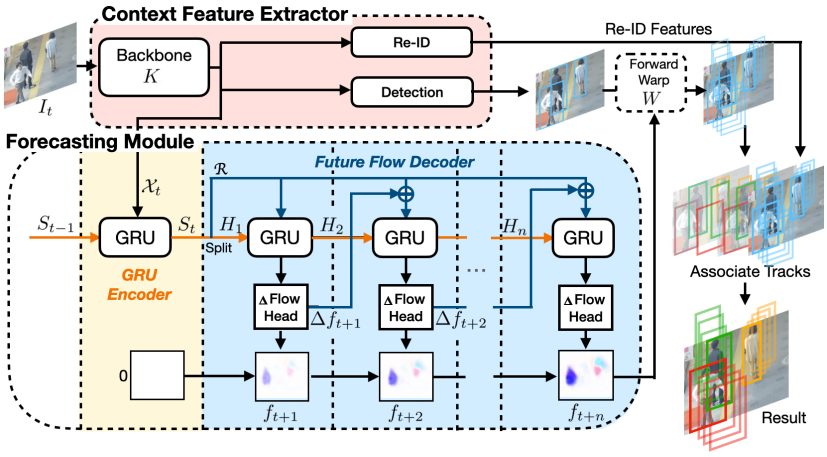

Fig. 2. The proposed S2F2 framework. Our model contains two modules: (a) a context feature extractor for processing I_t, and (b) a forecasting module for aggregating past features to predict future flow maps. Forecasting is accomplished by forward warping the detection results, followed by a tracking algorithm to associate them into tracks.

representation of the locations and the appearances of the objects in I_t, and offer sufficient information for the forecasting module. We explain the functionality of each head in the following paragraphs.

- **Heatmap Head.** The heatmap head is responsible for estimating the locations of the centers of different bounding boxes in an input image.
- **Offset and Size Heads.** The function of the offset head is to precisely locate the objects, while the role of the size head is to estimate the height and width of the target bounding boxes.
- **Re-ID Head.** The function of this head is to extract features for generating a unique identifier for each person so as to track its bounding boxes across different frames.

3.4 Forecasting Module

The forecasting module is in charge of encoding temporal features based on $\mathcal{X}_t$, and predicting the future optical flow maps which are later used to derive the future trajectories. It contains an encoder and a decoder, which are depicted in Fig 2 and explained as follows.

GRU Encoder Block. The goal of the gated recurrent unit (GRU) encoder block is to generate an embedding S_t from $\mathcal{X}_t$. It is a single convolutional GRU (ConvGRU) [25]. At timestep t, $\mathcal{X}_t$ is passed into the ConvGRU along with the corresponding previous embedding S_{t-1} to derive the updated $S_t = GRU(S_{t-1}, \mathcal{X}_t)$. S_t can be considered as a summary of the past state embeddings up to t. Note that at $t = 1$, $\mathcal{X}_1$ is also utilized as the initial state embedding S_0.

Future Flow Decoder Block. The objective of the future flow decoder is to predict n residual future flow estimations $\{\Delta f_{t+1}, \Delta f_{t+2}, ..., \Delta f_{t+n}\}, \Delta f \in \mathbb{R}^{2\times w\times h}$, where each estimation is an update direction used to update a fixed flow field initialized with zeros. More specifically, this decoder generates a set of future flow maps $F = \{f_{t+1}, f_{t+2}, ..., f_{t+n}\}$, where $f_{t+1} = \Delta f_{t+1} + f_t$. Each of the predicted flows in F has the same initial reference frame I_t (i.e., f_n is the optical flow from frame I_t to I_{t+n}, instead of $I_{t+(n-1)}$ to I_{t+n}). This design choice aims to avoid error accumulation while forming forecasting predictions. Similar to the encoder block, the decoder also contains a ConvGRU. It takes the state embedding S_t as its input, and splits S_t into a hidden state H_1 and an input R. They are then fed separately into the ConvGRU to generate the next hidden state $H_2 = GRU(H_1, R)$, which is utilized by a Δ *flow head* to produce Δf_{t+1}. This, in turn, is used to generate the subsequent input to the ConvGRU by concatenating Δf_{t+1} with R. The above procedure repeats n times, where each iteration stands for a timestep into the future.

To train the future flow decoder block, a loss function consisting of two parts are employed. The first part is a supervised loss for the centers of future bounding boxes, given by:

$$L_{Center} = \sum_{\tau=t+1}^{t+n} \sum_{i=1}^{K_\tau} \|c_\tau^i - \hat{c}_\tau^i\|_1 = \sum_{\tau=t+1}^{t+n} \sum_{i=1}^{K_\tau} \|c_\tau^i - (c_t^i + f_\tau(c_t^i))\|_1, \tag{1}$$

where f_τ is the estimated future flow, c_τ^i and $\hat{c}_\tau^i$ represent the centers of the annotated bounding box b_τ^i and the predicted bounding box $\hat{b}_\tau^i$ of pedestrian i at timestep τ, respectively, and K_τ denotes the number of pedestrians in the image at timestep τ. For each center c_τ, the forecasted center $\hat{c}_\tau$ can be inferred from f_τ, i.e., $\hat{c}_\tau = c_t + f_\tau(c_t)$. Please note that, instead of directly warping the entire frame I_t using f_τ, we only warp the centers of bounding boxes appeared at timestep t. This is because warping the entire frame may cause occlusions, and leave behind duplicate pixels. The second part further refines the flow maps and stabilizes the training process with the structural similarity index (SSIM) loss adopted in several unsupervised optical flow estimation works [7,29,16,23]:

$$L_{SSIM} = \sum_{\tau=t+1}^{t+n} SSIM\Big(I_t, W(I_\tau, f_\tau)\Big), \tag{2}$$

where $W(\cdot)$ is the backward warping operator, and the SSIM loss L_{SSIM} is obtained by calculating the similarities of the corresponding pixels between I_t and its warped frame I_τ.

3.5 Online Association with Forecasting Refinement

In order to enhance the tracking performance with the information provided from the forecasted results, we modify the original tracking algorithm of FairMOT [31] by not only considering the current bounding boxes b_t, but taking the bounding

boxes $\hat{b}_t$ forecasted from the previous timestep $t-1$ into consideration. In the original design, only the bounding boxes predicted with confidence scores higher than a threshold δ are associated into tracks. However, this might result in missing objects and fragmented trajectories, since objects with low confidence scores are neglected (e.g., occluded objects) [30]. To alleviate this issue, we reduce the threshold value if the distance of any $\hat{c}_t$ and c_t is within a predefined range r, which can be formulated as:

$$\forall i \in K, \delta_i = \begin{cases} \delta/2, & \text{if } \exists \hat{c}_t, \|\hat{c}_t - c_t^i\|_1 < r \\ \delta, & \text{otherwise} \end{cases}, \tag{3}$$

where K denotes the number of pedestrians in I_t, and δ_i is the threshold for pedestrian i. This design allows the bounding boxes with lower confidence scores to be re-considered and associated if their previously forecasted locations are nearby. We examine the effectiveness of this design in Section 4.4.

3.6 Training Objective

We trained S2F2 in an end-to-end fashion by minimizing the following objective:

$$L_{all} = \frac{1}{e^{w_1}} L_{det} + \frac{1}{e^{w_2}} L_{id} + \frac{1}{e^{w_3}} L_{fut} + w_1 + w_2 + w_3, \tag{4}$$

where $L_{fut} = L_{Center} + L_{SSIM}$, w_1, w_2 and w_3 are learnable parameters, and L_{det} and L_{id} are the losses for the detection and Re-ID tasks, respectively. In Eq. (4), we modify the formulation of the uncertainty loss proposed in [9] to balance the detection, Re-ID, and forecasting tasks.

4 Experimental Results

In this section, we first briefly introduce the settings used for training and validation, then evaluate S2F2 in terms of its tracking and forecasting performance.

4.1 Data Curation for Forecasting without Camera Movement

In this work, we examine the proposed S2F2 on the subset of the widely-adopted MOT17 and MOT20 Challenge Datasets [14,3]. The video sequences can be classified into two categories based on whether the ego-motion of the camera is involved. Image sequences with ego-motion are considered to be hard cases for the forecasting task, since additional designs may be required to handle the viewpoint movements. Some research works [21,26,17] focus on the image sequences from a first-person moving perspective, however, in our work, we concentrate on the model's capability of both tracking and forecasting, and thus the movements from the camera are not considered. As a result, we select a subset of video sequences from MOT17 and MOT20 without camera movement to form our dataset, named StaticMOT. The details of StaticMOT are shown in Table 1. We train and evaluate S2F2 on StaticMOT, with each sequence presented in Table 1 split into halves to form the training and validation sets, respectively.

Table 1. The video sequences contained in StaticMOT. Please note that 'Density' refers to the average number of pedestrians per frame.

Sequence	Frames	Density	Viewpoint	Sequence	Frames	Density	Viewpoint
MOT17-02	300	31.0	eye level	MOT20-02	1390	72.7	elevated
MOT17-04	525	45.3	elevated	MOT20-03	1002	148.3	elevated
MOT17-09	262	10.1	eye level	MOT20-05	1657	226.6	elevated
MOT20-01	214	62.1	elevated	Average	764	126.8	-

4.2 Trajectory Forecasting Results

In this section, we compare our approach with two conventional trajectory forecasting algorithms and a recent learning-based method STED [21]. To fairly compare different methods, the pre-processed bounding boxes and the necessary past trajectories of the pedestrians are generated by S2F2 from the validation set of StaticMOT. Tracks that are not continuously detected for six frames are discarded, resulting in around $470,000$ tracks for evaluation. We predict three future frames, corresponding to around one second of forecasting into the future.

1) Baselines:

- **Constant Velocity & Constant Scale (CV-CS):** We adopt the simple constant velocity model, which is used widely as a baseline for trajectory forecasting models and as a motion model for MOT. We only use the previous three frames to compute the velocity, instead of the whole past history, as this setting delivers better performance.
- **Linear Kalman Filter (LKF) [8]:** LKF is one of the most popular motion models for MOT, and is widely used for tracking objects and predicting trajectories under noisy conditions. We use the implementation in [31], and use the last updated motion value for forecasting. Unlike CV-CS, all the previous bounding box locations of a tracked object are utilized.
- **STED [21]:** STED is a recent two-stage pedestrian forecasting model with a GRU based encoder-decoder architecture. Instead of encoding image features, it encodes pre-computed bounding box information along with features extracted from pre-generated optical flow to forecast future bounding boxes. We follow the original implementatio of STED, and train it on the ground truth tracks from the StaticMOT training set for 20 epochs.

2) Forecasting Metrics:

- **(ADE, FDE):** Average displacement error (ADE) is defined as the mean Euclidean distance between the predicted and ground-truth bounding box centroids for all predicted bounding boxes, and final displacement error (FDE) is defined similarly but only for the final timestep.
- **(AIOU, FIOU):** Average intersection-over-union (AIOU) is defined as the mean intersection-over-union (IOU) of the predicted and ground truth bounding boxes for all predicted boxes, and final intersection-over-union (FIOU) is the mean IOU for the bounding boxes at the final timestep.

Table 2. The forecasting results evaluated on the StaticMOT validation set. The latency reported is evaluated on an NVIDIA Tesla V100 GPU.

Model	ADE(↓)	FDE(↓)	AIOU(↑)	FIOU(↑)	Latency (ms)
CV	14.481	20.196	0.673	0.594	-
LKF	20.635	24.323	0.581	0.512	-
STED	16.928	23.761	0.654	0.570	623.480
Ours	**12.275**	**16.228**	**0.704**	**0.643**	**13.788**

Table 3. The detailed forecasting results evaluated on the StaticMOT validation set.

Sequences	Boxes	AIOU (↑)			ADE (↓)			FIOU (↑)			FDE (↓)		
Model	-	CV	LKF	**Ours**	CV	Kal	**Ours**	CV	Kal	Ours	CV	Kal	**Ours**
MOT17-02	4202	0.58	0.536	**0.602**	17.136	22.112	**16.171**	0.514	0.483	**0.544**	22.423	23.447	**20.656**
MOT17-04	19506	0.702	0.654	**0.724**	11.672	16.304	**10.802**	0.638	0.585	**0.677**	15.651	19.187	**13.631**
MOT17-09	1302	0.508	0.426	**0.58**	67.857	62.232	**52.954**	0.364	0.248	**0.488**	104.309	121.644	**72.297**
MOT20-01	6343	0.606	0.495	**0.656**	23.962	33.395	**18.59**	0.483	0.371	**0.559**	36.096	44.601	**26.683**
MOT20-02	49985	0.647	0.534	**0.676**	21.449	31.446	**18.428**	0.547	0.435	**0.591**	30.977	37.856	**25.2**
MOT20-03	103031	0.698	0.568	**0.727**	8.351	15.334	**7.007**	0.627	0.524	**0.672**	11.571	14.666	**9.231**
MOT20-05	286362	0.67	0.592	**0.703**	15.165	20.45	**12.811**	0.592	0.521	**0.643**	20.96	24.906	**16.805**
Average	67247	0.673	0.581	**0.704**	14.481	20.635	**12.275**	0.594	0.512	**0.643**	20.196	24.323	**16.228**

3) Quantitative Results: Table 2 shows the quantitative results in terms of ADE/FDE and AIOU/FIOU for all methods on StaticMOT. The latency of S2F2 and STED are also included for comparison. The latency is calculated for the forecasting part only, and is tested on sequence MOT20-05 with the largest pedestrian density. Please note that for STED, the time needed for pre-computing optical flow is not included. It can be observed that, the proposed S2F2 outperforms all baselines, while running several times faster than STED. This is because STED requires feature extraction for every person in a scene. Another reason is that STED was designed for videos from the first person perspective, while the majority of StatcMOT are elevated sequences. Table 3 shows more detailed results on every sequence.

4) Qualitative Results: Fig. 3 shows three examples of the successful scenarios selected and evaluated from our StaticMOT validation set. From the left to the right, the scenarios are: (1) a person behind two people walks away from the viewpoint, (2) a person moves to the right and takes a sharp turn due to the lockers in his way, and (3) a person makes a right turn to follow the crowd. In the first scenario, the person's bounding boxes from different timesteps become closer to each other due to the increase in their distances from the viewpoint. This can be forecasted by S2F2, but is unable to be correctly predicted by CV-CS. In the second scenario, CV-CS also fails to estimate the trajectory of the person. However, S2F2 incorporates features from the whole images, enabling it to anticipate the lockers in the person's way. In the third scenario, since S2F2 makes predictions for all objects concurrently based on a dense flow field, it is thus capable of capturing the spatial correlations between different objects, allowing it to forecast the future trajectory of the person by taking into account

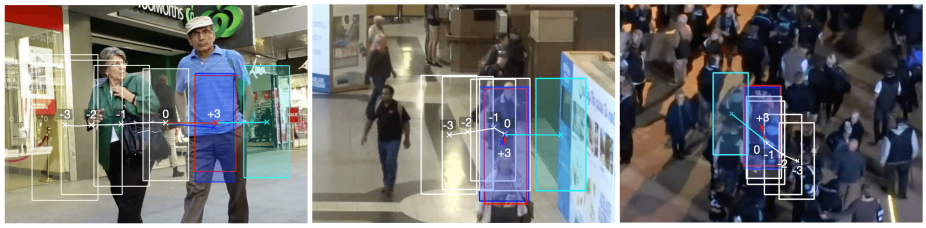

Fig. 3. Examples where the forecasting results made by S2F2 outperform CV-CS. From the left to the right, a pedestrian (1) walks away from the viewpoint, (2) makes a sharp turn due to the lockers in his way, and (3) makes a right turn to follow the crowd. The bounding boxes are highlighted in different colors to represent the ground truth (red), the past locations (white), and the predictions made by CV-CS (aqua) and those made by S2F2 (dark blue). The predictions are one second into the future.

Table 4. A comparison between the detection results of FairMOT [31] and S2F2. The results marked with * are taken directly from the FairMOT paper. The MOT17 test results are taken from the evaluation server under the "private detection" protocol.

Method	Dataset	MOTA(↑)	MOTP(↑)	IDs(↓)	IDF1(↑)
*FairMOT**	MOT17 test	69.8	-	**3996**	**69.9**
Ours	MOT17 test	**70.0**	80.15	4590	**69.9**
*FairMOT**	MOT17 val	67.5	-	**408**	69.9
Ours	MOT17 val	**67.7**	80.3	513	**71.0**
FairMOT	StaticMOT	73.1	80.5	**2283**	76.4
Ours	StaticMOT	**73.6**	**80.5**	2307	**76.6**

the behavior of the crowd. The failure scenarios are shown in Fig 4. From the left to the right, the scenarios are: (1) a person suddenly turns and runs to the left, (2) a person walks towards the viewpoint and is occluded by another person walking to the right, and (3) a person comes to a crosswalk and turns left instead of turning right to cross the street. In the first scenario, it is difficult for S2F2 to forecast sudden movements of the person. In the second scenario, it is possible for S2F2 to sample the wrong object center from our predicted flow when multiple people are overly close to each other. In the third scenario, our model incorrectly predicts the direction of that person because it is different from the majority of the people in the scene. More visualizations of our forecasting results and predicted future optical flow are shown in Fig 8.

4.3 Multiple Object Tracking Results

In addition to forecasting, Table 4 further compares the tracking results of S2F2 and FairMOT [31], the framework that S2F2 is based on. From top to bottom, the three categories correspond to the models trained on the whole official MOT17 training dataset [14], the training split of MOT17 from [31], and our StaticMOT, respectively. For each category, S2F2 and FairMOT are trained with the same

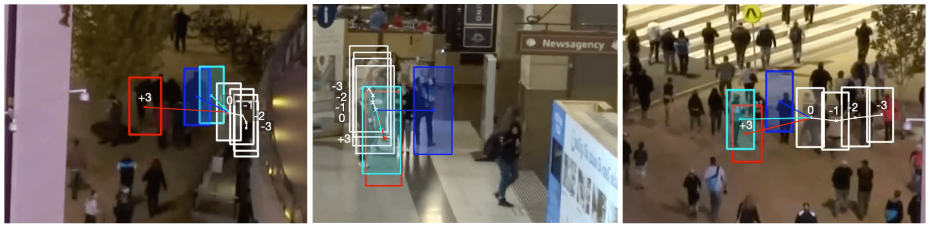

Fig. 4. Examples of the unsuccessful forecasting results made by S2F2. From the left to the right, a person (1) suddenly turns and runs to the left, (2) walks towards the viewpoint and is occluded by another person walking to the right, and (3) comes to a crosswalk and turns left instead of turning right to cross the street. The bounding boxes are highlighted in different colors to represent the ground truth (red), the past locations (white), and the predictions made by CV-CS (aqua) and those made by S2F2 (dark blue). The predictions are one second into the future.

set of data samples, and do not use any additional fine-tuning. It is observed from the results that our performance is on par or even slightly better than that of FairMOT for certain metrics, implying that the addition of our forecasting module does not affect its tracking capability. Note that S2F2 performs slightly worse than FairMOT in terms of the ID switch (IDS) metric. This might be due to the fact that FairMOT is trained on independent images, while S2F2 is trained on image sequences, thus causing slight overfitting.

4.4 Ablation Studies

In this section, we dive into a set of ablation studies to discuss the rationales of our design decisions and validate them.

Inference Speed. In Fig. 5, we separately time the detection, tracking, and forecasting portions of S2F2 on videos from StaticMOT with different numbers of pedestrians to validate our claim of consistent computational burden. The inference time of the two stage tracker (i.e., STED) is also included. All models are run on a single NVIDIA Tesla V100 GPU. As shown in the figure, the forecasting portion of S2F2 takes approximately 0.01 seconds per frame regardless of the number of pedestrians in the scene. On the other hand, the two stage tracker's inference time grows with the number of pedestrians. This supports our claim.

GRU Encoder Optimization. In this section, we validate the effectiveness of the design of the GRU encoder adopted in S2F2 and compare it against two different variants. Fig. 6 illustrates a comparison of the three architectures. The main objective of this ablation analysis is to validate whether incorporating features beneficial for predicting the optical flow from timestep $t-1 \rightarrow t$ would help the prediction of the future optical flow maps $F = \{f_{t+1}, f_{t+2}, ..., f_{t+n}\}$. To achieve this objective, an additional "past flow decoder" with a design similar to

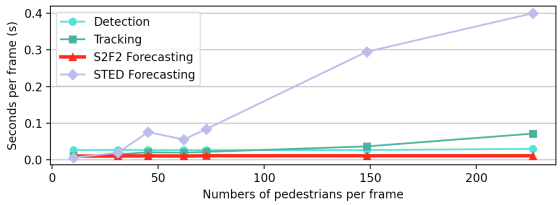

Fig. 5. A comparison of the inference time between a two-stage approach (STED) and our proposed one stage approach (S2F2).

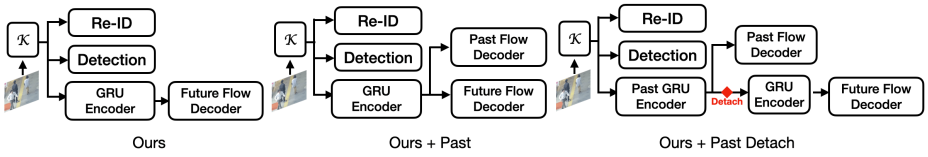

Fig. 6. An illustration of the three different model architectures discussed in Table 5.

the original future flow decoder block is incorporated into the variants shown in Fig. 6. It is trained using the unsupervised warping loss presented in Eq. (2) to predict the optical flow from $t-1 \rightarrow t$. The incorporation of the past flow decoder ensures that the relative motion between frames $t-1 \rightarrow t$ could be encoded in the feature embedding. For the two variants, $Ours+Past$ is the case where only the past flow decoder block is added. Notice that in $Ours+Past$, the gradients from both the past and future flow decoders are used for updating the GRU Encoder and the backbone network K. On the other hand, the variant $Ours+PastDetach$ further includes an additional "past GRU encoder" which is only updated by the gradients from the past flow decoder. The original GRU encoder is placed after the past GRU encoder, but is detached such that its gradients are not utilized for updating the past GRU encoder. This design aims to examine whether the features beneficial for predicting optical flows from $t-1 \rightarrow t$ could benefit the future flow prediciton.

Table 5 shows the results of the three different architectures on the MOT17 validation set, with MOTA representing the detection and tracking accuracy, and FDE representing the forecasting performance. It can be observed from Table 5 that both $Ours+Past$ and $Ours+PastDetach$ perform relatively unsatisfactory as compared to $Ours$. The detection results of $Ours + PastDetach$ are better than those of $Ours+Past$. However, the forecasting results demonstrate a different trend. The reasons are twofold. First, the features needed for predicting the optical flow from $t-1 \rightarrow t$ might not be suitable for predicting the future optical flow maps. This is supported by the evidence that $Ours + PastDetach$, which extracts features solely by the past GRU encoder, delivers the worst forecasting performance. Second, multi-task learning with tasks that need different representations might harm the performance of each individual tasks. As a result, $Ours+PastDetach$ shows better detection results as compared to $Ours+Past$.

Table 5. Ablation results for the GRU encoder optimization. All results are trained and validated on MOT17.

Ours + Past		Ours + PastDetach		Ours w/o L_{SSIM}		Ours	
MOTA(↑)	FDE(↓)	MOTA(↑)	FDE(↓)	MOTA(↑)	FDE(↓)	MOTA(↑)	FDE(↓)
66.0	43.097	66.2%	46.869	67.0	43.021	**67.7**	**39.891**

Table 6. Ablation results regarding changes to the tracking algorithm.

	Ours + BYTE		Ours w/o refinement		Ours	
Dataset	MOTA(↑)	FDE(↓)	MOTA(↑)	FDE(↓)	MOTA(↑)	FDE(↓)
MOT17	66.2	39.626	67.6	40.014	**67.7**	39.891
StaticMOT	73.4	16.146	73.3	16.249	**73.6**	16.228

An ablation study on the effectiveness of the loss function L_{SSIM} described in Section 3.4 is also presented in Table 5 under the column '$Ours$ w/o L_{SSIM}', corresponding to the case trained without L_{SSIM}. It can be observed that the performance declines if L_{SSIM} is not employed. Fig 7 further depicts that if L_{SSIM} helps S2F2 to concentrate on predicting the optical flows of the pedestrains.

Effectiveness of the Forecasting Refinement for Online Association. In this section, we validate the effectiveness of our forecasting refinement discussed in Section 3.5, and compare it with two different tracking variants. The results are presented in Table 6, in which $Ours + BYTE$ corresponds to the case where the tracking algorithm is replaced by ByteTrack [30] (denoted as BYTE). In this experiment, we use the implementation of BYTE that does not take Re-ID into consideration. On the other hand, '$Ours$ w/o $refinement$' corresponds to the case where the original tracking algorithm from FairMOT [31] is utilized without the forecasting refinement. It can be observed from the results that our proposed forecasting refinement does benefit the detection and tracking performance, thus validating its effectiveness. In contrast, $Ours + BYTE$ does not yield the most superior results, which might be due to the fact that ByteTrack [30] requires accurate detections. This can be inferred from the fact that $Ours + BYTE$ performs better when the model is trained on StaticMOT (a larger dataset) than the case where the model is trained solely on MOT17 (a smaller dataset).

5 Conclusion

In this paper, we presented the first single-stage framework, named S2F2, for predicting multiple human trajectories from raw video images. S2F2 performs detection, Re-ID, and forecasting of multiple pedestrians at the same time, with consistent computational burden even if the number of pedestrians grows. S2F2 is able to outperform two conventional trajectory forecasting algorithms, and a

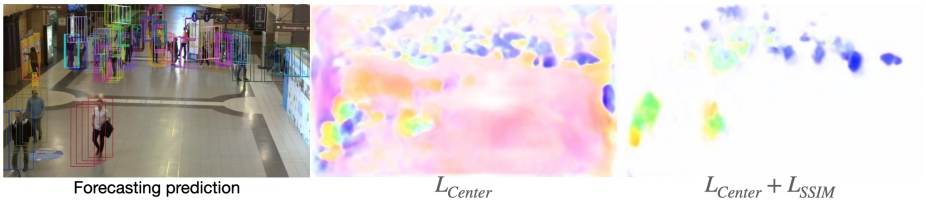

Fig. 7. Visualizations of the predicted flow maps from S2F2 trained with different loss terms.

Fig. 8. The tracking results and the predicted flow maps of S2F2 on the validation set of StaticMOT. Bounding boxes with different colors represent different identities.

recent two-stage learning-based model [21], while maintaining its tracking performance on par with the contemporary MOT models. We hope this sheds light on single-stage pedestrian forecasting, and facilitates future works in this direction.

6 Acknowledgments

This work was supported by the Ministry of Science and Technology (MOST) in Taiwan under grant number MOST 111-2628-E-007-010. The authors acknowledge the financial support from MediaTek Inc., Taiwan and the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) used in this research work. The authors thank National Center for High-Performance Computing (NCHC) for providing computational and storage resources. Finally, the authors would also like to thank the time and effort of the anonymous reviewers for reviewing this paper.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 961–971 (2016) 3
2. Ansari, J.A., Bhowmick, B.: Simple means faster: Real-time human motion forecasting in monocular first person videos on cpu. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10319–10326. IEEE (2020) 2, 3
3. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003[cs] (Mar 2020), arXiv: 2003.09003 7
4. Ess, A., Leibe, B., Schindler, K., , van Gool, L.: A mobile vision system for robust multi-person tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). IEEE Press (June 2008) 3
5. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2255–2264 (2018) 2, 3
6. Ivanovic, B., Pavone, M.: The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2375–2384 (2019) 2, 3
7. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 557–572. Springer (2020) 6
8. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960) 8
9. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7482–7491 (2018) 7
10. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer graphics forum. vol. 26, pp. 655–664. Wiley Online Library (2007) 3
11. Liu, Y., Li, R., Cheng, Y., Tan, R.T., Sui, X.: Object tracking using spatio-temporal networks for future prediction location. In: European Conference on Computer Vision. pp. 1–17. Springer (2020) 3
12. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3569–3577 (2018) 3
13. Makansi, O., Cicek, O., Buchicchio, K., Brox, T.: Multimodal future localization and emergence prediction for objects in egocentric view with a reachability prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4354–4363 (2020) 2, 3
14. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs] (Mar 2016), arXiv: 1603.00831 7, 10
15. Neumann, L., Vedaldi, A.: Pedestrian and ego-vehicle trajectory prediction from monocular camera. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10204–10212 (2021) 3

16. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12240–12249 (2019) 6
17. Rasouli, A., Kotseruba, I., Tsotsos, J.K.: Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 206–213 (2017) 3, 7
18. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European conference on computer vision. pp. 549–565. Springer (2016) 3
19. Shuai, B., Berneshawi, A.G., Modolo, D., Tighe, J.: Multi-object tracking with siamese track-rcnn. arXiv preprint arXiv:2004.07786 (2020) 2
20. Styles, O., Ross, A., Sanchez, V.: Forecasting pedestrian trajectory with machine-annotated training data. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 716–721. IEEE (2019) 3
21. Styles, O., Sanchez, V., Guha, T.: Multiple object forecasting: Predicting future object locations in diverse environments. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 690–699 (2020) 2, 3, 7, 8, 14
22. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. arXiv preprint arXiv:2103.14258 (2021) 2
23. Wang, Y., Wang, P., Yang, Z., Luo, C., Yang, Y., Xu, W.: Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8071–8081 (2019) 6
24. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 107–122. Springer (2020) 2
25. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015) 5
26. Yagi, T., Mangalam, K., Yonetani, R., Sato, Y.: Future person localization in first-person videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7593–7602 (2018) 3, 7
27. Yan, Y., Li, J., Qin, J., Bai, S., Liao, S., Liu, L., Zhu, F., Shao, L.: Anchor-free person search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7690–7699 (2021) 2
28. Yao, H.Y., Wan, W.G., Li, X.: End-to-end pedestrian trajectory forecasting with transformer network. ISPRS International Journal of Geo-Information **11**(1), 44 (2022) 2, 3
29. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1983–1992 (2018) 6
30. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021) 7, 13
31. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision pp. 1–19 (2021) 2, 3, 4, 6, 8, 10, 13
32. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European Conference on Computer Vision. pp. 474–490. Springer (2020) 2, 4