Multiple Object Forecasting: Predicting Future Object Locations in Diverse Environments

Olly Styles Tanaya Guha Victor Sanchez
University of Warwick
{o.c.styles, tanaya.guha, v.f.sanchez-silva}@warwick.ac.uk

Abstract

This paper introduces the problem of multiple object forecasting (MOF), in which the goal is to predict future bounding boxes of tracked objects. In contrast to existing works on object trajectory forecasting which primarily consider the problem from a birds-eye perspective, we formulate the problem from an object-level perspective and call for the prediction of full object bounding boxes, rather than trajectories alone. Towards solving this task, we introduce the Citywalks dataset, which consists of over 200k high-resolution video frames. Citywalks comprises of footage recorded in 21 cities from 10 European countries in a variety of weather conditions and over 3.5k unique pedestrian trajectories. For evaluation, we adapt existing trajectory forecasting methods for MOF and confirm cross-dataset generalizability on the MOT-17 dataset without fine-tuning. Finally, we present STED, a novel encoder-decoder architecture for MOF. STED combines visual and temporal features to model both object-motion and ego-motion, and outperforms existing approaches for MOF. Code & dataset link: `https://github.com/olly-styles/Multiple-Object-Forecasting`

1. Introduction

Predicting future events in video is a core problem in computer vision that has been studied in several contexts such as human action prediction [21], semantic forecasting [27], and road agent trajectory forecasting [22]. In this work, we focus on the task of pedestrian trajectory forecasting from video data, which has seen considerable research attention over recent years [19, 32, 1, 12, 44, 42]. Humans are a particularly challenging class of objects to predict, as they exhibit highly dynamic motion and may change speed or direction rapidly.

Much of the existing work on pedestrian trajectory forecasting considers the problem from a birds-eye view using footage from a fixed overhead camera, often considering each pedestrian as a single point in space [1, 12, 44]. This setting is effective for modeling crowd motion patterns and interactions with the environment. However, by simplifying each pedestrian as a point in space, salient visual features such as person appearance, body language, and individual characteristics are not considered. Prior research has shown that these features are of importance for trajectory prediction in settings such as anticipating if a pedestrian will cross the road [38, 30]. Furthermore, overhead perspectives are often not available in practical applications. As a result, trajectory forecasting from an object-level perspective has been studied in recent years [42], although suffers from a lack of large, high-quality datasets and standardized evaluation protocols.

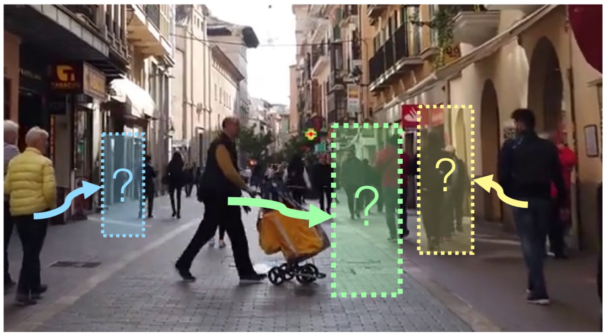


Figure 1: We introduce the new task of multiple object forecasting and the Citywalks dataset to facilitate future research.

Motivated by the above observations, we introduce a new formalization of the trajectory forecasting task: multiple object forecasting (MOF) (Fig. 1). MOF follows the same formulation as the popular multiple object tracking (MOT) task, but rather is concerned with predicting *future* object bounding boxes and tracks in upcoming video frames, rather than the bounding boxes and tracks in the current frame. Future bounding box prediction has previously been studied in constrained settings such as on-board a moving vehicle with odometry information [3, 43]. In contrast, MOF follows the unconstrained MOT setting, which utilizes only image information where data from other sensors is not available. This setup poses several challenges, such as variations in object scale, non-linear motions, and ego-motion. MOF has a number of possible applications such as object tracking [10] (particularly through occlusions), robotic navigation [20], and autonomous driving [17].

To facilitate research on the MOF problem, we construct the Citywalks dataset. Citywalks is a large and diverse dataset collected from a first-person perspective in 21 European cities with considerable variability in many facets such as weather, object appearance, illumination, object scale, and pedestrian density. Citywalks is annotated using automated methods for detection and tracking and is considerably more diverse than existing datasets [29, 34, 28] for trajectory forecasting. We evaluate existing models adapted for MOF on Citywalks and propose a novel encoder-decoder model. Our model, STED, combines visual features extracted from optical flow with temporal features and outperforms existing models on the MOF task.

The contributions of this work are as follows:

1. We introduce MOF, a new formulation of the trajectory forecasting problem (Section 3).
2. We introduce and publicly release Citywalks, a challenging dataset for MOF with considerably more geographical variety than existing datasets (Section 4).
3. We propose STED, a Spatio-Temporal Encoder-Decoder model for MOF which combines visual and temporal features (Section 5). Experimental evaluation using two datasets confirms the benefits of our proposed approach (Section 6).

2. Related work

In this section, we summarize the main contributions in the fields of pedestrian trajectory forecasting and MOT. We also provide an overview of existing datasets for both tasks and their limitations.

2.1. Multiple object tracking

Methods for MOT typically follow a tracking-by-detection paradigm that relies heavily on the accuracy of single-frame detections and models to associate detections across time. Reasonable MOT performance can be obtained with high-quality detections and simple constant velocity motion assumptions [2], and better still when combined with a visual appearance association metric [41]. Constructing more sophisticated methods capable of modeling non-linear motion can improve tracking performance, particularly in scenarios with occlusion [10]. However, trajectory forecasting for improved tracking is challenging due to small datasets, which results in overfitting. One approach proposed to overcome this issue is to consider the future trajectory as a binary classification problem [36] or using explicit external memory to avoid memorization [9]. We adopt a more straightforward approach to address overfitting: building a larger dataset.

2.2. Pedestrian trajectory forecasting

Pedestrian trajectory forecasting has been studied extensively in a surveillance setting from fixed cameras from a birds-eye view [1, 44, 12, 7, 46]. Methods typically focus on interactions between pedestrians and social conventions such as the pioneering Social Long-Short-Term-Memory (Social-LSTM) model [1], in addition to scene semantics. These methods do not typically consider visual cues, and many simplify each pedestrian to a point in space. Recently, Liang et al. [24] proposed one of the first approaches for trajectory forecasting using visual features. Their method encodes appearance using a person keypoint detector and joint modeling of future pedestrian trajectory and activity.

Most related to our paper, a small number of works consider trajectory forecasting from an object-level perspective. Predicting object trajectories from on-board moving vehicles, in particular, has been studied extensively [17, 3, 40]. Methods typically use additional information sources specific to a vehicle setting, such as odometry information. In an inspiring work outside of the vehicle domain, Yagi et al. [42] propose a model that uses past locations, ego-motion, and pedestrian keypoints to estimate future trajectory in first-person videos. Their model outperforms existing state-of-the-art approaches; however, accurate pedestrian keypoint estimation is not always practical, especially in low-resolution or low-lighting scenarios. In contrast, our approach does not rely on pedestrian keypoint estimation.

2.3. Existing datasets

Many large datasets with annotated pedestrian bounding boxes have been released such as Citypersons [47], BDD-100K [45] and EuroCity Persons [4]. However, these datasets do not contain object tacking annotations. Older datasets such are KITTI [11] and Caltech-USA [8] provide full object tracks, although these datasets are considerably smaller with more limited geographical variety than our new dataset.

Several datasets have been created explicitly for pedestrian trajectory forecasting, such as UCY [23], ETH [29], and Stanford Drone [35]. These datasets are recorded from a birds-eye view, making them suitable for modeling social and environmental factors. However, such datasets are not well suited to MOF due to being captured at a perspective from which extracting visual features is challenging.

Few public datasets exist for object-level view trajectory forecasting. Most similar to ours, the MOT-17 dataset [28] contains annotated pedestrian bounding boxes from both first-person and overhead cameras. However, MOT-17 contains only 14 video sequences. Our dataset, Citywalks, contains 358 video sequences.

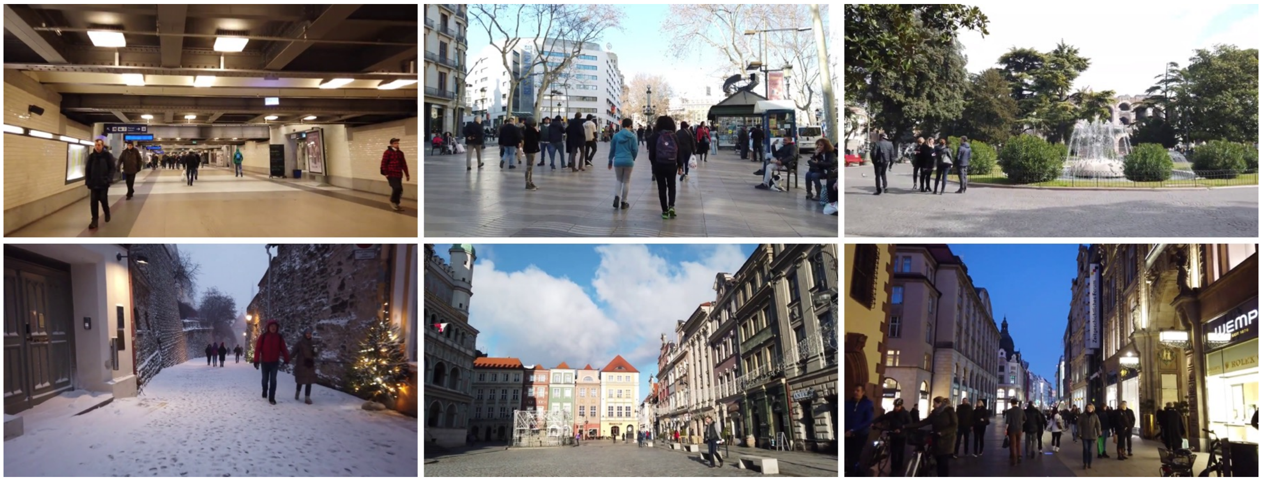


Figure 2: Example frames from the Citywalks dataset. Citywalks is markedly larger and more diverse than existing datasets.

3. Multiple object forecasting

MOF follows a similar problem formulation to the prevalent MOT task. In this section, we formalize MOF and the metrics used for evaluating models.

3.1. Problem formulation

Consider a sequence of n video frames $f_0, f_1, \ldots, f_{n-1}$. Given the t^{th} frame f_t, the task of object detection is to associate each identifiable object $i \in \mathcal{I}$ in the frame with a set of coordinates $b_t^i = (x_t, y_t, w_t, h_t)$ which represent the centroid (x_t, y_t), width, and height of the object bounding box, and $\mathcal{I}$ is the set of all identifiable objects. Given all the framewise detections $\{b_0^i\}, \{b_1^i\}, \ldots, \{b_n^i\}$ for all $i \in \mathcal{I}$, the task of MOT is to associate each detection b_t^i with a unique object identifier $k \in 1, 2 \ldots K$, where K is the total number of unique objects across all frames, such that each object is tracked across the set of n frames.

We extend the MOT task to MOF, shown in Fig. 1. Given $f_{t-p}, f_{t-p+1}, \ldots, f_t$ with associated object detections $\{b_{t-p}^i\}, \{b_{t-p+1}^i\} \ldots \{b_t^i\}$ and tracks, we define MOF as the joint problem of predicting the future bounding boxes $\{b_{t+1}^i\}, \{b_{t+2}^i\}, \ldots, \{b_{t+q}^i\}$ and associated object tracks of the upcoming $f_{t+1}, f_{t+2}, \ldots, f_{t+q}$ video frames for each object present in frame f_t, where p is the number of past frames used as input and q is the number of future frames to be predicted. In this work, we use $p = 30$ and $q = 60$, corresponding to 1 second in the past and 2 seconds into the future at 30Hz.

3.2. Evaluation metrics

We adopt the average displacement error (ADE) and final displacement error (FDE) metrics from the trajectory forecasting literature [1]. ADE is defined as the mean Euclidean distance between predicted and ground-truth bounding box centroids for all predicted bounding boxes, and FDE is defined similarly for the centroid at the final timestep only. We also use the average and final intersection-over-union (AIOU and FIOU) metrics. AIOU is defined as the mean IOU of the predicted and ground truth bounding boxes for all predicted boxes, and FIOU is the IOU for the box at the final timestep only.

4. Citywalks Dataset

Our newly-constructed Citywalks dataset comprises of 358 video sequences containing footage from 21 different cities in 10 European countries.

4.1. Data collection

We extract footage from the online video-sharing site YouTube[1]. Each original video consists of first-person footage recorded using an Osmo Pocket camera with gimbal stabilizer held by a pedestrian walking in one of the many environments for between 50 and 100 minutes. Videos are recorded in a variety of weather conditions, as well as both indoor and outdoor scenes. Example frames showcasing the variety of the dataset are shown in Fig. 2.

4.2. Video clip filtering

One of the fundamental challenges of MOF is the bounding box motion caused by both ego-motion and object motion. Large displacements resulting from significant ego-motion pose a problem and may overwhelm the training process. To mitigate the impact of large ego-motions, we filter the dataset by removing high motion segments. Global motion is estimated by extracting dense optical flow and selecting short video clips from windows with a mean optical

[1] Videos are obtained from `https://www.youtube.com/c/poptravelorg`

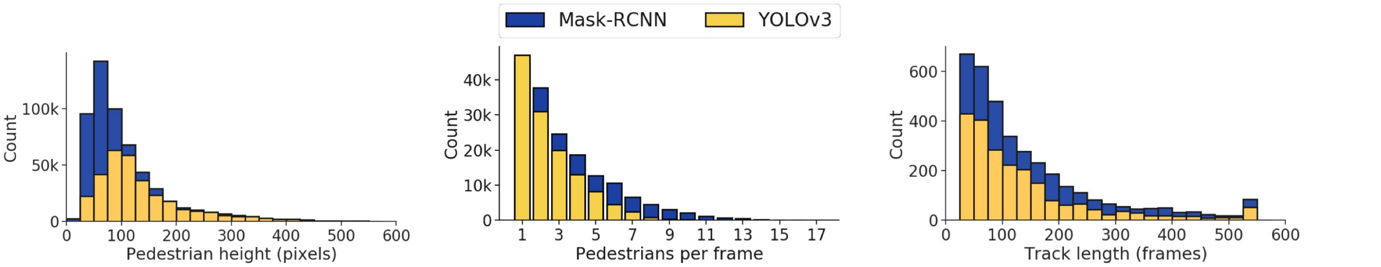


Figure 3: Citywalks annotation statistics.

Table 1: Citywalks metadata.

Video clips	358
Resolution	1280 × 720
Framerate	30hz
Clip length	20 seconds
Unique cities	21
Weather conditions	Sun/Rain/Snow/Overcast
Time of day	Day/Night
Labelled objects per frame	0 - 17
Unique tracks (YOLOv3)	2201
Unique tracks (Mask-RCNN)	3623

flow magnitude below a threshold. Specifically, we downsample video frames to 128×64 pixels for faster computation and extract dense optical flow using FlowNet2-S [15]. We then select 20-second clips from longer videos using segments containing frames that do not exceed a mean optical flow magnitude threshold of 1.5.

4.3. Annotations

Once clips are selected, pedestrians are detected using an object detection algorithm. We provide annotations for two object detectors: YOLOv3 [31] and Mask-RCNN [13]. Both detectors are trained using the MS-COCO [25] dataset and generalize well to Citywalks. For the YOLOv3 annotations, images are downsampled to 416×416 pixels before detection, to simulate detection quality under low processing time requirements. We use a resolution of 1024×1024 for detection using Mask-RCNN to obtain the best detection performance. Note that we leave any attempts to combine the two annotation sets (such as in [37]) for future work. Following the detection phase, pedestrians are tracked using DeepSORT [41], which uses a Kalman filter and person re-identification model to associate detections across frames. We then discard tracks shorter than 3 seconds as the previous 1 second of bounding box data is used to predict the next 2 seconds. Dropping short tracks reduces the number of false positives in the annotation set, as we observe that erroneous tracks typically do not last longer than 3 seconds. Each video clip is also manually annotated with the city of recording, time of day, and weather condition. Annotation statistics are shown in Fig. 3, and metadata are shown in Table 1.

5. Proposed model

In this section, we present STED, an encoder-decoder architecture for MOF that combines visual and temporal features. The proposed architecture has three components: (i) A bounding box feature encoder based on a Gated Recurrent Unit (GRU) [6] that extracts temporal features from past object bounding boxes (ii) A CNN-based encoder that extracts motion features directly from optical flow, and (iii) a decoder implemented as another GRU for generating future bounding box predictions given the learned features. An overview of our model is shown in Fig. 4.

5.1. Bounding box feature encoder

Our bounding box encoder extracts features from past bounding box coordinates of each object i represented in terms of its centroid, width and height $b_t^i = (x_t, y_t, w_t, h_t)$. In addition, we compute the velocity in the x and y directions, (v_t^x, v_t^y), change in width, Δw_t, and change in height, Δh_t. This results in an 8-dimensional vector associated with each object bounding box $B_t^i = (x_t, y_t, w_t, h_t, v_t^x, v_t^y, \Delta w_t, \Delta h_t)$.

For each observed timestep, a GRU (GRU-1 in Fig. 4) takes the vector B_t^i as input and outputs an updated hidden state vector h_t^e. This update is repeated for all timesteps, resulting in a single hidden state vector h_t^e at the final timestep which summarizes the entire sequence of bounding boxes. The 256-dimensional feature vector ϕ_b from a fully connected layer (FC-1 in Fig. 4) is used as a compact representation of the history of bounding boxes.

5.2. Optical flow feature encoder

We adapt Dynamic Trajectory Predictor (DTP) [40] to learn features directly from optical flow. Flow frames, F_t, are extracted from within object bounding boxes obtained using YOLOv3 or Mask-RCNN at each timestep. A stack of 10 frames are sampled uniformly from timesteps $t-29$ to t inclusively, representing 1 second of motion history.

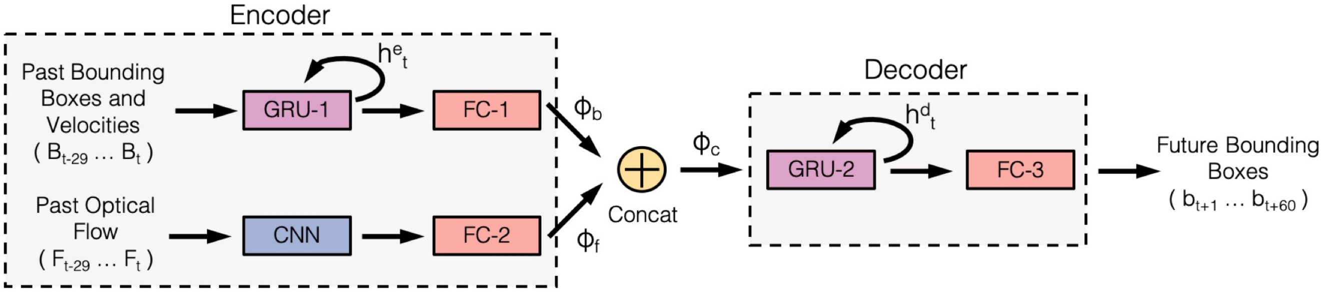

Encoder
Past Bounding Boxes and Velocities (B_{t-29} … B_t)
Past Optical Flow (F_{t-29} … F_t)
GRU-1
h^e_t
FC-1
CNN
FC-2
ϕ_b
ϕ_f
Concat
ϕ_c
Decoder
GRU-2
h^d_t
FC-3
Future Bounding Boxes (b_{t+1} … b_{t+60})

Figure 4: STED consists of a Gated Recurrent Unit (GRU), a Convolutional Neural Network (CNN), and two fully-connected (FC) layers for feature encoding. Our decoder takes the encoded feature vector ϕ_c as input and outputs predicted object bounding boxes for the next 2 seconds using another GRU and FC layer.

The stack of 10 horizontal and 10 vertical frames are used as input to a CNN which takes the $20 \times 224 \times 224$ stack of frames as input and is trained to predict future object bounding boxes. The 2048-dimensional feature vector ϕ_f from the final fully connected layer (FC-2 in Fig. 4) is used as a compact representation of optical flow features. As optical flow captures both object motion and ego-motion, the vector ϕ_f encodes information from these two motion sources. Using optical flow as the input of our encoder rather than features from a person keypoint estimation model [42] avoids the challenges relating to inaccurate keypoint estimations.

5.3. Decoder

Following the feature encoding stage, we use another GRU to generate the estimated sequence of future bounding boxes, enabling the model to generate predictions for an arbitrary number of timesteps into the future. The two feature vectors, ϕ_f and ϕ_b, are concatenated resulting in a single feature vector ϕ_c representing both optical flow and bounding box history. For each future timestep to be predicted, the decoder GRU (GRU-2 in Fig. 4) receives two inputs: The concatenated feature vector ϕ_c, and the internal hidden state h^d_{t-1}. The GRU outputs a new value for h^d_t at each timestep. Given each generated hidden state, a final fully connected layer generates the predicted bounding box for each timestep. Rather than representing object bounding boxes by their absolute location [44] or relative displacement from the previous bounding box [42], we adopt the formulation of [40] and represent the bounding box centroid as the relative change in velocity. The decoder generates a vector $(\Delta v^x, \Delta v^y, \Delta w, \Delta h)$, representing the change in velocity along the x and y-axes, and the change in bounding box width and height. The untrained model is initialized to the case where $\Delta v^x = \Delta v^y = 0$ (constant velocity) and $\Delta w = \Delta h = 0$ (constant scale). This formulation results in a better initialization than absolute or relative locations.

6. Performance evaluation

6.1. Baseline models

We adapt the following models for MOF, which are originally developed for trajectory forecasting. Each model is modified for full bounding box prediction assuming object scale is constant, or by adding additional output channels representing bounding box height and width for the learning-based approaches.

Constant Velocity & Constant Scale (CV-CS): We adopt the simple constant velocity model, which is used widely as a baseline for trajectory forecasting models [1, 42, 12] and as a motion model for MOT [48, 39, 33]. We use the previous 5 frames to compute the velocity, and find that using a constant scale performs better than linearly extrapolating a change in width and height.

Linear Kalman Filter (LKF) [16]: The LKF is a widely-used method for tracking objects and predicting trajectories under noisy conditions. We use an LKF with initial parameters chosen using cross-validation and use the last updated motion value for forecasting. The LKF is one of the most popular motion models for MOT [41, 26, 18].

Future Person Localization (FPL) [42]: We adapt FPL, which uses pedestrian pose extracted using OpenPose [5] and ego-motion estimation using optical flow extracted with FlowNet2 [15].

Dynamic Trajectory Predictor (DTP) [40]: We adapt DTP, which uses a CNN with past optical flow frames as input to predict future bounding boxes.

6.2. Implementation details

Clips from Citywalks are split into 3 folds, and the test set is further divided 50% for validation and 50% for testing for each fold. We use inter-city cross-validation, i.e., footage from cities in the validation/testing sets *do not* appear in the training set. This challenging evaluation setup ensures that pedestrian identities from the training set do not appear at test time, and prevents models from overfit-

Table 2: Results averaged over 3 train-test splits on Citywalks with our two annotation sets using YOLOv3 and Mask-RCNN. DTP and FPL predict object centroids only, so IOU metrics are not applicable.

	YOLOv3				Mask-RCNN			
Model	ADE (↓)	FDE (↓)	AIOU (↑)	FIOU (↑)	ADE (↓)	FDE (↓)	AIOU (↑)	FIOU (↑)
CV-CS	32.9	60.5	51.4	26.7	31.6	57.6	46.0	21.3
LKF [16]	34.3	62.1	49.1	25.5	32.9	59.0	43.9	20.1
DTP [40]	28.7	52.4	–	–	26.7	48.5	–	–
FPL [42]	30.2	53.4	–	–	28.6	49.8	–	–
DTP-MOF	29.0	52.2	54.6	30.8	27.3	49.2	49.6	25.1
FPL-MOF	31.6	55.7	53.0	30.9	29.3	51.0	44.9	22.6
STED	**27.4**	**49.8**	**56.8**	**32.9**	**26.0**	**46.9**	**51.8**	**27.5**

ting to a particular environment.

Bounding box feature encoder. Bounding box vectors B_t^i (defined in Section 5.1) are computed by taking the velocity of the object over the previous 5 timesteps, i.e., $v_t^x = x_t - x_{t-4}$ and $v_t^y = y_t - y_{t-4}$. Our feature encoder consists of a GRU with 512 hidden units which uses B_{t-1}^i and the previous hidden state vector h_{t-1}^e as input and outputs an updated hidden state vector h_t^e. We use GRUs rather than LSTMs as recurrent units in STED as we find the performance is similar while GRUs is less computationally demanding.

Optical flow feature encoder. We compute optical flow for each video frame using FlowNet2 [15]. The flow from within each pedestrian bounding box is then cropped, clipped to a range of -50 to 50, scaled to a fixed size of 256×256, and normalized to a range of 0 to 1. We perform standard data augmentation, taking a random crop of size 224×224 and randomly horizontally flipping frames with probability 0.5 during training. We train the optical flow feature encoder using ResNet50 [14] as the backbone CNN architecture for 10k iterations with a batch size of 64 and learning rate of 1×10^{-5} to predict future object locations as described in [40] and then freeze the weights to use our flow encoder as a fixed feature extractor.

Decoder. As described in Section 5.3, our decoder takes the concatenated feature vector ϕ_c as input. The decoder consists of another GRU with 512 hidden units. For each of the 60 timesteps to be predicted, the decoder takes ϕ_c and previous hidden state h_{t-1}^d and outputs a new hidden state h_t^d. A linear layer takes the hidden state and generates a predicted bounding box for the respective timestep. The optical flow feature encoder is used as a fixed feature extractor, while the bounding box encoder and decoder are trained jointly end-to-end using an initial learning rate of 1×10^{-3}, which is halved every 5 epochs. We use a batch size of 1024 and train the model for 20 epochs. The model is optimized using the smooth $\mathcal{L}_1$ loss, which we find to be more robust to outliers in the training data than the $\mathcal{L}_2$ loss.

Table 3: Ablation study evaluating the bounding box (BB), optical flow (OF) encoders separately. Results are the mean of both annotation sets.

Model	ADE / FDE (↓)	AIOU / FIOU (↑)
BB-encoder	29.6 / 53.2	51.5 / 27.9
OF-encoder	27.5 / 50.0	53.2 / 28.8
Both encoders	**26.7 / 48.4**	**54.3 / 30.2**

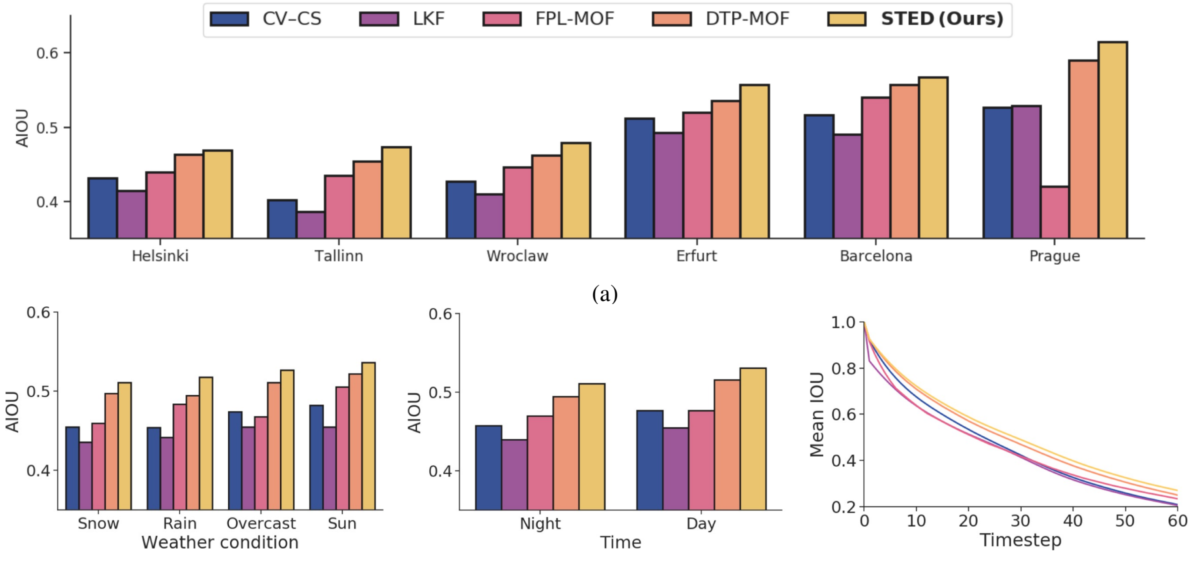
6.3. Results

We evaluate each model on the Citywalks dataset using both annotation sets and evaluate each component of STED separately. Finally, we evaluate the cross-dataset generalizability of each model on the MOT-17 dataset [28].

Results on Citywalks. Table 2 shows the ADE / FDE[2] and AIOU / FIOU of all methods on Citywalks with both annotation sets. We evaluate the original DTP and FPL models for trajectory forecasting, as well as the versions modified for MOF. STED consistently performs better than existing approaches across all metrics, resulting in more precise bounding box forecasts. Fig. 6 shows example bounding box predictions. STED implicitly anticipates both object and ego-motion in a diverse range of environments and situations. Fig. 7 shows failure cases. The model performs poorly in challenging conditions such as large ego-motions and when the pedestrian scale is small.

We further break down performance on Citywalks in Fig. 5. We find that most models perform better for sequences recorded in cities with clear weather conditions (e.g., Barcelona, Prague) than, in particular, snow (e.g., Tallinn, Helsinki). To confirm this intuition, we further plot the performance in different weather conditions and at different times of the day. Finally, we plot the mean IOU at all predicted timesteps 1 to 60. The IOU of the predicted and ground-truth bounding boxes predictably declines quickly, particularly for earlier timesteps. STED maintains the best

[2] A displacement of 50 pixels corresponds to 2.5% of the total frame size at a resolution of 1280×720.



(a)

(b)

(c)

(d)

Figure 5: Performance analysis on Citywalks. Here, we report performance on both validation and test sets for all 3 folds to cover the entire dataset. Performance is broken down by (a) top 3 and bottom 3 cities by AIOU, (b) weather condition, (c) time of day and (d) future timestep.

IOU throughout the full prediction horizon.

Ablation study. We evaluate the benefits of each component of our proposed model by evaluating them separately. Specifically, we use the bounding box encoder feature vector ϕ_b as input to the decoder, rather than the concatenated feature vector ϕ_c. We repeat this for the optical flow encoder feature vector ϕ_f. Table 3 show the results of our ablation study on Citywalks. Both the bounding box and optical flow encoders contribute to the overall performance.

Computational complexity. The most computationally expensive component of STED is computing optical flow. Our implementation uses FlowNet2, which requires 123ms to compute on an Nvidia GTX 1080 GPU [15]. This model may be replaced by more efficient methods, although we found the quality of optical flow to impact overall performance. Additional components, such as the CNN architecture or number of hidden units in the GRUs may be modified if real-time performance is required, at some cost in forecasting accuracy.

Cross-dataset evaluation. In order to evaluate the generalizability of models trained on Citywalks, we use the popular MOT-17 dataset [28]. We use sequences 2, 9, 10, and 11 from the MOT-17 train set and discard sequences 4 and 13 as these sequences are filmed from an overhead perspective. We also discard sequence 5 due to the low image resolution and frame rate. We follow a similar preprocessing setup to Citywalks, discarding tracks shorter than 3 seconds. We also ensure pedestrians are occluded no more than 50% of their total bounding box size using the annotations provided, resulting in 83 unique pedestrian tracks. We take each model trained on Citywalks and evaluate using each of the four sequences. Note that we do not modify the models and crucially we *do not* fine-tune on MOT-17. Table 4 shows encouraging results suggesting that models trained on Citywalks generalize cross-dataset and to human-annotated bounding boxes. However, due to the small size of the MOT-17 dataset, these results should be treated with caution.

Table 4: Results on MOT-17 after training on fold 3 of Citywalks. Models are not fine-tuned on MOT-17.

Model	ADE / FDE (↓)	AIOU / FIOU (↑)
CV-CS	58.9 / 104.7	43.8 / 21.5
LKF [16]	62.0 / 110.2	41.6 / 20.1
FPL [42]	56.9 / 96.3	–
DTP [40]	55.2 / 99.0	–
FPL-MOF	58.0 / 98.4	41.4 / 20.4
DTP-MOF	52.2 / 92.4	**47.7 / 26.1**
STED	**51.8 / 91.6**	46.7 / 24.4

7. Conclusion

We have introduced the task of multiple object forecasting and created the Citywalks dataset to facilitate future research. Crucially, we have shown that models trained on the Citywalks dataset can predict future object bounding

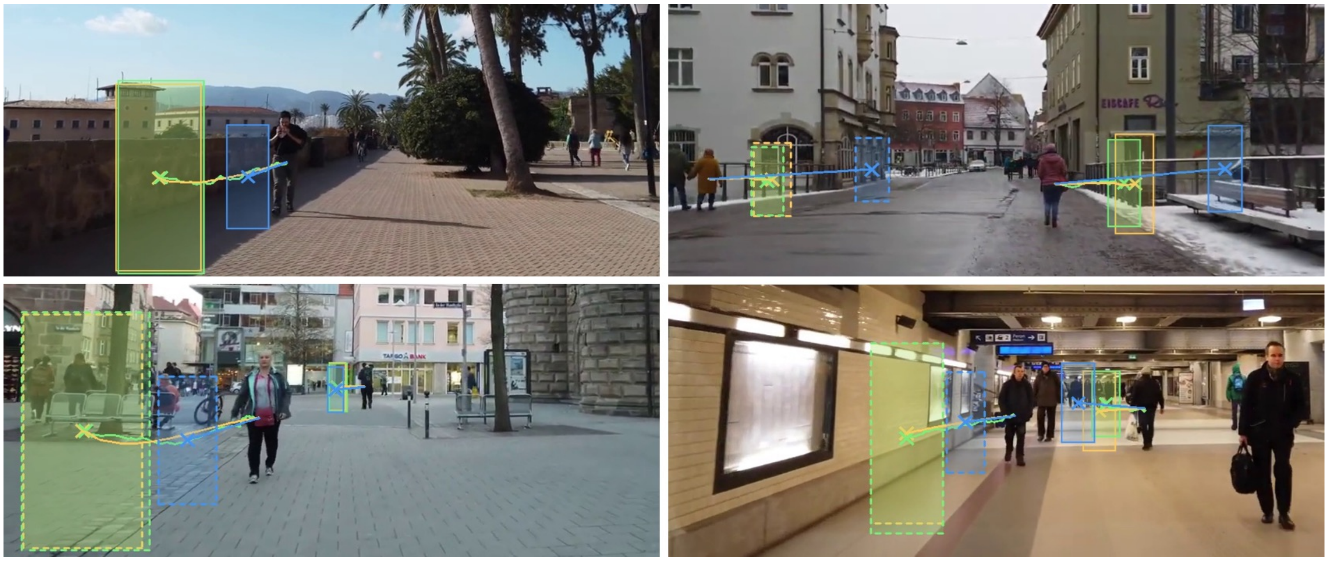


Figure 6: Example successful object forecasts using our proposed model. Colours represent ground truth (**Green**), constant velocity and scale (**Blue**), and STED (**Yellow**). Forecasts are made for each of 60 timesteps in the future for all pedestrians in the scene, but here we visualize the predicted bounding box at $t = 60$ only and at most two pedestrians per frame for clarity. Line type (dashed/solid) denotes unique pedestrians. More example available at: `https://youtu.be/GPdNKE6fq6U`

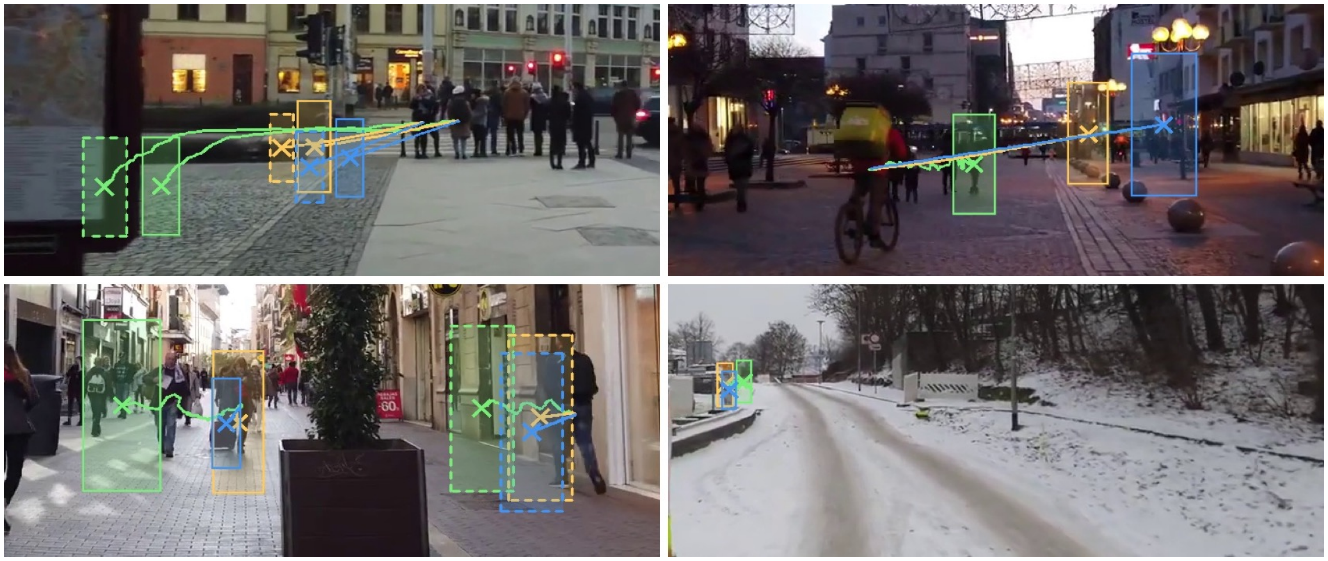


Figure 7: Example unsuccessful object forecasts using our proposed model. Colours represent ground truth (**Green**), constant velocity and scale (**Blue**), and STED (**Yellow**). The examples highlight the difficulty of the Citywalks dataset, which contains several distant pedestrians and non-linear motions.

boxes on the MOT-17 tracking benchmark more precisely than existing methods used by multiple object tracking. Our encoder-decoder model, STED, forecasts object bounding boxes up to 2 seconds in the future and anticipates non-linear motions. This development shows promise for building more sophisticated object forecasting models to aid object tracking in order to address common problems such as occlusions and missed detections.

Acknowledgements

We would like to thank Shanaka Perera and Shuyang Sun for their insightful comments. This work is funded by the UK EPSRC (grant no. EP/L016400/1) and the EU Horizon 2020 project IDENTITY (Project No. 690907). Our thanks to NVIDIA for supporting this research with their generous hardware donation. We would also like to thank Daniel Sczepansky for collecting the videos used in this research.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Computer Vision and Pattern Recognition*, 2016.

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[3] A. Bhattacharyya, M. Fritz, and B. Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Computer Vision and Pattern Recognition*, 2018.

[4] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila. The eurocity persons dataset: A novel benchmark for object detection. *arXiv preprint arXiv:1805.07193*, 2018.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Computer Vision and Pattern Recognition*, 2017.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] C. Choi and B. Dariush. Looking to relations for future trajectory forecast. *arXiv preprint arXiv:1905.08855*, 2019.

[8] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Transactions on Pattern Analysis and Machine Intelligence*, 34(4), 2011.

[9] K. Fang, Y. Xiang, X. Li, and S. Savarese. Recurrent autoregressive networks for online multi-object tracking. In *Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[10] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In *Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[12] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Computer Vision and Pattern Recognition*, 2018.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *International Conference on Computer Vision*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016.

[15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Computer Vision and Pattern Recognition*, 2017.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[17] C. G. Keller and D. M. Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. *Transactions on Intelligent Transport Systems*, 2014.

[18] H. Kieritz, S. Becker, W. Hübner, and M. Arens. Online multi-person tracking using integral channel features. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 122–130. IEEE, 2016.

[19] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *Eurpoean Conference on Computer Vision*, 2012.

[20] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[21] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*. Springer, 2014.

[22] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[23] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, 2007.

[24] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Eurpoean Conference on Computer Vision*, 2014.

[26] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, volume 5, page 8, 2018.

[27] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun. Predicting deeper into the future of semantic segmentation. In *IEEE International Conference on Computer Vision*, pages 648–657, 2017.

[28] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[29] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[30] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *Intelligent Vehicles Symposium*, 2017.

[31] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv:1804.02767*, 2018.

[32] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *Computer Vision and Pattern Recognition workshop*, 2015.

[33] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

[35] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016.

[36] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017.

[37] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[38] S. Schmidt and B. Färber. Pedestrians at the kerb–recognising the action intentions of humans. *Transportation research part F: traffic psychology and behaviour*, 12(4), 2009.

[39] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] O. Styles, A. Ross, and V. Sanchez. Forecasting pedestrian trajectory with machine-annotated training data. In *Intelligent Vehicles Symposium*, 2019.

[41] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[42] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato. Future person localization in first-person videos. In *Computer Vision and Pattern Recognition*, 2018.

[43] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. *arXiv:1809.07408*, 2018.

[44] S. Yi, H. Li, and X. Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *Eurpoean Conference on Computer Vision*, 2016.

[45] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018.

[46] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng. Srlstm: State refinement for lstm towards pedestrian trajectory prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[47] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[48] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.