Towards Robust Human Trajectory Prediction in Raw Videos

Rui Yu and Zihan Zhou*

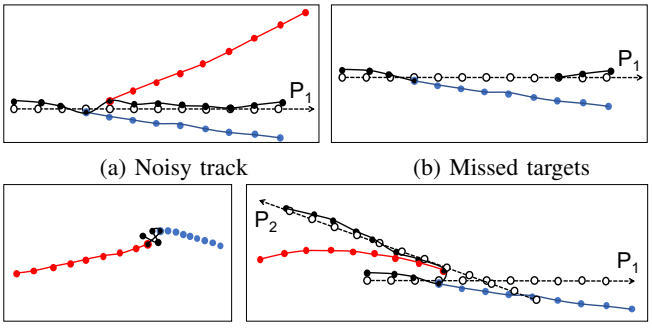
***Abstract*— Human trajectory prediction has received increased attention lately due to its importance in applications such as autonomous vehicles and indoor robots. However, most existing methods make predictions based on human-labeled trajectories and ignore the errors and noises in detection and tracking. In this paper, we study the problem of human trajectory forecasting in raw videos, and show that the prediction accuracy can be severely affected by various types of tracking errors. Accordingly, we propose a simple yet effective strategy to correct the tracking failures by enforcing prediction consistency over time. The proposed "re-tracking" algorithm can be applied to any existing tracking and prediction pipelines. Experiments on public benchmark datasets demonstrate that the proposed method can improve both tracking and prediction performance in challenging real-world scenarios. The code and data are available at https://git.io/retracking-prediction.**

I. Introduction

Driven by emerging applications such as autonomous vehicles, service robots, and advanced surveillance systems, human motion prediction has received increased attention in recent years [1]. In the literature, most studies apply regression models on the subjects' past trajectories to recursively compute the target positions several time steps into the future. Some traditional methods are based on motion models such as linear models, Kalman filters [2], Gaussian process regression models [3], and social force models [4]. Recently, data-driven deep learning models such as Long Short-Term Memory (LSTM) have been shown to achieve higher prediction accuracies, thanks to their ability to modeling complex temporal dependencies and human interactions in the sequential learning problem [5], [6].

In the aforementioned methods, the subjects' past movements, which serve as the input, are assumed to be given. However, in real-world scenarios, the system often needs to first estimate the past trajectories from raw video data. For example, for an autonomous vehicle to safely and efficiently navigate in city traffics, it is necessary to understand and predict the movement of pedestrians from the video stream captured by its on-board cameras. Since most prediction methods do not explicitly consider the errors and uncertainties incurred by detection and tracking, directly applying them often leads to *inconsistent predictions* over time.

In Fig. 1, we illustrate several common cases of inconsistent predictions in raw videos. As seen in Fig. 1(a), the estimated tracks may not strictly follow the ground truth (GT) trajectory of a subject. Because of the apparent change of direction due to the noisy estimation, the predictions at time t could be very different from those at time $t+1$. Besides, the tracking results may contain various types of errors including missed targets, spurious tracks, and ID switches. As a result, the predictions at consecutive time steps (if exist) could differ significantly (Fig. 1(b)-(d)).



(a) Noisy track (b) Missed targets

(c) Spurious track (d) ID switch

Fig. 1. Common causes of inconsistent human trajectory predictions in raw videos. In each figure, we show the ground truth trajectory (black circles), tracker outputs (black dots), predictions at time t (blue dots), and predictions at time $t+1$ (red dots).

In this work, we study the problem of human motion prediction in raw videos. We show empirically that, due to the aforementioned reasons, there is a significant performance gap between prediction in raw videos and that using manually labeled human trajectories, especially when reliable detection and tracking is difficult to obtain (*e.g.*, small objects, crowded scenes, camera movements).

As an attempt to bridge this gap, we propose a simple yet effective strategy to improve the prediction performance in raw videos by enforcing *temporal prediction consistency*, a property largely ignored by prior work. Specifically, given the results obtained by any tracking algorithms, we first apply a smoothing filter to the estimated tracks. Then, we repeatedly run the prediction model at every tracked location, reconstruct a new track for each human subject by comparing the similarity of prediction results for points at consecutive time steps, and generate the final predictions using the new track. The advantage of our "re-tracking" algorithm is three-fold. First, compared to the original tracks, the results obtained by our algorithm have significantly fewer missed targets, spurious tracks, and ID switches. Second, the predicted future trajectories are less sensitive to the noises in the original tracks, thus are more accurate. Third, our "re-tracking" algorithm is independent of the tracking and prediction methods, thus can be applied to any existing tracking-prediction pipeline as a standalone module.

We conduct experiments on popular human motion prediction benchmarks. Since our goal is to predict the movement of *all* pedestrians in the video frame, we systematically

*R. Yu and Z. Zhou are with College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802, USA {rzy54, zuz22}@psu.edu

This work is supported by NIH Award R01LM013330.

evaluate the proposed method against baselines in terms of both tracking and prediction. We show that our method can simultaneously improve the tracking and prediction accuracies by addressing the issues shown in Fig. 1. For example, it can reduce the number of ID switches by more than 65% on the SDD test sets [7].

In summary, the contributions of this work are as follows. (1) We study the problem of human trajectory prediction in raw videos, which has received less attention in the literature so far. We analyze the relationship between tracking and prediction, and illustrate the challenges in achieving temporally consistent predictions for this problem. (2) We propose a "re-tracking" algorithm with the goal of improving temporal prediction consistency. We show that it leads to better tracking and prediction performance on public benchmark datasets.

II. Related Work

A. Human Trajectory Prediction

There is rich literature on understanding and predicting human motion from visual data. We refer readers to [1] for a comprehensive review of existing methods. Below we provide a brief overview of recent data-driven methods which are most relevant to our work.

Inspired by the recurrent neural network (RNN) models for sequence generation [8], [5] first proposed to use the RNN to solve the human trajectory prediction problem. Following their work, various deep networks were developed by integrating techniques such as attention models [9], [10], generative adversarial networks (GAN) [6], pose estimator [11], variational autoencoder (VAE) [12], graph neural networks (GNN) [13], [14], and Transformer networks [15]. These methods represent human subjects as 2D points on the ground plane and only employ static environmental images, thus ignore the temporal context in the raw videos.

Several works use raw video frames for human motion prediction. [16] proposed a multi-task network to jointly predict future paths and activities of the pedestrians from raw videos. But the ground truth bounding boxes for the past time steps are still given. [17] proposed to predict pedestrian locations in first-person videos. In their new dataset, some heuristics are used to choose successfully detected human locations and poses as ground truth. [18] proposed a two-stream framework with RNN and CNN to jointly forecast the ego-motion of vehicles and pedestrians with uncertainty. In both works, detection and tracking errors were treated as a minor nuisance to demonstrate the robustness of the proposed method. Recently, however, it is argued in [19] that extracting human trajectories from raw videos remains a challenging problem. It further leveraged unlabeled videos for training deep prediction models. However, their evaluation is still based on ground truth observations.

B. Joint Tracking and Prediction

Human motion models have also been used to assist tracking. A linear model with constant velocity assumption [20] is by far the most popular model. However, real-world human movement patterns are often complicated. In [21], a non-linear model was used to handle the situation that the targets may move freely. To further consider the interaction among targets, [22], [23] proposed to use the social force model [4]. The most closely related work to ours is [24], which proposed a "tracking-by-prediction" paradigm. It compares short-term predictions (*i.e.*, next two frames) with each detection for data association. In our work, we directly compare long-term predictions using the trajectory-wise Mahalanobis distance for data association, with a focus on generating temporally consistent predictions.

Another recent line of work [25], [26], [27], [28], [29], [30], [31] studies joint 3D detection, tracking, and motion forecasting of vehicles in traffic scenes from 3D LiDAR scans. These methods are similar to ours as they do not rely on past ground truth trajectories for motion forecasting. For prediction evaluation, they use a fixed IoU threshold (*i.e.*, 0.5) to associate detections with ground truth bounding boxes in the current frame, and report the Average Displacement Error (ADE) and Final Displacement Error (FDE) at a fixed recall rate (*e.g.*, 80% as in [30]).

The influence of detection and tracking on motion forecasting is investigated more carefully in [32]. Instead of associating estimated tracks to past GT trajectories, it directly matches predicted trajectories with future GT trajectories, and reports the ADE-over-recall and FDE-over-recall curves. But such measures are incomplete in the sense that it does not consider all types of errors in the pipeline. For example, it is possible to simultaneously achieve high recall and low ADE by generating a large number of hypotheses, but also introducing many false positives (*i.e.*, ghost trajectories).

III. Problem Analysis and Method Overview

As mentioned before, given a raw video, we wish to predict the future movement of all human subjects in the scene. Formally, at any time step t, a person in the scene is represented by his/her xy-coordinates (x_t, y_t) on the ground plane. Following previous work, we formulate the task as a sequence generation problem. Given any time step of interest t, let Ω_t be the set of all human subjects in the video frame I_t. Our goal is to predict their positions for time steps $t + [1 : t_{pred}]$, $\forall s \in \Omega_t$.

In this work, we consider a general two-stage approach to tackle this problem. The first step is to perform *detection and tracking* to obtain a set of object instances Ω'_t in the frame I_t. Each instance $s' \in \Omega'_t$ is associated with a sequence of coordinates representing the subject's past movement $\{(x'_{t-t_{obs}}, y'_{t-t_{obs}}), \ldots, (x'_t, y'_t)\}$. Next, for each instance, a prediction model (*e.g.*, LSTM) takes the movement history as input and generates its future movement predictions. In practice, however, the set of tracked instances Ω'_t suffers from issues including missed targets, spurious tracks, and ID switches. Further, the past trajectories may contain noises and errors. As a result, we often observe inconsistency in the predicted trajectories at consecutive time steps (Fig. 1).

In view of the inconsistent predictions, we ask the following question: *Is it possible to construct a new set of*

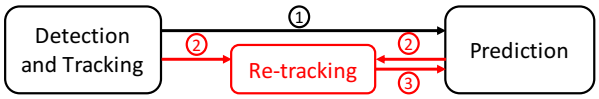


Fig. 2. Proposed pipeline. (1) Given the outputs of an existing tracking method, we predict a subject's future movement at each tracked location. (2) The re-tracking module uses the predictions to build a new set of tracks. (3) With the new tracks, we are able to generate more consistent predictions in raw videos.

tracks with which the prediction model will generate more consistent results? In this work, we show that it is possible, by considering future predictions in tracking. The resulting algorithm is a standalone method that uses a prediction model to improve the estimated tracks (Fig. 2). For any time t, it constructs a new set of object instances Ω''_t, where each $s'' \in \Omega''_t$ is associated with a new sequence of locations. We call it a "re-tracking" method because it operates on, and further refines, the outputs of existing tracking methods. In the next section, we describe our method in detail.

IV. Proposed Re-tracking Method

As input to our method, we assume that we are given a set of trajectories $\{S_1, \ldots, S_N\}$, where $S_k = \{(x_t^k, y_t^k)\}_{t=t_a^k}^{t_b^k}$. Here, t_a^k, t_b^k denote the starting and ending time steps, respectively. We disregard the identity of the trajectory and treat each location (x_t^k, y_t^k) as an independent observation at time t. Let O_t be the set of all observations in time t. Our goal is to associate the observations across different time steps to recover the subjects' trajectories.

To build the new tracks, we explicitly take the prediction performance into account. First, we filter the original tracks to improve the prediction consistency. Second, we compare the predictions made at different time steps and use the differences as a cue to recover missed targets and remove outliers in the tracks. The design of our re-tracking method is based on the following simple ideas:

Smoothing the input sequence: We observe that, at any time t, the most recent relative motion (or instantaneous velocity) $\Delta_t^k = (x_t^k - x_{t-1}^k, y_t^k - y_{t-1}^k)$ is a dominant predictor for a subject's future movement. In [33], a similar observation has also been made, regardless of the prediction model. In practice, this suggests that small perturbations to the subject's estimated location could have a significant impact on the prediction outputs (Fig. 1(a)). Based on this observation, we propose to use Holt–Winters method [34], a classic technique in time series, to smooth the instantaneous velocities in an online manner. The smoothed sequences are then used to predict the subject's future movement.

Repeated predictions: In most prior work on human motion prediction, a subject's trajectory is partitioned into small segments on which the prediction is performed given the ground truth locations for the first t_{obs} time steps. In practice, however, the past t_{obs} locations may not always be available. In this work, we propose to make a prediction from *every* observation (*i.e.*, tracked location) whenever it is possible. Obviously, this will lead to a lot of redundant predictions, but also comes with two benefits: First, it enables prediction of the subject's movement even if detection and tracking fail for some of the time steps, thus improves the prediction recall. Second, using the repeated predictions, we are able to build a new track for each subject that has fewer missed targets and outliers, as we explain next.

Algorithm 1 Re-tracking by Prediction

Input: A set of observations $\{O_t\}_{t=1}^T$; max age t_{max}; matching distance threshold d_{min};
Initialize: $M_a = \emptyset$;
for each frame t **do**
  Perform Hungarian matching: $M_m, M_{um}, O_{um} = Hungarian(M_a, O_t)$;
  for each matched track $m \in M_m$ **do**
    Smooth the associated observation o as in Eq. (3);
    Update P_m with the smoothed observation;
    $a_m \leftarrow 0$;
  end for
  for each unmatched track $m \in M_{um}$ **do**
    $a_m \leftarrow a_m + 1$;
  end for
  for each unmatched observation $o \in O_{um}$ **do**
    Start a new track $m = (P_o, 0)$ and add to M_a;
  end for
  for each track $m \in M_a$ **do**
    if $a_m > t_{max}$ **then**
      Remove m from M_a;
    end if
  end for
  Output: M_a;
end for

Re-tracking by prediction: Since now each observation $o \in O_t$ is associated with a prediction $P_o = \{p_{t+1}^o, \ldots, p_{t+t_{pred}}^o\}$, we can group the observations based on the difference in predictions to re-build the trajectory of each human subject. Unlike most tracking methods which perform data association based on the bounding box distance in the *current frame*, our method uses (long-term) future predictions. This allows us to connect observations across multiple time steps, and remove observations whose predictions are very different from the others (*i.e.*, outliers).

A. Algorithm

Now we describe our re-tracking algorithm in detail. The overall procedure is summarized in Algorithm 1. In the algorithm, we maintain a set of active tracks M_a at each time step. Each track $m \in M_a$ is associated with a prediction P_m and an age a_m and denoted by $m = (P_m, a_m)$.

Distance measure. Given a track m and an observation o, we need to compute the distance between them. To this end, We assume each predicted location in P_o follows a Gaussian distribution: $p_u^o \sim \mathcal{N}(\mu_u^o, \Sigma_u^o), \forall u \in t + [1 : t_{pred}]$. Similarly, for each prediction in P_m we have $p_u^m \sim \mathcal{N}(\mu_u^m, \Sigma_u^m)$. The Mahalanobis distance between two distributions is:

$$d(p_u^m, p_u^o) = \sqrt{(\mu_u^m - \mu_u^o)^T (\Sigma_u^m + \Sigma_u^o)^{-1} (\mu_u^m - \mu_u^o)}. \quad (1)$$

Note that in the above Gaussian distribution, μ_u^o is simply the location predicted by the model, and Σ_u^o is a diagonal

matrix whose entries along the diagonal are equal to $(\sigma_u^o)^2 = (u-t) \times \sigma^2$. Here, σ^2 is a constant, and the coefficient $u-t$ represents our belief that the prediction becomes more and more uncertain into the future. The distribution for p_u^m is defined in a similar way. Then, we define the distance between P_m and P_o as:

$$d(P_m, P_o) = \frac{1}{|T_{m,o}|} \sum_{u \in T_{m,o}} d(p_u^m, p_u^o), \tag{2}$$

where $T_{m,o}$ is the set of overlapping timestamps between the predictions P_m and P_o with $|T_{m,o}| \leq t_{pred} - 1$. Based on the distance measure in Eq. (2), we use Hungarian algorithm [35] for data association between M_a and O_t.

Updating a track. When a track m is associated with a new observation o, we use the observation to update the track m and generate a new prediction P_m. Recall that, because of the noises in the tracking results, directly using the observations as input to the prediction model may produce inconsistent predictions (Fig. 1(a)). Therefore, we apply a smoothing filter to the estimated track. Note that, instead of directly smoothing the observed locations, we smooth the relative motion. This is because the most recent relative motion is shown to be a dominant predictor for future movement [33].

Specifically, let $\{o_{t_a}, o_{t_a+1}, \ldots, o_t\}$ be the sequence of observations associated with m up to time step t. We first compute the relative motion $\Delta_u = o_u - o_{u-1}, \forall u \in [t_a + 1 : t]$. Then, we use the Holt–Winters method (also known as double exponential smoothing) to recursively compute the smoothed motion:

$$\begin{aligned} \Delta'_t &= \alpha \Delta_t + (1-\alpha)(\Delta'_{t-1} + b_{t-1}) \\ b_t &= \beta(\Delta'_t - \Delta'_{t-1}) + (1-\beta) b_{t-1} \end{aligned} \tag{3}$$

Note that a track m may not have an associated observation at every time step. In such cases, we use a simple linear interpolation to recover the full time series. Finally, we use Δ'_t to reconstruct the past trajectory of the subject, which is then used as input to the prediction model to generate P_m.

V. Experiments

A. Experimental Settings

Datasets. We evaluate the proposed method primarily on the *Stanford Drone Dataset* (SDD) [7]. SDD is a widely used benchmark for human trajectory prediction, containing traffic videos captured in bird's-eye view with drones. Following [10], [36], we use the standard data split [37] with 31 videos for training and 17 videos for testing. During testing, we conduct pedestrian detection and tracking at 30 fps on all testing videos ($129,432$ frames in total) except for an extremely unstable one (*Nexus-5* with $1,062$ frames), then evaluate the prediction performance at 2.5 fps to predict 12 future time steps (4.8 sec) as in previous works [10], [37].

Evaluation metrics. To systematically evaluate the performance of the pipeline, we ask the following two questions:

- How many subjects are correctly tracked at any time t?
- For those tracked subjects, what are the differences between the predicted trajectories and ground truth?

For the first question, we employ MOT metrics [38] of Identity F1 score ($IDF1$) and MOT Accuracy ($MOTA$):

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDSW_t)}{\sum_t GT_t}, \tag{4}$$

where FP_t, FN_t, $IDSW_t$, and GT_t represent the number of false positives, false negatives, identity switches, and ground truth annotations at frame t, respectively.

For the second question, we use ADE to evaluate the performance of a prediction method, which is the average mean square error between the ground truth future trajectory and the predicted trajectory. In our problem, however, the inputs are the estimated tracks that we need to match the set of tracked instances Ω'_t with the set of all human subjects Ω_t in the video frame. For a pair of object instances $(s, s'), s \in \Omega_t, s' \in \Omega'_t$, we compute the distance of the pair as:

$$d_{obs}(s, s') = \frac{1}{t_{obs}} \sum_{u=t-t_{obs}+1}^{t} (x_u - x'_u)^2 + (y_u - y'_u)^2. \tag{5}$$

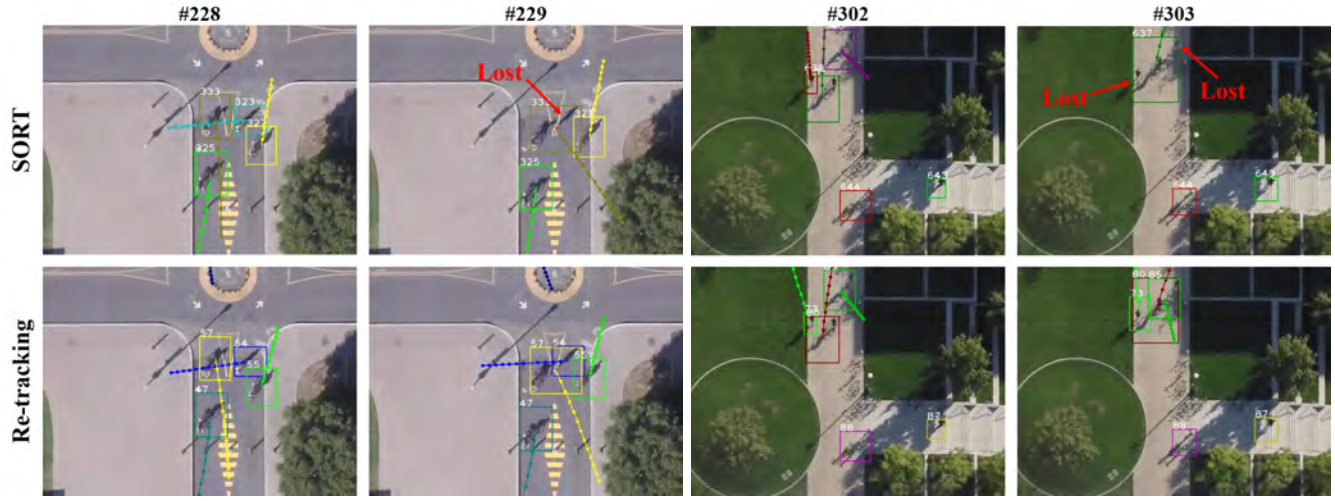
Then, the pairwise distances are fed to the Hungarian algorithm to obtain a one-to-one correspondence between all the tracks and the ground truth. We consider a ground truth subject s correctly matched with s' if their distance is below a threshold τ. Thus, we only compute the ADE on the set of correctly matched subjects M as obtained in the tracking evaluation: $ADE = \frac{1}{|M|} \sum_{(s,s') \in M} d_{pred}(s, s')$, where

$$d_{pred}(s, s') = \frac{1}{t_{pred}} \sum_{u=t+1}^{t+t_{pred}} (x_u - x'_u)^2 + (y_u - y'_u)^2. \tag{6}$$

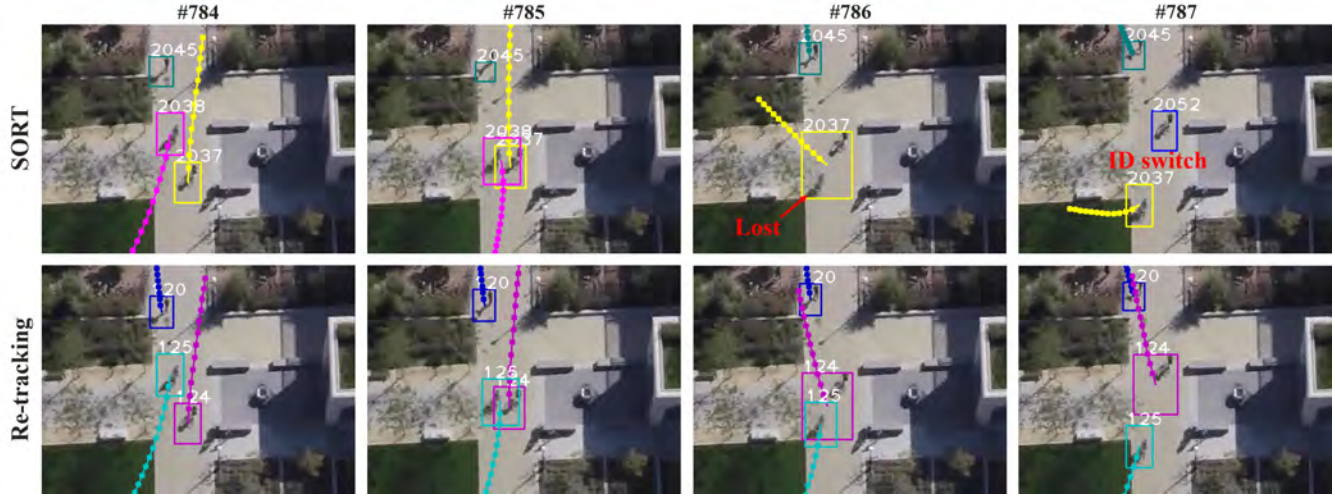
Obviously, the choice of threshold τ has an impact on the prediction evaluation, because the prediction accuracy depends not only on the prediction model, but also on how much the input track deviates from the true location of the subject. By varying the value of τ, we can obtain different association recalls and the corresponding ADE values, and then plot an ADE-over-recall curve.

Baseline and implement details. It is non-trivial to detect small objects from bird's-eye view [19] that the state-of-the-art object detectors [39] failed on SDD. In this study, a motion detector [40] is adopted for SDD but does not perform well in challenging situations such as shaky camera and crowded area. For pedestrian tracking, we utilize a popular tracker SORT [41] with the "tracking-by-detection" framework based on Kalman filter and conduct the proposed re-tracking algorithm upon SORT. For trajectory prediction, we use the ground truth trajectories to train an LSTM encoder-decoder model [5] by using $L2$ loss and Adam optimizer with learning rate 1×10^{-4} for 50 epochs (reduced by a factor of 5 at the 40th epoch). The observation length for prediction is 4 time steps (1.6 sec). The smoothing parameters in Eq. (3) are set to $\alpha = \beta = 0.5$.

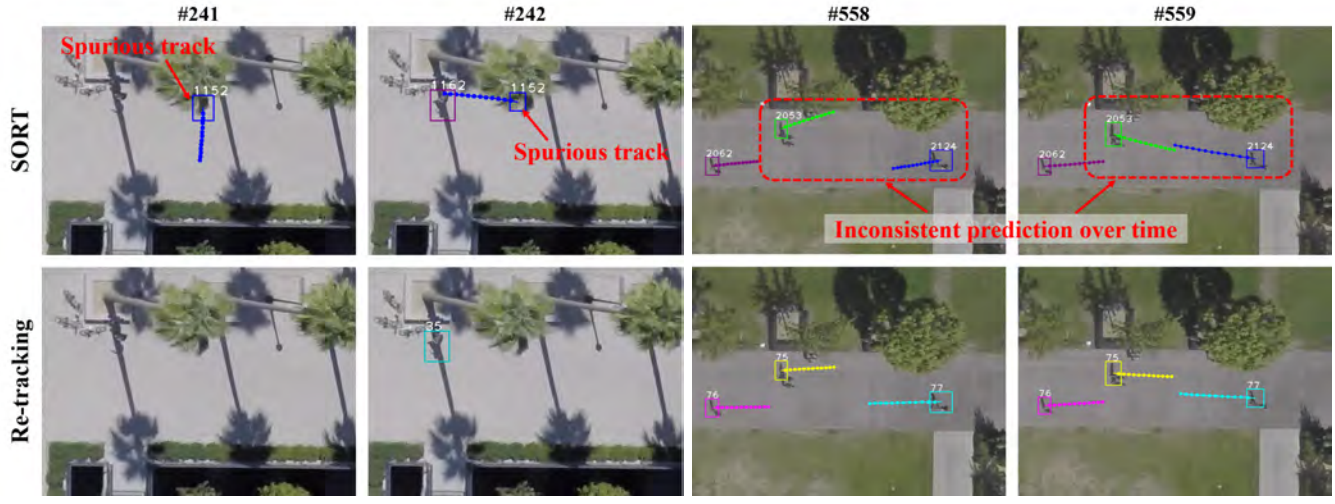
To resolve the differences in scale among different videos, we convert the image coordinates (in pixel) of the center of each tracked bounding box to the world coordinates (in meter) with the given homography matrices and report the tracking and prediction performance in world coordinates.



(a) Scene Little-3 (b) Scene Hyang-0



(c) Scene Hyang-3



(d) Scene Coupa-0 (e) Scene Nexus-6

Fig. 3. Visualization of tracking and prediction at consecutive prediction time steps on SDD dataset. In each subfigure, the first row shows SORT [41] results (baseline); the second row shows the re-tracking results. (a) SORT lost ID323 when the subject walked close to ID333. (b) SORT only maintained ID637 and lost the other two nearby subjects. (c) SORT lost ID2038 and switched to ID2037. (d) SORT generated a spurious track (ID1152) for a swaying tree. (e) Due to camera shake, the predictions based on SORT tracks were inconsistent over time, causing large prediction errors. The re-tracking method was able to handle the aforementioned problems.

B. Case Studies

We first visualize the results on different SDD sequences to analyze the effect of re-tracking. In each sub-figure of Fig. 3, the first row shows the tracking results of SORT at consecutive prediction time steps; the second row shows those of re-tracking. The bounding boxes with ID numbers represent the tracks. Each prediction is denoted by a path with 12 future points. There are some boxes without predictions, due to insufficient number of past observations (< 4). By default, our analysis will use the ID numbers of SORT.

Missed targets. Fig. 3(a) and (b) show examples where the re-tracking method correctly handles the missed targets. In Fig. 3(a), one can see that ID323 was crossing the road at Time #228. At Time #229, ID323 got close to ID333, and the detector only produced one large bounding box for the two persons. SORT allocated the box to ID333, thus lost the track of ID323. In contrast, as seen in the second row, our re-tracking method maintained both IDs at Time #229. This is because the re-tracking method uses long-term predictions for association, thus was able to match track ID323 with a future track (of the same person) as the two people separate.

Fig. 3(b) shows a similar situation of three persons. Since there was only one detection box at Time #303, SORT only kept ID637 and lost the other two nearby subjects. The re-tracking methods maintained all tracks using the prediction-based association.

ID switches. Fig. 3(c) shows an example where the re-tracking method avoids ID switches in the "meet and separate" situation. At Time #784 and #785, ID2037 and ID2038 approached each other. Then at Time #786, the detector only generated one box, and SORT allocated it to ID2037 but lost ID2038. When the pedestrians separated at Time #787, the detector produced two boxes for them again, but SORT associated ID2037 with the wrong one. As a result, the prediction of ID2037 at #787 was very different from those at previous time steps. In contrast, the re-tracking method successfully recovered both tracks, because it preferred similar predictions (*i.e.*, the prediction of ID2037 at #785 and the prediction of ID2052 at #787) during association.

Spurious tracks. The re-tracking method can also eliminate spurious tracks. As seen in Fig. 3(d), the movement of a swaying tree introduced false detections, which were tracked by SORT as ID1152. However, the spurious track led to diverse predictions at Time #241 and #242. In our re-tracking algorithm, the two predictions could not be associated. As a result, the track ID1152 would be divided into short segments and subsequently deleted due to not surviving a probationary period. Note that SORT also utilizes a probationary period. But it relies on the bounding box distances for association, thus was unable to filter such false detections.

Noisy tracks. Finally, Fig. 3(e) shows the effect of re-tracking on resolving the inconsistent predictions due to noisy tracks. In the scenes with even a slight camera shake (*e.g.*, Nexus-6), the predictions from SORT tracks can be quite different at two consecutive time steps. As seen in Fig. 3(e), the predictions of ID2053 and ID2124 at Time #559 are very different from those at #558. By smoothing the history trajectories, the same prediction model can generate more stable predictions from the re-tracking outputs, resulting in smaller prediction errors.

TABLE I

TRACKING PERFORMANCE ON SDD DATASET.

Method	IDF1↑	MOTA ↑	IDSW↓	FP↓	FN↓
SORT	36.0	22.6	3,611	17,517	45,159
Re-tracking	44.9	30.1	1,246	11,441	47,242

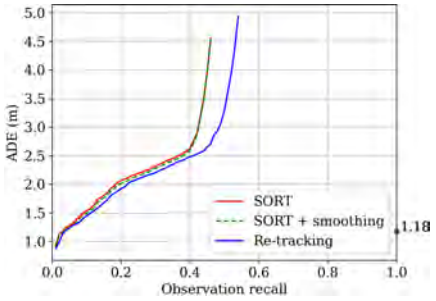


Fig. 4. Comparison of ADE-over-recall curves on SDD testing sets.

C. Quantitative Results on SDD

Tracking results. As discussed before, tracking in SDD videos is difficult because of the small objects, crowd scenes, camera movements, and other factors. As seen in Table I, the baseline method (SORT) yields high IDSW, FP, and FN numbers. For example, shaky cameras bring a lot of false positives, whereas crowded areas lead to false negatives. Compared with SORT, our re-tracking approach increases the overall MOTA by 7.5 points (22.6 to 30.1) and IDF1 by 8.9 points (36.0 to 44.9). Most notably, it reduces IDSW by more than 65% (3,611 to 1,246). The significant improvements in IDF1 and IDSW indicate that our re-tracking method yields more accurate associations, which in turn suggests that the proposed prediction-based distance metric Eq. (2) is more robust in real-world crowded scenes. Additionally, the decrease in FP (17,517 to 11,441) indicates that the re-tracking algorithm can effectively remove spurious tracks.

Prediction results. Fig. 4 shows the ADE-over-recall curves on the SDD testing sets, generated by using different threshold τ to associate the tracked subjects with the ground truth as in Eq. (5). In agreement with the MOT metrics, our re-tracking method can achieve higher recall (53.7%) than SORT (46.0%). Besides, with the same LSTM model, the re-tracking method also yields smaller prediction ADE than SORT under the same observation recall values. Note that the improvements in prediction performance are derived mainly from two aspects: reduced ID switches and smoothed observations. As illustrated in Fig. 3(c), ID switch can induce incorrect predictions, and the prediction model often makes inconsistent predictions over time with noisy tracks, as shown in Fig. 3(e).

As a comparison, we also report the overall ADE for the prediction model given the ground truth observations. As shown in Fig. 4, prediction based on "perfect" tracking (*i.e.*, 100% recall) yields an ADE of 1.18. However, when taking the tracking results as input, the prediction model can achieve

(a) ADE on Nexus-6 (b) Past trajectories at Nexus-6 #559

Fig. 5. Analysis of smoothing effect. (a) Comparison of ADE-over-recall curves on Nexus-6. (b) Visualization of past trajectories (1st row: original SORT; 2nd row: smoothed SORT.)

a comparable ADE only at very low recall levels (*i.e.*, $< 10\%$). This clearly shows that there still exists a large room for future improvements.

Effect of smoothing. As an ablation study, we also evaluate the effect of track smoothing on trajectory prediction. We directly apply Eq. (3) to every SORT track and re-evaluate the prediction performance. The overall ADE-over-recall curve is shown in Fig. 4 with the label "SORT+smoothing". By smoothing the tracks, the ADE drops slightly at different recall levels. The mean decrease in ADE across all recalls is 0.040 with a standard deviation of 0.006.

For individual videos with camera shake (*e.g.*, Nexus-6), smoothing the tracks can significantly improve the prediction performance, as shown in Fig. 5(a). The mean decrease in ADE across all recalls is 0.158 with a standard deviation of 0.020. We visualize the history SORT tracks at prediction time #559 on Nexus-6 in the first row of Fig. 5(b). As analyzed in [33], the prediction of neural networks heavily relies on the most recent two points, which explains the correlation between the SORT tracks in Fig. 5(b) and the inconsistent predictions in Fig. 3(c). The proposed smoothing method can reduce the observation noise level, thus improves the prediction consistency.

Runtime. We conducted experiments on a desktop with an Intel i7-7700 CPU, 32GB RAM, and an Nvidia Titan XP GPU. The entire pipeline (including SORT, re-tracking, and prediction) takes about 0.004s to process a frame, thus is suitable for real-time, online applications.

D. Discussion

We have demonstrated that, by considering prediction consistency during re-tracking, it is possible to achieve better tracking and prediction results. However, as shown in Fig. 4, the ADE increases rapidly as the recall increases, especially when recall > 0.4. A close look at the experiment results reveals that recall $= 0.4$ approximately corresponds to setting the threshold $\tau = 2.0$ for association. For the SDD dataset, if the average distance between two instances is larger than 2.0, it is very unlikely that the two instances belong to the same subject. In other words, the association tends to be wrong, which explains the large errors in the long-term prediction for recall > 0.4. Meanwhile, for threshold $\tau \in [0, 1.5]$ (*i.e.*, mostly correct associations), the ADE roughly falls in the range $[1.0, 2.5]$, which is approximately equal to the observation error $[0, 1.5]$ plus 1.18 (*i.e.*, the ADE obtained using ground truth trajectories).

TABLE II
TRACKING PERFORMANCE ON WILDTRACK DATASET.

Method	IDF1↑	MOTA ↑	IDSW↓	FP↓	FN↓
SORT	41.4	14.1	1,182	12,117	14,454
Re-tracking	43.4	15.0	654	10,713	16,083

Therefore, to further improve the ADE-over-recall curve, the key is to increase the number of correctly tracked subjects. Although our re-tracking method increases the recall from 46.0% to 53.7%, it is still limited by the original detection and tracking algorithms. In particular, the re-tracking method will not recover subjects that are not tracked by SORT for a long period of time. This is evident by the large number of false negatives (FN) in Table I for both SORT and our re-tracking method. Besides, compared to SORT, using re-tracking has two opposite effects on FN: (1) It can reduce FN by recovering missed targets, as shown in Fig. 3(a)-(c). (2) Since re-tracking method relies on future predictions, it will discard short tracks where predictions are unavailable, thus possibly increase FN.

To further improve the recall, one possible direction is to learn a joint model for detection, tracking, and prediction. The intuition is that, future predictions can be used to infer the subject's location during tracking, whereas accurate tracking results can improve the prediction accuracy. Recently, several work proposed to perform such joint inference on point cloud data [25], [26], [27], [28], [29], [30], [31]. However, one challenge in applying similar ideas to datasets like SDD is that state-of-the-art data-driven detectors [39] failed to detect small objects in bird's-eye view. In this work, we resort to a motion-based detector instead. It is interesting to develop a model that leverages the motion cues in videos for joint detection, tracking, and prediction.

Alternatively, one may learn a prediction model that is more robust to tracking errors and noises. For example, since noisy tracks often lead to inconsistent predictions, one could compute the difference in predictions at consecutive time steps and use it as a loss term for training the model.

E. Results on WILDTRACK

As further verification, we also conduct experiments on WILDTRACK [42], a large-scale dataset for multi-camera pedestrian detection, tracking, and trajectory forecasting [43]. It consists of 7 videos (∼35 min) captured by 7 calibrated and synchronized static cameras (60 fps) in a crowded open area in ETH Zurich. The original dataset annotated 7×400 frames at 2 fps. We annotated additional 7×500 frames, resulting in 94,361 annotations in total. In this study, 7×600 frames are used for training and the rest 7×300 for testing. Different from SDD, the WILDTRACK cameras were mounted at the average human height, and we are able to obtain satisfactory pedestrian detection results using a pretrained Mask R-CNN [39] model. We conduct SORT tracking at 10 fps and make predictions with LSTM model at 2 fps for future 9 time steps (4.5 sec) based on 3-step observations (1.5 sec). The main challenge in WILDTRACK is the occlusions among pedestrians in the crowd.

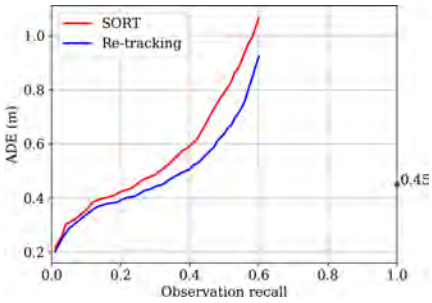


Fig. 6. Comparison of ADE-over-recall curves on WILDTRACK

We report quantitative results on WILDTRACK in Table II and Fig. 6, and refer readers to supplementary video for visualizations. As seen in Table II, re-tracking improves the tracking performance in IDF1 (41.4 to 43.4), MOTA (14.1 to 15.0), IDSW (1,182 to 654), and FP (12,117 to 10,713). The significant drop in IDSW (about 45%) again reflects the strength of our method in creating more accurate associations. According to Section V-D, we only report the ADE-over-recall curve where the threshold $\tau \leq 2.0$ (corresponding to recall ≤ 0.6) in Fig. 6. At every recall level, the LSTM model achieves a lower ADE based on the re-tracking trajectories. The improvement in both tracking and prediction performance on WILDTRACK shows that the re-tracking method is not only effective in bird's-eye videos (*e.g.*, SDD) but also eye-level videos (*e.g.*, WILDTRACK).

VI. Conclusion

In this paper, we study human trajectory forecasting in raw videos. We carefully analyze how false tracks and noisy trajectories affect prediction accuracy. We illustrate the importance of temporal consistency in prediction, and propose a "re-tracking" algorithm to enforce prediction consistency over time. Through case studies, we demonstrate that our re-tracking algorithm can address different types of tracking failures. On the SDD and WILDTRACK benchmark datasets, the proposed method consistently boosts both the tracking and prediction performance. As one of the first attempts to bridge the gap between tracking and prediction, this study leaves several research opportunities for human trajectory prediction in raw videos.

References

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

[2] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. H. Lau, M. C. Lin, and D. Manocha, "BRVO: predicting pedestrian trajectories using velocity-space reasoning," *I. J. Robotics Res.*, vol. 34, no. 2, pp. 201–217, 2015.

[3] D. A. Ellis, E. Sommerlade, and I. D. Reid, "Modelling pedestrian trajectory patterns with Gaussian processes," in *ICCV Workshops*, 2009, pp. 1229–1234.

[4] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, 1995.

[5] A. Alahi, K. Goel, V. Ramanathan, F. Li, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *CVPR*, 2016, pp. 961–971.

[6] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: socially acceptable trajectories with generative adversarial networks," in *CVPR*, 2018, pp. 2255–2264.

[7] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *ECCV*, 2016, pp. 549–565.

[8] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013.

[9] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *ICRA*, 2018, pp. 1–7.

[10] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "SoPhie: An attentive GAN for predicting paths compliant to social and physical constraints," in *CVPR*, 2019, pp. 1349–1358.

[11] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, and M. Cristani, "MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses," in *CVPR*, 2018, pp. 6067–6076.

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

[13] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," in *NeurIPS*, 2019, pp. 137–146.

[14] A. Mohamed, K. Qian, M. Elhoseiny, and C. G. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *CVPR*, 2020, pp. 14 412–14 420.

[15] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *ECCV*, 2020, pp. 507–523.

[16] 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 *CVPR*, 2019, pp. 5725–5734.

[17] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, "Future person localization in first-person videos," in *CVPR*, 2018, pp. 7593–7602.

[18] A. Bhattacharyya, M. Fritz, and B. Schiele, "Long-term on-board prediction of people in traffic scenes under uncertainty," in *CVPR*, 2018, pp. 4194–4202.

[19] Y. Ma, X. Zhu, X. Cheng, R. Yang, J. Liu, and D. Manocha, "AutoTrajectory: Label-free trajectory extraction and prediction from videos using dynamic points," in *ECCV*, 2020, pp. 646–662.

[20] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. J. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *ICCV*, 2009, pp. 1515–1522.

[21] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *CVPR*, 2012, pp. 1918–1925.

[22] S. Pellegrini, A. Ess, K. Schindler, and L. J. V. Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *ICCV*, 2009, pp. 261–268.

[23] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *CVPR*, 2011, pp. 1345–1352.

[24] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Tracking by prediction: A deep generative model for mutli-person localisation and tracking," in *WACV*, 2018, pp. 1122–1132.

[25] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," in *CVPR*, 2018, pp. 3569–3577.

[26] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *CoRL*, 2018, pp. 947–956.

[27] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *ICRA*, 2020, pp. 9491–9497.

[28] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "Pnpnet: End-to-end perception and prediction with tracking in the loop," in *CVPR*, 2020.

[29] S. Casas, C. Gulino, S. Suo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *ECCV*, 2020, pp. 624–641.

[30] L. L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, "End-to-end contextual perception and prediction with interaction transformer," in *IROS*, 2020, pp. 5784–5791.

[31] M. Shah, Z. Huang, A. Laddha, M. Langford, B. Barber, S. Zhang, C. Vallespi-Gonzalez, and R. Urtasun, "LiRaNet: End-to-end trajectory prediction using spatio-temporal radar fusion," *CoRR*, vol. abs/2010.00731, 2020.

[32] X. Weng, J. Wang, S. Levine, K. Kitani, and N. Rhinehart, "Inverting the pose forecasting pipeline with SPF2: Sequential pointcloud forecasting for sequential pose forecasting," *CoRL*, vol. 1, p. 2, 2020.

[33] C. Schöller, V. Aravantinos, F. Lay, and A. C. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, 2020.

[34] C. C. Holt, F. Modigliani, J. F. Muth, H. A. Simon, and P. R. Winters, *Planning Production, Inventories, and Work Force*. Prentice Hall, 1960.

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *IROS*, 2019, pp. 6150–6156.

[37] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, "Trajnet: Towards a benchmark for human trajectory prediction," *arXiv preprint*, 2018.

[38] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017, pp. 2961–2969.

[40] "OpenCV motion detector," https://github.com/methylDragon/opencv-motion-detector, accessed: 2021-02-23.

[41] A. Bewley, Z. Ge, L. Ott, F. T. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016, pp. 3464–3468.

[42] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. M. Bagautdinov, P. Fua, L. V. Gool, and F. Fleuret, "WILDTRACK: A multi-camera HD dataset for dense unscripted pedestrian detection," in *CVPR*, 2018, pp. 5030–5039.

[43] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *CoRR*, vol. abs/2007.03639, 2020.