REWIND TO TRACK: PARALLELIZED APPRENTICESHIP LEARNING WITH BACKWARD TRACKLETS

Jiang Liu[1,2,*], *Jia Chen*[2,*], *De Cheng*[2], *Chenqiang Gao*[1,†], *Alexander G. Hauptmann*[2]

[1] Chongqing Key Laboratory of Signal and Information Processing,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2] Carnegie Mellon University, Pittsburgh, PA 15213, USA

ABSTRACT

Data association, which could be categorized into offline approaches and the online counterparts, is a crucial part of a multi-object tracker in the tracking-by-detection framework. On the one hand, classical offline data association methods exploit all the video data and have high computation cost, which makes them unscalable to long-term offline video data. On the other hand, online approaches have much lower computation cost, but they suffer from ID-switches and tracklet drifting problem when directly applied to offline data as they are only aware of "past" observations. In this paper, we propose a mixed style tracker, which is not only as efficient as the online tracker but also aware of "future" observations in offline setting. We start from a Markov Decision Process (MDP) online tracker and design a parallelized apprenticeship learning algorithm to learn both the reward function and transition policy in MDP. By proposing a *rewind to track* strategy to generate backward tracklets, future detections in offline data are efficiently utilized to obtain a more stable similarity measurement for association. Experiment results show that our approach achieves the state-of-the-art performance on challenging datasets.

Index Terms— multi-object tracking, data association, apprenticeship learning

1. INTRODUCTION

Multiple Object Tracking (MOT) in videos is one of the key components in multimedia content analysis field and it has various applications in different scenarios, including activity recognition, autonomous driving and intelligent surveillance systems. Most state-of-the-art algorithms[1, 2, 3, 4] pursue the *tracking-by-detection* approach as the detector of certain object such as pedestrian starts to work reliably well. In this kind of approach, the inputs are the bounding boxes in each frame and the core problem is called *data association* that links bounding boxes among different frames to form trajectories.

The data association could be further categorized into *online* and *offline* styles according to the survey paper[5]:
Online: The online style data association handles the image sequences in a step-by-step way, i.e., it associates object detections from current frame only with previously obtained tracklets. Therefore, it is natural for the *online style trackers* to handle *online style data*, in which frames and detections are inputted in a stream manner. Popular online approaches usually employ probabilistic inference [6, 7] or determinative models[8, 9] for association. The efficiency of these methods leads the online trackers scalable to real-time and long-term vision tracking tasks, such as autonomous driving.
Offline: The *offline style trackers* strictly require *offline data*, i.e., all frames and detections as inputs. By exploiting observations from all frames, offline trackers are more robust and stable. During association, the object detections are usually represented as nodes in the graph and the edges exhibit potential links among the detections. Hereby, it could be further formulated as a maximum flow[10] or minimum cost problem[11]. The association can usually be solved globally via Dynamic Programming, in which the time complexity increases exponentially with the number of object detections[12]. Thus, it is unscalable to directly apply offline tracker to long-term videos. Other methods[13, 14] seek hierarchy solution by merging tracklets from divided video clips to lower the computation cost. However, the association error in each level of tracklet is also piled hierarchically along with the merging operation.

Based on the above advantages and disadvantages of online and offline style association, an ideal tracker applicable to long-term offline videos should be both scalable as an online tracker and stable as an offline one. It is tentative to think that directly applying online tracker to offline data will work. Unfortunately, without the clue from future frames, the accumulated tracking error usually results in tracklet drifting and ID-switches problem. To address the contradiction, we propose a mixed style tracker by exploiting *offline data* (future observations) in an improved *online tracker*[15]. We name it as *rewind to track*, since we utilize the future detections by generating backward tracklets from rewinding the video.

∗: Contributes equally. †: Corresponding author.

978-1-5090-6067-2/17/$31.00

We then employ an online tracker to track in forward order and use generated backward tracklets for similarity measurement. Compared with a single detection, a backward tracklet forming from a series of detections in temporal domain, is more robust and stable for data association. We view each object in the video sequence as an agent following the Markov Decision Process (MDP) consisting of several states such as *lost* and *tracked*. Different from reinforcement learning in which policy is learned and the reward function is given, we are only given the ground truth behavior sequence, i.e., the ground truth trajectories while neither policy nor reward function is known. This turns out to be an apprenticeship learning (AL) problem[16], which could be decomposed to iteratively solve reinforcement learning (RL): known reward function but unknown policy, and inverse reinforcement learning (IRL): known policy but unknown reward function. That is, each iteration involves two phases: RL phase and IRL phase.

A typical learning algorithm[16] for AL is designed for single agent situation. However, in MOT settings, there exists multiple agents representing different objects. The *polling variant of AL*[15] expands typical AL to the multiple scenario by sequentially polling each object in the learning process. Specifically, they use the policy learned from the j-th object o_j in previous RL phase to learn the reward function for the next object o_{j+1} in the polling sequence. Their underlying implicit assumption is that behavior sequences of neighboring objects o_j and o_{j+1} should be similar. However, the trajectories of different objects vary a lot as the real world situation is quite complex. For example, some easy object trajectories involve no occlusion and appearance change while the difficult cases suffer from drastic changes. To tackle the issue of polling variant of AL, we propose a *parallel variant of AL*. We parallelize each phase of AL across all objects to get a more robust and stable update of both the reward function and policy. In summary, our paper have the following contributions:

- We propose a mixed style tracker, which works efficiently as an online tracker while also as robust as an offline tracker.
- By rewinding the video sequence, we utilize backward tracklets in similarity measurement for association.
- A parallel variant of apprenticeship learning algorithm is proposed to efficiently learn both the reward function and policy for MDP.
- Experiment results reveal that our method achieves the state-of-the-art performance on challenging datasets.

2. PROBLEM FORMULATION

2.1. Mixed style tracker

We give the formulation of our mixed style tracker considering the data feed and tracklet expanding strategy. As shown in Fig.1, given the observations (the dots) of all frames in an offline video, the mixed style tracker follows the way of an online tracker to gradually extend existing trajectories (represented by colored dots linked with solid line) with current detections (hollowed dots) frame by frame. With formulated

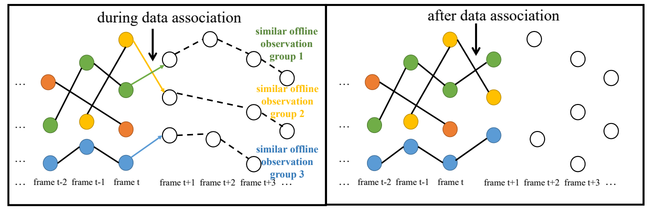

Fig. 1. The data association process of a mixed style tracker.

tracklets in frame t, the tracker needs to determine which detection in frame $t+1$ to associate for each tracklet. An online tracker directly compares tracklet and individual detections in $t + 1$ frame to obtain the result. While our mixed style framework also takes cues from observations after frame $t+1$ to construct similar detection groups (dots linked by dashed line). For a particular detection in frame $t+1$, its corresponding offline detection group is utilized for similarity measurement during association. Later, after similarity comparison, only the detections in frame $t + 1$ are merged into the trajectories.

The advantages of our mixed style tracker lie in three aspects. Firstly, since the final trajectories are still formulated in an online manner, the efficiency and scalability is preserved. Secondly, rather than merging tracklets globally as used in hierarchy offline solutions, the offline data is only used for similarity measurement between a particular detection and a tracklet. Therefore, the stability for data association is improved, since we utilize multiple detections in time domain to measure the similarity. Last but not least, the error association would not be accumulated hierarchically. The mixed style tracker could correct it when associating detection in following frames.

2.2. Multi-object tracking based on MDP

In our mixed style tracker, the lifetime of each object is modeled by a finite Markov Decision Process (MDP) which consists of four components $(s, a, \pi, R(s, a))$. $s \in S$: the states of each object at a particular time, representing the finite statuses of an ojbect determined by its previous trajectory. $a \in A$: action taken to transfer the state of a object. The policy π, defines a mapping from state space S to action space A by maximize the reward function $R(s, a)$. Specifically, as shown in Fig.2, an object is categorized into four states in each frame:
active: A newly detected object is initialized as *active*. Then it enters into *inactive* or *tracked* based on whether it is a valid detection.
tracked: An object could keep *tracked*, if and only if its historic tracklet could be extended to current frame. Otherwise it would be transfered to the *lost* via a_4.
lost: When an object is *lost*, it chooses its next state with three

options:1) via a_5: transits back to *tracked* by associating itself with detections in current frame; 2) via a_6: keeps the *lost* state; 3)via a_7: transfers to *inactive* state when lost for a long time. Since other objects in *lost* also need to be evaluated when making this decision, policy in this state is actually equivalent to data association.
inactive: An invalid or permanently lost object enters into the state.

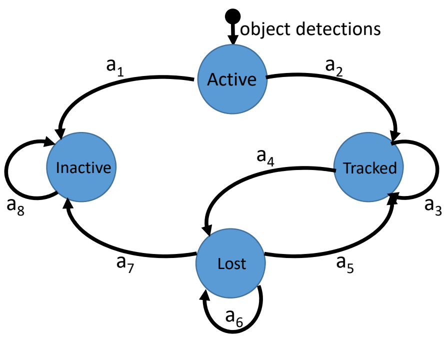

Fig. 2. The MDP state map for mixed style tracker.

Object in each state could be represented by a feature vector $\phi(s)$. The reward function is a linear mapping of the feature: $R(s, a) = w \cdot \phi(s)$. In MOT setting, the ϕ could be a vector which encodes the appearance and motion information of an object. The expected value $V^{\pi}(s_{t_0})$ of a transition policy $\pi(s) \to a$ at time t_0 is evaluated by its afterwards reward expectation:

$$E[V^{\pi}(s_{t_0})] = w \cdot \mu(\pi) \tag{1}$$
$$\mu(\pi) = E[\Sigma_{t=t_0}^{\infty} \gamma^t \phi(s_t)|\pi], \tag{2}$$

where γ is a discount factor with $0 \leq \gamma \leq 1$. Nonetheless, in our mixed style tracker, both the policy π and reward function parameter w is unknown, while only ground truth trajectories are provided in the training set. In other words, we are given the expert's behavior sequence of each object: $D = \{s_{t_1}, a_{t_1}, s_{t_2}, a_{t_2}, \ldots, s_{t_n}, a_{t_n}\}$. We need to find an optimal reward function $\tilde{R}(s, a)$ and policy $\tilde{\pi}$ which best approximates the expected policy value $\tilde{E}[V^{\pi}(s_{t_i})]$ to the ground truth $E^*[V^{\pi}(s_{t_i})]$ at each time t_i. Compared to reinforcement learning (RL) and inverse reinforcement learning (IRL) where either reward function or policy is known, our problem belongs to apprenticeship learning (AL) [16].

3. SOLUTION

In this section, we first describe our method utilizing offline data by generating backward tracklets, named as *rewind to track*. Then a parallel variant of AL is introduced to learn the reward functions and policies in MDP. Feature representation and implementation details are provided in Section 3.3, 3.4.

3.1. Future tracklet generation via rewinding

In order to obtain the similar observation groups in Fig 1, we rewind the video sequence to generate "future" tracklets via time efficient tracking algorithms. Such tracklets are actually backward tracklets, which are complementary to the forward ones. The association error usually happens when occlusion occurring among similar detections belong to different object in a single frame. However, when we tracking from backward, the occlusion problem may be resolved since two targets usually do not occlude each other to an exact same ending frame. Thus, some occlusions are easier to handle in one direction than the other. Then we starts our mixed style tracker in forward order from the first frame. We use a linear mapping function to measure the similarity between a tracklet and a single object detection. The similarity of the i-th object o_i^t and the k-th detection in the t-th frame is defined as follows:

$$sim(o_i^t, d_k^t) = \Sigma_{q=0}^{t'} w^T \phi(o_i^t, d_k^{t,q}) + b, \tag{3}$$

where $d_k^{t,q}$ are sampled detections in frame $t + q$ on the backward tracklet that d_k^t belongs to. In other words, instead of calculating the similarity purely based on an individual tracklet-detection pair (o_i^t, d_k^t), we utilize t' additional object detections on the backward tracklet of d_k^t for measurement. Then the sum along all tracklet-detection pairs' similarity is employed for mixed style association.

3.2. Parallel apprenticeship learning for MDP tracker

The apprenticeship learning algorithm designed for a single agent[16] begins by randomly picking up a policy $\pi^{(0)}$ and computing the related feature expectation $\mu(\pi^{(0)})$. Then the parameters $w^{(p)}$ of reward function and the optimal policy $\pi^{(p)}$ are iteratively learned until convergence, where p is the index of iteration.

However, there are more than one expert behavior sequence D defining the ground truth trajectory for each object in MOT task. a polling variant of AL[15] is employed by conducting iterative AL algorithm for each individual object sequentially. To be specific, the estimated reward function is updated from a particular object o_j independently at one time. Later the new policy $\pi^{(p)}$ is recomputed and applied to the next object o_{j+1} in the polling sequence in order to obtain its feature expectation $\mu_j(\pi^{(p)})$. Assuming that a specific training video sequence has K frames and N ground truth object to track, the video must be processed N times for each object in one iteration. The tracklet features need to be calculated for $N \times K$ times as well, resulting in high computation cost. In addition, since the expert's behavior for each object are learned via polling, the updated reward function and policy from o_j may not be generalizable to o_{j+1}. Therefore, if the behavior of neighboring objects fluctuates a lot, it is very difficult for the polling algorithm to converge to a good solution.

We propose a parallel variant of AL to address the above issues by simultaneously considering all agents' behavior. We minimize the sum of the differences between estimated feature expectation μ_j and the expert's counterpart $\mu_{E,j}$ as following:

$$w^{(p)} = \underset{w}{\operatorname{argmax}} \min \Sigma_{j=1}^{N} w^T(\mu_j(\pi^{(p-1)}) - \mu_{E,j}(\pi^{(p-1)})) \tag{4}$$

$$\pi^{(p)} = \underset{\pi}{\operatorname{argmax}} \Sigma_{j=1}^{N} E[V_j^{\pi}(s_t)]\Big|_{\tilde{R^{(p)}}(s,a)=w^{(p)}\cdot\phi(s)} \tag{5}$$

$$\forall j, \mu_j^{(p)} = \mu_j(\pi^{(p)}). \tag{6}$$

The optimization for $w^{(p)}$ in Eq.4 is actually equivalent to find the maximum margin hyperplane to separate the set of points from expert's expectation and that of k objects simultaneously. This could be obtained via the solver for SVM classifier efficiently. By employing our parallel apprenticeship learning algorithm, a batch of features expectation from different objects are utilized when optimizing for $w^{(p)}$. Besides, the video only needs to be processed once since the optimal policy $\pi^{(p)}$ is also obtained by maximizing the sum of value expectation of all agents in Eq.5.

We start from the *lost* state to explain the detailed implementation of our parallelized AL algorithm. Denote the j-th specific object in i-th video sequence as $o_{i,j}$. It enters the t-th frame in *lost* state with a historic tracklet $o_{i,j}^t$. Assuming d_k^t is the k-th object detection in this frame. The policy in *lost* state should predict a binary label y of the tracklet-detection pair $(o_{i,j}^t, d_k^t)$ to determine whether they should be linked (with $y = +1$, a_5 is taken) or not (with $y = -1$, a_6 is taken). Therefore, we could define reward function in *lost* state $R_{lost}(s, a)$ as following:

$$R_{lost}(s,a)\big|_{o_{i,j}^t} = y(a)\left[\max_{1\leq k\leq N_t}\left(w_{lost}^T\phi(o_{i,j}^t, d_k^t) + b_{lost}\right)\right], \tag{7}$$

where N_t is the number of detections in the t-th frame.

During the p-th iteration of the training for the *lost* state policy, a ground truth tracklet-detection pair $(o_{i,j}^t, d_k^t)$ is added to the training batch S as a positive sample when the previous policy $\pi^{(p-1)}$ misses the association. On the other hand, a negative training sample is added to S when $o_{i,j}^t$ is erroneously linked to a wrong detection d_k^t. Via *rewind to track* strategy, t' extra tracklet-detection couples from the afterward $t+1$ to $t+t'$ frames are also added to S. Finally, our detailed parallelized learning algorithm for *lost* state data association is illustrated in Algorithm 1. For each object in the training sequence, we initialize it at the last frame it fails with policy $\pi^{(p-1)}$. An object is kept in *tracked* as long as its Forward-Backward (FB) error is smaller than a threshold according to the TLD tracker assumption used in [15]. Otherwise, the object is transited to *lost*. When an object is *lost* for a constant frames, the policy will transit it to *inactive*. The updating is conduced when all objects fail in tracking or the sequence is processed to the last frame.

Algorithm 1 Parallelized apprenticeship learning for *lost* state with backward tracklets utilization.

Input: Video sequences $V = \{v_i\}_{i=1}^{N}$, ground truth trajectories $O_i = \{o_{i,j}\}_{j=1}^{N_i}$ and object detections $D_i = \{d_j^i\}_{j=1}^{N_i'}$;
Output: reward function parameters (w_{lost},b_{lost}) for lost status data association;
Initialization of reward function: $w_{lost}^0 \leftarrow w_0, b_{lost}^0 \leftarrow b_0, S \leftarrow \emptyset$
Initialization for each target $o_{i,j}$ in each v_i: set MDP of $o_{i,j}$ in *tracked* after $t_{start}(i,j) \leftarrow$ index of the first frame where $o_{i,j}$ correctly detected
$p \leftarrow 0$
repeat
  $p \leftarrow p + 1$
  for each video v_i in V **do**
    $t \leftarrow 1$
    while $t \leq$ last frame of v_i **do**
      for target $o_{i,j}$ in v_i which $t_{start}(i,j) \geq t$ **do**
        Follow policy π^{p-1}, compute $\mu_{i,j}^{\pi}$ as Eq.6, choose action a
        Compute ground truth action: a_{gt}
        if state is *lost* and $a \neq a_{gt}$ **then**
          $S \leftarrow S \cup \{\phi(o_{i,j}^t, d_k^t), y_k\}$
          $S \leftarrow S \cup \{\phi(o_{i,j}^t, d_k^{t,q}); y_k\}, 1 \leq q \leq t'$
          Save failure position: $t_{start}(i,j) \leftarrow t$
        else
          State transfer: Execute action a
        end if
        If all targets failed **then** break;
      end for
    end while
  end for
  Obtain new reward function parameters (w_{lost}^p, b_{lost}^p): solve Eq.4 with S
  Obtain new policy π^p: solve Eq.5 with (w_{lost}^p, b_{lost}^p)
until all targets are successfully tracked.

3.3. Feature representation

In addition to encode the low-level features including FB error, distance, overlap, etc. In $\phi(o_{i,j}^t, d_k^t)$ as used by [15][1], we also employ person re-identification (ReID) features[17] as high-level appearance representation. The motivation is that finding the optimal detection to link is similar to retrieve the nearest correspondence in the detection gallery with historic tracklet as input. Since the ReID feature is designed to discriminate person from different camera views, it is intrinsically robust to occlusion, pose and illumination changes happened in tracking videos. Our 256D ReID features are extracted from the fully connected layer of the domain guided dropout CNN network[18]. Then the cosine similarity is employed to measure the correlation between the appearance of trajectory $o_{i,j}^t$ and the detection d_k^t. The policy for *active* state is learned on the training set with ground truth bounding boxes via a linear SVM classifier. The 2D positions, width, height and detection scores are normalized as a 5D feature.

3.4. Implementation details

In each MDP process training iteration for the reward parameters $w^{(p)}$, we down sample the positive samples with a ratio of 0.5, so that the numbers of positive and negative samples are balanced. The number of detections from backward track-

[1]The full list of low-level features is: FB error, Normalized Correlation Coefficients (NCC), height ratio, overlap between predicted and actual bounding box, normalized detection score, distance between object and detection.

lets utilized in *rewind to track* t' is 10 for both training and test stages. In the test stage, *rewind to track* firstly uses an online MDP tracker to generate backward tracklets. Then our mixed style tracker is established for each object following the learned policy. The number of templates used in the TLD model is empirically set at 10 for all targets. A new associated bounding box is added to the template when the object returns to *tracked* from the *lost* state. The similarity measurement between tracklet and detection is calculated via Eq.3. Later, the Hungarian algorithm is used for data association among *lost* state objects and detections in each frame.

4. EXPERIMENTS

4.1. Experiment setup

Our mixed style tracker with parallelized AL algorithm is assessed on Multiple Object Tracking (MOT) Challenge 2015[19]. It provides a large collection of datasets in the multi-object tracking community with a common evaluation metric for performance comparison. Both of the training set and the test set contains 11 sequences. Since there is a time limitation to submit results of test set to the online evaluation protocol. We also follow the splitting strategy used in [15] to separate a validation set including 6 sequences to evaluate the impact of each component in our mixed style tracker. For all evaluations, we employ the provided object detections from the aggregated channel features (ACF) detector. The CLEAR MOT metric is used to evaluate the performance, including Multiple Object Tracking Accuracy (MOTA), measuring the tracker performance; Multiple Object Tracking Precision (MOTP), measuring the object detection performance; Mostly Tracked trajectories (MT), trajectories which are Partially Tracked (PT), Mostly Lost trajectories (ML); number of ID Switches (IDS).

Table 1. Evaluation results on the 6 validation sequences of MOT Challenge 2015 dataset.

Dataset	Method	MOTA	MOTP	MT	PT	ML	IDS
TUD-Campus	OnlineMDP[15]	51.53	72.02	1	7	0	13
	AL-poll-ReID	54.92	**72.68**	3	5	0	6
	AL-parallel-online	55.71	72.36	3	4	1	11
	AL-parallel-mixed	**57.61**	71.55	**3**	5	**0**	**5**
ETH-Sunnyday	OnlineMDP[15]	35.79	**77.38**	5	13	12	59
	AL-poll-ReID	47.69	76.67	8	12	10	33
	AL-parallel-online	49.09	76.34	5	13	13	17
	AL-parallel-mixed	**51.08**	76.67	**8**	12	**10**	**16**
ETH-Pedcross2	OnlineMDP[15]	9.13	71.98	2	24	107	80
	AL-poll-ReID	11.34	71.26	4	31	98	79
	AL-parallel-online	12.22	71.52	3	23	107	67
	AL-parallel-mixed	**13.40**	**72.51**	**5**	30	**97**	**64**
ADL-Rundle-8	OnlineMDP[15]	**19.49**	72.74	6	13	9	**28**
	AL-poll-ReID	14.82	72.58	5	14	9	44
	AL-parallel-online	15.18	72.08	6	14	8	114
	AL-parallel-mixed	16.03	**72.75**	**6**	13	**9**	42
Venice-2	OnlineMDP[15]	32.21	74.15	6	15	5	50
	AL-poll-ReID	31.17	**74.59**	4	17	5	38
	AL-parallel-online	33.19	74.06	6	14	6	40
	AL-parallel-mixed	**34.90**	74.39	**7**	15	**4**	**37**
KITTI-17	OnlineMDP[15]	62.23	72.00	1	8	0	2
	AL-poll-ReID	62.87	71.67	1	8	0	3
	AL-parallel-online	62.91	71.78	1	8	0	3
	AL-parallel-mixed	**63.91**	**72.78**	**2**	6	**0**	**1**

4.2. Contribution of each component

We evaluate the contribution from each component of our mixed style tracker to the tracking performance separately on the validation set in Table 1:
OnlineMDP: original online MDP-based tracker trained via polling variant of AL[15].
AL-poll-ReID: adding person ReID feature to OnlineMDP.
AL-parallel-online: replacing the poll variant of AL by parallel variant of AL in AL-poll-ReID.
AL-parallel-mixed: replacing the online style tracker with our mixed counterpart.
Comparing AL-poll-ReID to OnlineMDP, we see that using person ReID features could improve the general tracking performance (MOTA) on most sequences. Parallel variant of AL (AL-parallel-online) does help to learn a better solution compared to the polling variant of AL (AL-poll-ReID) and improves the performance by 0.91%. Our full model, AL-parallel-mixed, achieves the best performance on almost all datasets with more tracked objects (MT, PT) and much fewer ID-switches (IDS). This implies that employing backward tracklets during data association could significant improve the tracking stability. We also compare the convergence speed between AL-parallel-mixed with AL-poll-ReID. It is measured by the ratio of successfully tracked targets. The comparison between polling variant of AL and our parallelized version is shown in Fig.3. Trained via parallel variant of AL, our tracker could converge much faster than the polling version: in the 13-th of iteration, we have successfully tracked 100% of the training targets, whereas the polling one needs 39 iterations.

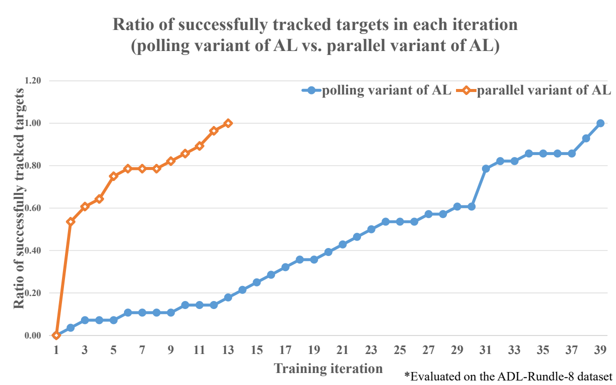

Fig. 3. Ratio of objects successfully tracked in each iteration on the ADL-Rundle-8 dataset.

4.3. Evaluation results on test set

We report the performance of our mixed style tracker on the MOT Challenge 2015 test set in Table 2. The training sequences are selected according to their names for each test sequence respectively. We compare our approach with both offline methods[1, 2, 3] and the online counterparts[15, 4]. From Table 2, we could see that our framework improves significantly over other methods on all metrics except MOTP. The tracking precision (MOTA) outperforms the second best with 2.3%. The IDS number is merely 580, which is also significant lower than the others. We also obtains the most partially tracked trajectories (16.00%) and the least lost trajectories (34.40%) on the test set. On the other hand, our tracking precision (MOTP) representing the object detection performance is a little bit lower than those of others (0.2% lower than the best). This implies a few of the objects are misclassified to the *inactive* state.

Table 2. Evaluation results on MOT Challenge 2015 test set.

Method	MOTA	MOTP	MT(%)	PT(%)	ML(%)	IDS
LP2D[1]	19.80	71.20	6.70%	52.10%	41.20%	1649
MotiCon[20]	23.10	70.90	10.40%	48.30%	41.30%	1018
LINF1[2]	24.50	71.30	5.50%	29.90%	64.60%	744
LP_SSVM[3]	25.20	**71.70**	5.80%	41.20%	53.00%	646
SCEA[4]	29.10	71.10	8.90%	43.80%	47.30%	604
OnlineMDP[15]	30.30	71.50	13.00%	48.60%	38.40%	690
AL-parallel-mixed	**32.60**	71.30	**16.00%**	**49.60%**	**34.40%**	**580**

5. CONCLUSION

In this paper, we propose a mixed style tracker that incorporates "future" observations in a mixed style tracker to process offline data. It not only retains the efficiency of online tracker but also achieves robustness of offline tracker by utilizing more observations. We model multiple object tracking by viewing each object as an agent following the Markov Decision Process. In order to effectively utilize offline data, backward tracklets are generated by rewinding the video sequence. A parallelized apprenticeship learning algorithm is proposed to efficiently learn both reward function and policy for state transition. Furthermore, the person ReID features extracted from deep neural networks are used for appearance measurement so that the robustness to occlusion and pose change is improved. We evaluate our tracker on a challenging MOT benchmark and experiment results exhibit that our framework outperforms the state-of-the-art methods significantly.

6. ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (No.61571071), the Natural Science Foundation of Chongqing Science and Technology Commission (No. cstc2014jcyjA40048).

7. REFERENCES

[1] Roberto Henschel, Laura Leal-Taixé, and Bodo Rosenhahn, "Efficient multiple people tracking using minimum cost arborescences," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 265–276.

[2] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking," in *European Conference on Computer Vision, 2016*. Springer, 2016, pp. 774–790.

[3] Shaofei Wang and Charless C Fowlkes, "Learning optimal parameters for multi-target tracking with contextual interactions," *International Journal of Computer Vision*, pp. 1–18, 2016.

[4] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon, "Online multi-object tracking via structural constraint event aggregation," in *Computer Vision and Pattern Recognition, 2016*, 2016, pp. 1392–1400.

[5] Wenhan Luo, Junliang Xing, Xiaoqin Zhang, Xiaowei Zhao, and Tae-Kyun Kim, "Multiple object tracking: A literature review," *arXiv preprint arXiv:1409.7618*, 2014.

[6] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European Conference on Computer Vision, 2004*. Springer, 2004, pp. 28–39.

[7] Louis Kratz and Ko Nishino, "Tracking with local spatio-temporal motion patterns in extremely crowded scenes," in *Computer Vision and Pattern Recognition, 2010*. IEEE, 2010, pp. 693–700.

[8] Zhen Qin and Christian R Shelton, "Improving multi-target tracking via social grouping," in *Computer Vision and Pattern Recognition, 2012*. IEEE, 2012, pp. 1972–1978.

[9] Suna Kim, Suha Kwak, Jan Feyereisl, and Bohyung Han, "Online multi-target tracking by large margin structured learning," in *Asian Conference on Computer Vision, 2012*. Springer, 2012, pp. 98–111.

[10] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.

[11] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Computer Vision and Pattern Recognition, 2011*. IEEE, 2011, pp. 1201–1208.

[12] Hao Jiang, Sidney Fels, and James J Little, "A linear programming approach for multiple object tracking," in *Computer Vision and Pattern Recognition, 2007*. IEEE, 2007, pp. 1–8.

[13] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *European Conference on Computer Vision, 2012*, pp. 343–356. Springer, 2012.

[14] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah, "Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *Computer Vision and Pattern Recognition, 2015*. IEEE, 2015, pp. 4091–4099.

[15] Yu Xiang, Alexandre Alahi, and Silvio Savarese, "Learning to track: Online multi-object tracking by decision making," in *International Conference on Computer Vision, 2015*. IEEE, 2015, pp. 4705–4713.

[16] Pieter Abbeel and Andrew Y Ng, "Apprenticeship learning via inverse reinforcement learning," in *International Conference on Machine learning, 2004*. ACM, 2004, p. 1.

[17] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Computer Vision and Pattern Recognition, 2016*, 2016, pp. 1335–1344.

[18] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang, "Learning deep feature representations with domain guided dropout for person re-identification," *arXiv preprint arXiv:1604.07528*, 2016.

[19] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[20] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese, "Learning an image-based motion context for multiple people tracking," in *Computer Vision and Pattern Recognition, 2014*. IEEE, pp. 3542–3549.