ADAPTIVE MATCHING STRATEGY FOR MULTI-TARGET MULTI-CAMERA TRACKING

Chong Liu[†‡] *Yuqi Zhang*[⋆] *Weihua Chen*[⋆] *Fan Wang*[⋆] *Hao Li*[⋆] *Yi-Dong Shen*[†*]

[†] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[‡] University of Chinese Academy of Sciences, Beijing China
[⋆] Machine Intelligence Technology Lab, Alibaba Group

ABSTRACT

Multi-Target Multi-Camera Tracking has a wide range of applications and is the basis for many high-level inference and prediction tasks. How to make the system perform efficiently on a large number of cameras is a crucial research issue. Previous works have proposed many matching strategies to reduce the matching range and improve the matching accuracy. However, these works require human participation when formulating matching strategies, which becomes infeasible as the scale of the camera system increases. To tackle this problem, we propose an adaptive matching strategy to replace manual rules when guiding the matching between cameras. Specifically, we use the Markov decision process to model the tracklets matching problem between cameras. Reinforcement learning and imitation learning are combined to predict a set of cameras where the tracking target might be located. The predicted candidate camera set can be used for inter-camera matching and association between tracklets. Moreover, our method can be trained with or without ground truth inter-camera trajectories, making it more practical in real scenarios. We evaluate our method on the city-scale tracking dataset Cityflow, and the proposed method is sufficient to replace manual rules, and finally improve the performance of the overall MTMCT system.

***Index Terms*—** MTMCT, Reinforcement learning, Imitation learning, Matching strategy

1. INTRODUCTION

Multi-Target Multi-Camera Tracking (MTMCT) aims to identify and track all target objects under multiple cameras. The resulting multi-camera target tracklets can be used for many advanced inference and prediction tasks (*e.g.*crowd behavior analysis, anomaly trajectories discovery). Generally, the MTMCT system includes two subtasks: 1) Single Camera Tracking (SCT): Detecting and tracking the target in each camera; 2) Inter-Camera Tracking (ICT): The association of target tracklets between different cameras. SCT as a standalone task, has already seen lots of related works [1, 2, 3]; therefore, the research focus of MTMCT has been on ICT, which has two aspects: 1) Re-identification (ReID) features and metric learning 2) tracklets association.

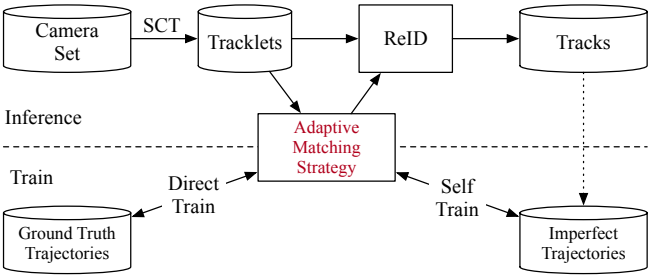

Fig. 1. **Pipeline of Multi-Target Multi-Camera Tracking.** AMS method uses tracklets to guide the association between cameras to achieve the purpose of accelerating and improving performance. AMS can use ground truth trajectories to direct train, or use imperfect trajectories to self train.

These two aspects supplement each other. ReID features serve as a crucial criterion in tracklets association, and the ReID features extracted from deep learning architecture have achieved significant improvements [4, 5, 6, 7, 8, 9]. The purpose of tracklets association is mainly to reduce the matching scope through statistics or rules to reduce false matches and increase accuracy and speed. Many works have studied the transfer of tracking targets between cameras: In DukeMTMC [10], the training set and test set are captured with the same camera topology, [11, 12] obtain the spatio-temporal topology between the cameras through statistical methods. For Cityflow [13] which has different scenarios in the training set and test set, [14, 15, 16] artificially formulate matching strategies by manually observing the camera connections in the test set.

From the previous works, we can see that most existing methods for tracklets association involve a large portion of manual design. For a camera system with a different topology, matching strategies need to be carefully designed by humans repetitively. This will become infeasible once the number of cameras is increased to hundreds or thousands in a city-

The work was done when Chong Liu was intern at Alibaba Group. This work was supported by Alibaba Group through Alibaba Research Intern Program and China National 973 program 2014CB340301.

* Corresponding author.

scale scenario. Therefore, how to automatically formulate robust matching strategies in different camera system scenarios is a very crucial research issue which also has great practical value.

To tackle the above problem, we propose an adaptive matching strategy to replace manual rules to guide the tracking and matching between cameras. We model the matching process in the form of reinforcement learning, where an agent is trained to predict the set of cameras where the tracking target will possibly appear in the next frame, by observing the current state of the target. We propose an adaptive matching strategy (AMS) model that combines Reinforcement Learning (RL) [17] and Imitation Learning (IL) [18]. The IL part learns from a teacher to imitate its behavior, however it will fail to produce reasonable behavior when the ground truth data is missing. In a complex multi-camera system, it is unlikely to have trajectories going through all possible routes. Therefore it is necessary to bring in RL for its great exploratory ability. By combining the strategic learner and non-strategic learner, AMS achieves both exploratory ability and stability. In addition, ground truth inter-camera trajectories are expensive to be annotated, and our model can be self-trained with imperfect trajectories at a certain accuracy level and still achieve reasonably well performance, showing its potential in the large-scale real world applications.

2. METHOD

2.1. Overview

The basic framework of MTMCT is shown in Figure 1. Our proposed AMS is mainly used to provide inter-camera matching strategies for ReID. We abstract the camera matching strategy as follows: each tracklet finds out the possible camera set of tracklets to limit the matching range of ICT. This has two advantages: **1)** reduce the range of matching, improve time efficiency, and provide the possibility for intelligent tracklets association between cameras in a large-scale surveillance environment. **2)** exclude some tracklets that are mismatched because of their similar appearance to improve matching accuracy.

The purpose of AMS is to predict for each tracklet the possible camera set of the tracklets with the same ID. ReID will match the tracklets according to the proposal of AMS. We try to make predictions under a novel reinforcement learning framework. The model framework is shown in Figure 2.

2.2. Problem Settings

Predicting the set of cameras in each frame of tracklets can be regarded as a sequence decision task. We follows the Markov Decision Process, which consists of states $s \in S$, actions $a \in A$, state transition functions $f(s, a)$, and rewards r.

State In t-th frame, the state $s_t = (c_t, g_t, b_t)$, where c_t is the one-hot vector of the camera where the current tracklets are located, g_t is the camera set with the same ID in the next frame, $b_t = [x_t, y_t, h_t, w_t]$ is the bounding box of the current tracklet at t-th frame, (x_t, y_t) are the center coordinates of the bounding box, h_t and w_t represent the height and width of the bounding box.

Action The predicted action $a_t = \{\alpha_1, \ldots, \alpha_i, \ldots, \alpha_N\} \in \{0, 1\}^N$ in the t-th frame is a N-dimensional vector, where N is the number of cameras in the camera system. For a_t, if the tracking target is in the i-th camera, then $\alpha_i = 1$; if the tracking target is not in the i-th camera, then $\alpha_i = 0$.

State Transition Function According to the state s_t and action a_t of frame t, the state of the next frame s_{t+1} can be determined by the state transition function: $s_{t+1} = f(s_t, a_t)$, if the target is found in the camera set of a_t, the state of the next frame of the target is returned; if the target is not found in the camera set of a_t, the current state is continued.

Reward The reward function $r_t = R(g_t, a_T)$ describes the relationship between the action a_t and the actual result. The closer to the actual result, the higher the reward we give:

$$R(g_t, a_t) = \begin{cases} 2 & \text{if } \sum_i^N g_t^i \dot{a}_t^i = N \\ -1 & \text{if } \sum_i^N g_t^i \dot{a}_t^i = 0 \\ \frac{\sum_i^N g_t^i \dot{a}_t^i}{N} & \text{else} \end{cases}, \quad (1)$$

where, g_t is the ground truth camera set in the next frame.

2.3. Adaptive Matching Strategy

Imitation Learning (IL) We follow the IL method of Behavioral Cloning [18], where the agent learns to imitate the teacher's behavior. The teacher demonstrates the teacher action g_t at each time step t, which represents the camera set of the target in the next frame. The agent learns from this supervised information by minimizing the Multi Label Soft Margin Loss [19] of the teacher's action g_t. The IL losses are as follows:

$$\begin{aligned} \mathcal{L}_{MLSM}(x, y) = &- \frac{1}{N} \sum_i^N (y_i \times \ln \frac{1}{1 + e^{-x_i}} \\ &+ (1 - y_i) \times \ln \frac{e^{-x_i}}{1 + e^{-x_i}}), \end{aligned} \quad (2)$$

$$\mathcal{L}^{IL} = \sum_t \mathcal{L}_t^{IL} = \sum_t \mathcal{L}_{MLSM}(p_t, g_t). \quad (3)$$

Different from Behavioral Cloning, we use Multi Label Soft Margin Loss for multi-classification, because the target may exist in multiple cameras in the next frame due to the overlapping field-of-view between cameras.

Reinforcement Learning (RL) In RL, we apply the strategy-based reinforcement learning method Advantage Actor-Critic

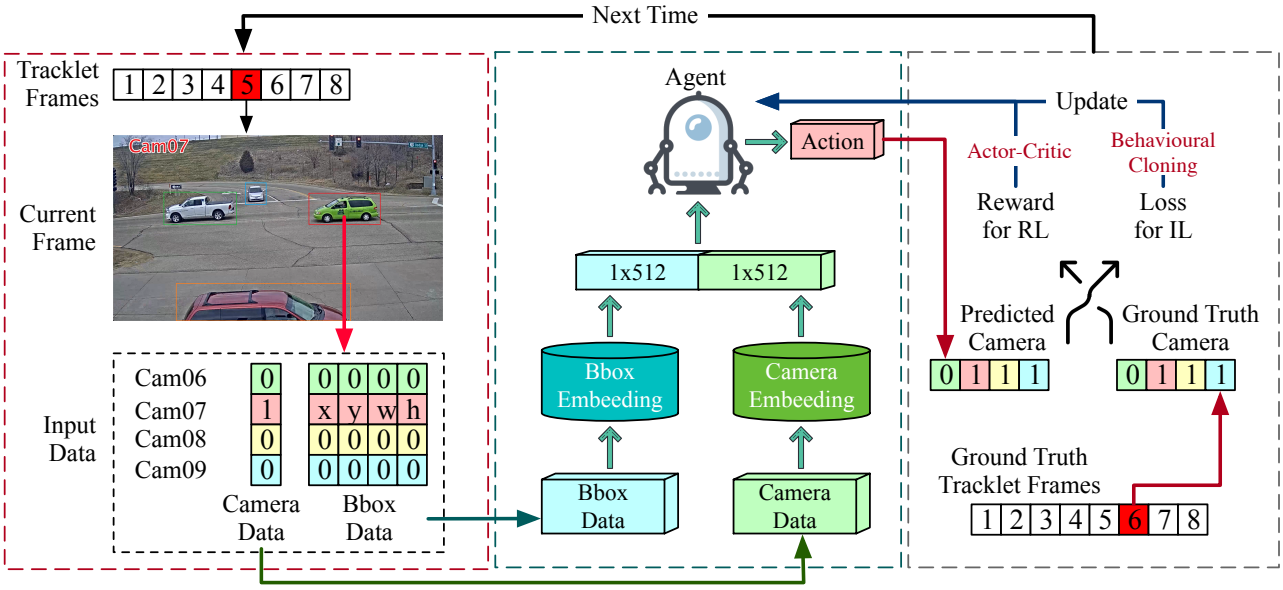

Fig. 2. **Overview of the architecture.** The agent takes the observation information of the current frame as input, and then outputs action to predict the set of tracking target cameras in the next frame. For each tracklet, specific forms of camera data and Bbox data are generated from the visual observation of the current frame. The agent predicts the next frame of action based on the encoded data and compares it with the ground truth, and updates the agent parameters in the form of RL and IL.

(A2C) [17], the agent takes action a_t from the distribution p_t through Bernoulli sampling, and learn from the reward $r_t = r(s_{t+1}, a_t)$. To this end, we use MLP to define the Critic network for calculating the state value: $v_t = \text{Critic}(h_t)$, where h_t is the input feature of the state s_t obtained by the LSTM encoder, and finally the loss function of RL is:

$$L^{RL} = \sum_t ((r_t + v_{t+1} - v_t)^2 - \ln(p_t(a_t)) \times (r_t + v_{t+1})), \tag{4}$$

where, $p_t(a_t)$ represents the probability of taking action a_t under distribution p_t. $\mathcal{L}^{RL}$ ensures that the behavior is chosen with the greatest reward as much as possible.

Mixed IL+RL Although IL can learn the corresponding behavior from ground truth, it cannot guarantee that the selected action can meet all requirements of the tracklets. Because the agent using IL is biased towards the behavior of the teacher, for the state where ground truth data is missing, the agent of IL cannot produce reasonable behavior. RL has a strong capability to explore unknown states, but the behavior given by the agent may be unstable. Therefore we propose to mix IL and RL to take advantage of both non-strategic learners and strategic learners. IL and RL agents share weights, which are updated based on the losses (Figure 2). The mixed loss is the weighted sum of $\mathcal{L}^{IL}$ and $\mathcal{L}^{RL}$: $\mathcal{L}^{AMS} = \mathcal{L}^{RL} + \lambda_{IL}\mathcal{L}^{IL}$. Within this formula, IL regulates the behavior of RL based on the supervision information, and RL increases IL's ability to explore unknown states.

2.4. Pipeline

Training As shown in Figure 1, our method has two methods in the training phase: **1) Direct training:** For a camera system with ground truth inter-camera trajectories, our method can directly use the supervised information for training; **2) Self training:** for a camera system without any supervision about the inter-camera trajectories, we can use the imperfect tracklets generated by the existing MTMCT system as training data for self-training. Compared with direct training, self-training has more practical value. For city-scale systems with thousands of cameras in the real world, it is impossible for us to establish complete supervision information for all cameras. Subsequent experiments proved that our method can also achieve superior performance in the settings of self-training without annotated ground truth.

Inference For the reasoning process as shown in Figure 1, we need to first generate tracklets within each camera using the existing SCT method [1, 2]. Then these intra-camera tracklets need to be serialized into the same format as those in the training process. We use AMS to generate a set of cameras at each frame for each tracklet, called frame-camera set (t, ζ_t). Then according to the time-camera set of each tracklet T_{ID}, the candidate set is filtered out according to the following rule: if two tracklets may exist in the same camera at the same frame, they may belong to the same tracking target. When matching between cameras in the MTMCT system, each tracklets T_{ID} only needs to be matched with its candidate tracklets C_{ID} based on their appearance similarity, which can largely reduce the number of matches and eliminate false matches.

Method	IDF1	IDP	IDR	MN(1e4)
LAMM [6]	63.00	60.70	66.00	-
Locality Awar [20]	65.19	63.00	67.70	-
S-T Cue[15]	66.53	-	-	-
S-T Cons[21]	68.65	-	-	-
Camera Link[14]	70.59	69.12	72.11	-
Baseline*[15]	62.76	60.29	65.44	1.7
S-T Cue*[15]	66.52	69.31	63.95	0.8
AMS-Pre	**68.10**	**66.54**	**69.73**	**0.2**
AMS-Self	**67.04**	**64.29**	**70.03**	**0.3**

Table 1. **MTMCT results on CityFlow dataset.** Both AMS-Pre and AMS-Self perform better than the compared manual method (S-T Cue). MN represents the matching number of inter-camera tracklets association. * reproduced results in my environment.

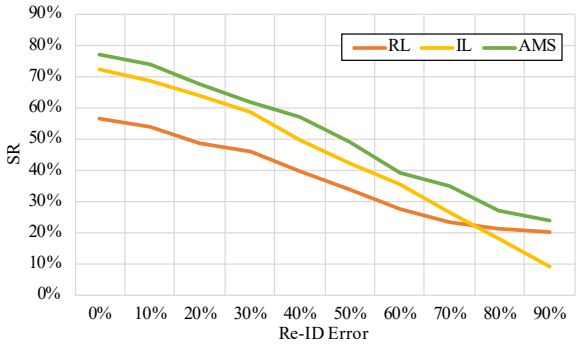

Fig. 3. **Action success rate under different ReID errors.** With the increase of ReID error, the performance of all modules will decrease, and AMS still has better performance.

3. EXPERIMENT

3.1. Dataset and Evaluation Setting

Dataset We use the CityFlow [13] dataset for evaluation. CityFlow is the largest and most representative MTMCT data set captured in real scenarios in a city. It contains 3.25 hours of traffic video from 40 cameras at 10 intersections in a medium-sized city, spanning approximately 2.5 kilometers. In addition, CityFlow covers a range of different road traffic types, including intersections, road extensions and highways.

Evaluation Metrics For MTMCT, we use IDF1, IDP and IDR as evaluation indicators. IDF1 [10] calculates the ratio of the number of correctly identified detections to the ground truth and the average number of calculated detections. Meanwhile, in order to measure the accuracy of AMS's prediction about the camera set where the target is located, we define Success Rate (SR): $SR = \frac{\sum_{T \in \mathcal{T}} TF_T}{\sum_{T \in \mathcal{T}} FN_T}$, where, $\mathcal{T}$ is the set of all tracklets, TF_T is the number of correctly predicted Truth Frames in tracklet T (only if all cameras in the current frame have been successfully predicted), FN_T It is the number of all frames of tracklet T.

Implementation Details For AMS, we use the same set of training parameters for direct training using ground truth trajectories and self training using imperfect trajectories. The training runs for 40,000 epochs, which takes about two hours under a Tesla P100. Specifically, we use RMSProp Optimizer [22] to optimize the network, the learning rate is 0.0001, and the hyperparameters of $\mathcal{L}^{AMS}$ are $\lambda_{IL} = 0.2$. In order to evaluate AMS compared to manual rules, we use the same settings as S-T Cue [15] for a fair comparison. Specifically, we use the same method to obtain intra-camera tracklets, the same ReID features for connecting two candidate tracklets, *et al.*. The only difference between the two MTMCT systems being compared is the matching strategy.

3.2. Evaluation and Ablation Study

Evaluation In the CityFlow dataset, manually designed rules have been proven to be effective [20, 15, 21, 14]. However, these methods use different SCT methods and ReID features, and directly comparing the MTMCT performance with all these factors entangled together is not fair. Therefore, by following the same experiment setups as S-T Cue and performing the comparison with the same baseline, we focus on the effect introduced by only switching between different matching rules. The results are shown in Table 1. Both AMS trained with ground truth trajectories (AMS-Pre) and AMS trained with imperfect trajectories (AMS-self) have better matching performance than S-T Cue. The AMS method also greatly reduces the number of matching when connecting tracklets across cameras.

Ablation Study Fig. 3 shows the success rate for IL, RL and AMS under different ReID error rates. As the error rate increases, the success rate for all methods decreases but AMS keeps outperforms the other two. When the error rate is too high, the performance of RL declines slows, because RL is more exploratory and is not limited by the low quality training data to some extent.

4. CONCLUSION

In this paper, we propose an adaptive matching strategy (AMS) to replace manual rules to guide the tracklet matching between cameras. Specifically, we use the Markov decision process to model the tracklets matching problem between cameras. We combine reinforcement learning and imitation learning to predict a set of cameras where the tracking target might be located. AMS builds a matching strategy to guide the tracklets association by predicting the transfer of the target between the cameras. Experimental results show that the proposed method can effectively reduce the number of tracklets matching between cameras, leading to a better and more robust strategy compared with manual matching.

5. REFERENCES

[1] Zhang Yifu, Wang Chunyu, Wang Xinggang, Zeng Wenjun, and Liu Wenyu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *IJCV*, pp. 3069–3087, 2021.

[2] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," in *Proc. ACM MM*, 2019, pp. 482–490.

[3] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang, "Towards real-time multi-object tracking," in *Proc. ECCV*, 2020, pp. 107–122.

[4] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang, "Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features," in *Proc. CVPR*, 2018, pp. 108–115.

[5] Ergys Ristani and Carlo Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proc. CVPR*, 2018, pp. 6036–6046.

[6] Yunzhong Hou, Liang Zheng, Zhongdao Wang, and Shengjin Wang, "Locality aware appearance metric for multi-target multi-camera tracking," *arXiv preprint arXiv:1911.12037*, 2019.

[7] Chong Liu, Xiaojun Chang, and Yi-Dong Shen, "Unity style transfer for person re-identification," in *Proc. CVPR*, 2020, pp. 6887–6896.

[8] Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin, "Cdtrans: Cross-domain transformer for unsupervised domain adaptation," *arXiv preprint arXiv:2109.06165*, 2021.

[9] Kai Chen, Weihua Chen, Tao He, Rong Du, Fan Wang, Xiuyu Sun, Yuchen Guo, and Guiguang Ding, "Tagperson: A target-aware generation pipeline for person re-identification," *arXiv preprint arXiv:2112.14239*, 2021.

[10] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. ECCV*, 2016, pp. 17–35.

[11] Na Jiang, SiChen Bai, Yue Xu, Chang Xing, Zhong Zhou, and Wei Wu, "Online inter-camera trajectory association exploiting person re-identification and camera topology," in *Proc. ACM MM*, 2018, pp. 1457–1465.

[12] Olly Styles, Tanaya Guha, and Victor Sanchez, "Multi-camera trajectory forecasting with trajectory tensors," *IEEE TPAMI*, 2021.

[13] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *Proc. CVPR*, 2019, pp. 8797–8806.

[14] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang, "Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models.," in *Proc. CVPR Workshops*, 2019, pp. 416–424.

[15] Zhiqun He, Yu Lei, Shuai Bai, and Wei Wu, "Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue.," in *Proc. CVPR Workshops*, 2019, pp. 203–212.

[16] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen, "City-scale multi-camera vehicle tracking guided by crossroad zones," in *Proc. CVPR Workshops*, 2021.

[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. ICML*, 2016, pp. 1928–1937.

[18] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., "End to end learning for self-driving cars," *arXiv e-prints*, pp. arXiv–1604, 2016.

[19] Andre Martins and Ramon Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. ICML*, 2016, pp. 1614–1623.

[20] Yunzhong Hou, Heming Du, and Liang Zheng, "A locality aware city-scale multi-camera vehicle tracking system," in *Proc. CVPR Workshops*, 2019, pp. 167–174.

[21] Peilun Li, Guozhen Li, Zhangxi Yan, Youzeng Li, Meiqi Lu, Pengfei Xu, Yang Gu, Bing Bai, Yifei Zhang, and DiDi Chuxing, "Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking.," in *Proc. CVPR Workshops*, 2019, pp. 222–230.

[22] T Tieleman and G Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *Technical Report.*, 2017.