Machine Intelligence Research
www.mi-research.net

20(2), April 2023, 194-206
DOI: 10.1007/s11633-022-1380-x

A Study of Using Synthetic Data for Effective Association Knowledge Learning

Yuchi Liu[1] Zhongdao Wang[2] Xiangxin Zhou[2] Liang Zheng[1]

[1] College of Engineering & Computer Science, Australian National University, Canberra 2601, Australia

[2] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Abstract: Association, aiming to link bounding boxes of the same identity in a video sequence, is a central component in multi-object tracking (MOT). To train association modules, e.g., parametric networks, real video data are usually used. However, annotating person tracks in consecutive video frames is expensive, and such real data, due to its inflexibility, offer us limited opportunities to evaluate the system performance w.r.t. changing tracking scenarios. In this paper, we study whether 3D synthetic data can replace real-world videos for association training. Specifically, we introduce a large-scale synthetic data engine named MOTX, where the motion characteristics of cameras and objects are manually configured to be similar to those of real-world datasets. We show that, compared with real data, association knowledge obtained from synthetic data can achieve very similar performance on real-world test sets without domain adaption techniques. Our intriguing observation is credited to two factors. First and foremost, 3D engines can well simulate motion factors such as camera movement, camera view, and object movement so that the simulated videos can provide association modules with effective motion features. Second, the experimental results show that the appearance domain gap hardly harms the learning of association knowledge. In addition, the strong customization ability of MOTX allows us to quantitatively assess the impact of motion factors on MOT, which brings new insights to the community.

Keywords: Multi-object tracking (MOT), data association, synthetic data, motion simulation, association knowledge learning.

Citation: Y. Liu, Z.Wang, X. Zhou, L. Zheng. A study of using synthetic data for effective association knowledge learning. *Machine Intelligence Research*, vol.20, no.2, pp.194–206, 2023. http://doi.org/10.1007/s11633-022-1380-x

1 Introduction

Multi-object tracking (MOT) is a compound system composed of several functional components, e.g., detection, visual representations, and association. Association is at the final stage of the MOT pipeline and is usually viewed as the core problem, aiming to connect bounding boxes with existing tracklets[1, 2]. The association module makes inferences according to appearance features (e.g., re-identification features), motion features (e.g., location and size of bounding boxes), or both of them.

In the community, what many solutions to the association have in common is that they are trained with real-world video data[3, 4]. However, there are several potential problems with this practice. First, annotating trajectories in video frames requires expensive labor costs. This potentially limits the scale of MOT training data. Second, privacy and ethics issues constrain the usage of real-world data in human-centered tasks, e.g., multiple pedestrian tracking.

Research Article
Special Issue on Large-scale Pre-training: Data, Models, and Fine-tuning
Manuscript received July 19, 2022; accepted October 13, 2022; published online March 8, 2023
Recommended by Associate Editor Jun Zhao

In this paper, we investigate how to use synthetic data in MOT, so as to avoid the concerns listed above. We build a 3D simulation engine, MOTX, for generating videos with multiple targets, rich annotations, and controllable visual factors. Such data offer an inexpensive way to acquire large-scale data with accurate labels. With MOTX, we aim to answer two interesting questions.

First, does the association knowledge learned from synthetic data work in real-world videos? A common weakness of synthetic data consists of its distribution difference with real-world data, especially regarding the image-style. In "Appearance-centered" tasks (e.g., re-identification and segmentation), to avoid failure in real-world test environments, models trained on synthetic data require additional training techniques, such as fine-tuning or domain adaptation on the real data[5–8]. However, association learning is different from appearance learning regarding data requirements. According to existing works[1, 2, 9], motion cues play an essential role in the association. While appearance realistic images are hard to simulate by the engine, it may be less difficult for motion cues, such as occlusion. Some sample results of appearance simulation and association scenario simulation are shown in Fig. 1.

Second, how do motion factors affect association

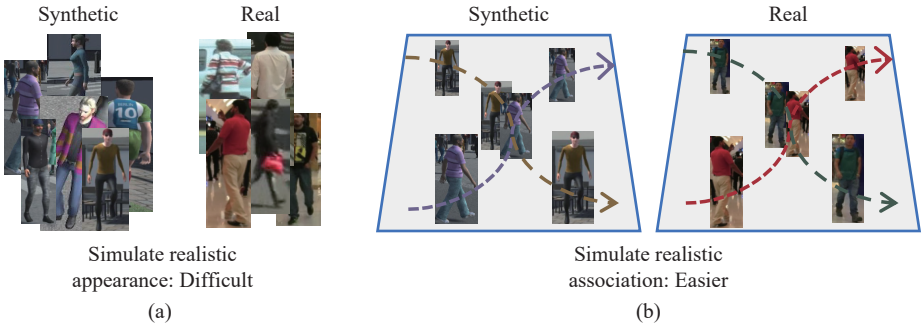


Fig. 1 Simulated appearance VS. simulated association scenarios. (a) Simulated appearance usually has an image-style discrepancy with the real-world appearance. For many appearance-centered tasks such as re-identification, such appearance domain gap compromises models that are trained on synthetic data and tested on real data. (b) In comparison, we show that synthetic data are as effective as real data in training association models. It suggests that association scenarios (e.g., trajectories and occlusions) have a small domain gap between the synthetic and the real.

knowledge learning? Existing datasets are mostly from the real world, such as MOT15. While these data benefit model training, that they are fixed offers us limited opportunities to understand how the system reacts to changing visual factors. For example, how does pedestrian density in the training set affect model accuracy? Can a model trained with static cameras be well deployed under moving-camera systems? In this paper, taking advantage of the strong customization ability of MOTX, we will make some initial investigations into these interesting directions.

In response, this paper makes a two-fold contribution. First and most importantly, we show that on several state-of-the-art association networks, association knowledge learned from synthetic data can be well adapted to real-world scenarios without a performance drop. Specifically, we synthesize datasets using MOTX by manually setting key parameters (e.g., camera view) to be close to real-world training sets.[1] Then, when the recent association networks are trained on such synthetic videos, they achieve similar or sometimes even better tracking accuracy compared with real data training. Our ablation studies on appearance and motion features suggest: 1) The appearance-discrepancy between synthetic data and real-world data can hardly harm the association knowledge learning. 2) 3D engines can well simulate motion cues in association scenarios. The above findings can be the reason for the competitiveness of synthetic data and imply that MOT benefits more from using synthetic data than "Appearance-centered" tasks. To our knowledge, this is a very early study of pondering the role of synthetic data in MOT.

Second, we perform empirical studies on how object-related and camera-related factors affect the learning of association knowledge. Specifically, we investigate two groups of factors: 1) pedestrian-related factors, such as density and moving speed; and 2) camera-related factors, including the camera view and camera moving state. In detail, with the proposed MOTX engine[2], motion factors are abstracted with system parameters, so we can readily simulate different scenarios by simply changing these parameters, e.g., setting the object velocity to 1 m/s. Our results shed light on the relationship between factors in training and testing data and MOT system performance.

2 Related work

Association methods in MOT. There are mainly two types of association: human-designed policies and parametric association modules. The former is usually seen in MOT works focusing on improving detection and appearance embedding[9–15]. They compute similarities between bounding boxes and objects according to predefined metrics. The most commonly used metrics are the intersection over union (IoU) score and the cosine similarity score between deep Re-ID features. Then, a bipartite matching algorithm (e.g., the Hungarian algorithm[16]) associates the bounding boxes with objects. The Kalman Filter[17] can also predict motion and smooth trajectories.

The latter uses neural networks to formulate the association stage. For example, DeepMOT[2] proposes a long short-term memory (LSTM) method to approximate the Hungarian matching algorithm[16]. MPNTracker[1] formulates sequences as graphs and designs a differentiable message passing network to predict the score for each box link between frames. Li et al.[9] and Papakis et al.[18] use a graph neural network to model appearance and motion (geometric) features and produce similarities between tracklets and detections. These parametric association modules are trained based on appearance and motion features. In this paper, we observe that parametric association modules trained with synthetic videos can be successfully deployed in real-world test sets without domain adaptation.

Learning from synthetic data for real-world applications. Synthetic datasets have been used in image

[1] Our manual parameter tuning is very efficient: A rough estimation of the motion parameters will be enough.

[2] This engine is publicly released at https://github.com/liuyvchi/MOTX.

classification[19, 20], object detection[21–23], multi-object tracking[21, 22, 24], semantic and instance segmentation[21, 22, 25], pose estimation[24, 26] and navigation[27]. Commonly used simulation platforms include Unity and Unreal. In this area, domain adaptation is mostly used. For example, Bąk et al.[5] use the cycle generative adversarial network to convert synthetic images into the real-world style. In comparison, there are much fewer works that do not need domain adaptation to get good performance in this area. Unlike the common practices using synthetic data to learn appearance features, this paper investigates the possibility of using synthetic data to focus on association module training in MOT.

Domain gap beyond appearance. While the domain gap caused by the image appearance is the most studied, there are some works studying other factors that lead to distribution differences between domains. Recently, Meta-sim[28] optimizes the probability grammar for scene content generation. Yao et al.[8] study the content-level domain gap in the vehicle re-identification task and show the feasibility of reducing the gap by editing synthetic data. This paper will identify and discuss factors beyond appearance (i.e., motion factors) that influence association learning in MOT.

3 MOTX engine

MOTX is a 3D rendering engine that receives a set of controllable factors related to objects, cameras, and others as inputs, and it outputs a 2D video together with ground truth annotations (Fig. 2). We build MOTX based on the Unity[29] game engine. Section 3.1 introduces controllable factors. Section 3.2 describes annotation acquisition.

3.1 Controllable factors

Object-related factors. Currently, MOTX focuses on tracking pedestrians. We collect 1 200 pedestrian 3D models with distinct appearances from the PersonX engine[30]. Controllable factors include pedestrian density, speed, and action. Density refers to the number of pedestrians inside the viewing frustum. Each pedestrian takes action $\{\text{walk}, \text{run}\}$ with a random speed drawn from a given speed distribution. The walking routes are randomly generated.

Camera-related factors. The viewing pose, spatial location, running path, and speed of the camera can be flexibly adjusted. In this paper, we mainly evaluate two commonly encountered camera views, the surveillance view (static camera, overlooking view) and the vehicle-mounted view (moving camera, near-horizontal view).

Others factors. MOTX supports changing other visual factors that can influence the final rendering, including scenes, resolution, and lighting (light direction, light intensity, light color, etc.). If not specified, all videos are recorded at the resolution of 1 024×768.

3.2 Annotation acquisition

Bounding box annotation. We transform the 3D locations of person models in the scene into 2D locations in the camera view. By calculating the top, bottom, left, and right vertices of people, we can obtain accurate bounding boxes for the holistic body. For occluded or partially visible persons, the engine can tell the occlusion relations, and we accordingly annotate the bounding boxes of visible parts as well.

Identity annotation. Identity labels are directly given by the engine. This avoids the re-labeling problem when a person leaves and re-enter the field of view, which is a common annotation mistake in real datasets.

In practice, we build assets in the Unity engine and provide user interfaces to configure the predefined status of controllable factors and trajectories for persons and cameras in a 3D scene. The Unity engine will render the video according to our configurations automatically. An example of rendering videos in MOTX is shown in Fig. 3.

4 Association knowledge

A multiple-object tracker is usually composed of a detector, an appearance model, and an association model. In this work, we argue that it is possible to learn the associ-

Fig. 2 MOTX is inexpensive and accurate in generating videos and their labels for association training. Controllable factors include (a) camera view, (b) camera moving state, (c) pedestrian density, and (d) pedestrian speed.

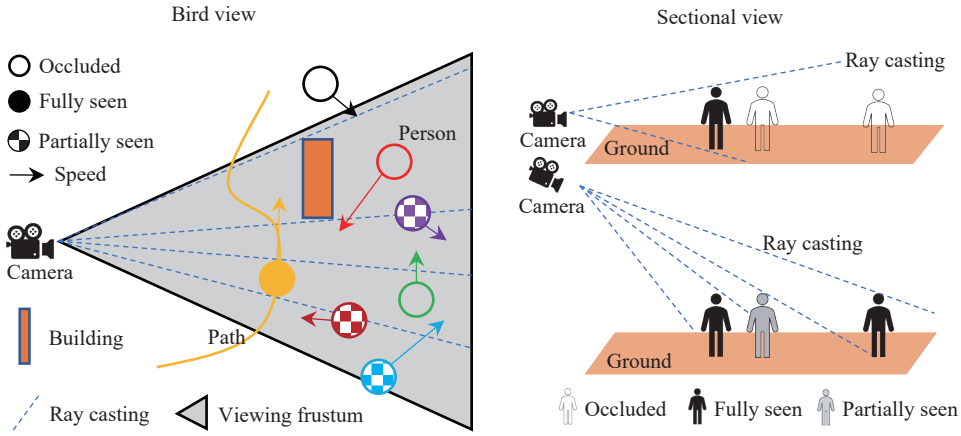


Fig. 3 Example of rendering in MOTX. Left: Bird view shows how we control the path and speed of persons and cameras. Right: Sectional view, where the change of camera view causes some persons to be invisible due to occlusions. Ray casting is used to check visibility: occluded persons do not have a bounding box annotation.

ation model with synthetic videos generated by the MOTX engine, while the learned association knowledge is applicable to real-world data without domain adaptation. Preliminary, we give a definition of association knowledge and briefly review how existing methods learn it.

4.1 Definition of association knowledge

Given a set of detected bounding boxes $\mathcal{D}_t$ and tracked objects $\mathcal{O}_t$ at frame t, the assignment between the i-th bounding box d_i in $\mathcal{D}_t$ and the j-th object o_j in $\mathcal{O}_t$ is noted as a_{ij}, where $a_{ij} \in \{0, 1\}$. $a_{ij} = 1$ denotes that d_i is associated with o_j. Otherwise, d_i belongs to other tracklets. The association module in an MOT system usually aims to optimize the assignment matrix $\mathcal{A}$ at frame t:

$$\mathcal{A}_t^* = \arg\max_{\mathcal{A}_t} \sum_{i=1}^{|\mathcal{D}_t|} \sum_{j=1}^{|\mathcal{O}_t|} a_{ij} s_{ij}$$
$$\text{s.t. } \mathcal{A}_t \in \{0,1\}^{|\mathcal{D}_t| \times |\mathcal{O}_t|}; \sum_{i=1}^{|\mathcal{D}_t|} a_{ij} \leq 1 \; ; \sum_{j=1}^{|\mathcal{O}_t|} a_{ij} \leq 1 \quad (1)$$

where a_{ij} is the entry of $\mathcal{A}_t$ and s_{ij} is the association score between d_i and o_j. If $\sum_{i=1}^{|\mathcal{D}_t|} a_{i,j} = 0$, none of the bounding boxes in $\mathcal{D}_t$ should be connected to o_j. Similarly, $\sum_{j=1}^{|\mathcal{O}_t|} a_{ij} = 0$ indicates that the bounding box d_i does not belong to any objects in O_t. In this case, d_i can be a new object, or the object ID that d_i belongs to is missing in currently tracked objects.

We define association knowledge $\mathcal{K}$ as a metric function that takes appearance features, motion features, or both of them as input and outputs the similarity score,

$$s_{ij} = \mathcal{K}(i, j, \mathcal{F}_a, \mathcal{F}_m) \quad (2)$$

where $\mathcal{F}_a = \{f_a(d_1), \cdots, f_a(d_{|\mathcal{D}_t|}), f_a(o_1), \cdots, f_a(o_{|\mathcal{O}_t|})\}$ is the joint set of appearance features from both detections and existing tracklets, and $\mathcal{F}_m$ is similar but contains motion features. In practice, appearance features are widely represented by the Re-ID features, while motion features usually contain geometric information such as the locations and the sizes of bounding boxes[1, 9].

In early literature, the association knowledge $\mathcal{K}$ is commonly modeled with human-designed policies. For instance, a simple policy is to only consider motion cues $\mathcal{F}_m$, ignoring appearance cues $\mathcal{F}_a$. Specifically, the bounding boxes belonging to the same object ID in two adjacent frames should be closer than those belonging to different object IDs. Based on this observation, we use the IoU of the bounding boxes as the association score (Fig. 4(a)). Another simple yet effective human-designed policy is to use the cosine similarity between the Re-ID features as the association score. Similarly, the cosine similarity belonging to the same ID has a larger value than that computed from Re-ID features extracted from different identities (Fig. 4(b)).

Human-designed policies are sub-optimal as it is difficult for them to take full advantage of both appearance and motion cues. Beyond human-designed policies, the more recent arts[1, 2, 9, 18] attempt to learn association knowledge directly from data with a parametric model, i.e., $s_{ij} = \mathcal{K}_{\boldsymbol{\theta}}(i, j, \mathcal{F}_a, \mathcal{F}_m)$. As illustrated in Fig. 4(c), both $\mathcal{F}_a$ and $\mathcal{F}_m$ are taken as input by the association model, and the model learns its parameter $\boldsymbol{\theta}$ by applying stochastic gradient descent (SGD) on a labelled dataset. During inference, $\mathcal{K}_{\boldsymbol{\theta}}$ output predictions with a single forward pass. The most prevalent choice of the parametric model is the graph neural network (GNN)[31]. In Section 6, we show, by empirical experiments, that it is possible to learn association knowledge from synthetic data.

5 Experiment setup

Comparison pipeline. This paper aims to compare synthetic data and real data on their effectiveness when they are used to learn association knowledge. The experimental setup is briefly illustrated in Fig. 5. The associ-

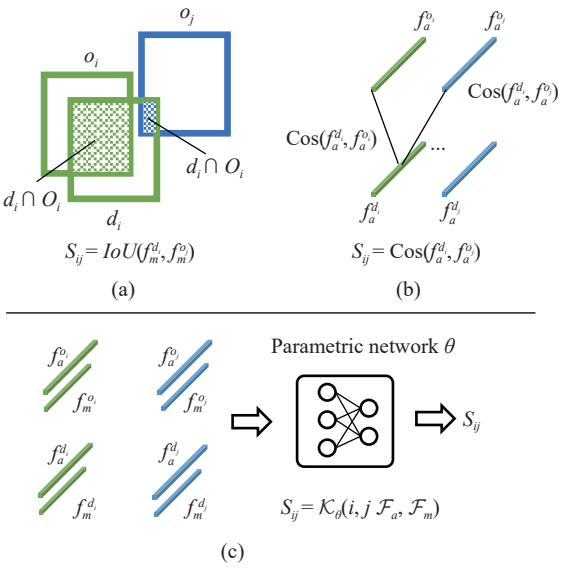


Fig. 4 Illustration of different types of association knowledge. (a) and (b) represent human-designed association knowledge. They compute the association score s_{ij} using the IoU score of bounding boxes and the cosine similarity between the Re-ID features, respectively. (c) uses the parametric network θ to predict the association score.

ation algorithm is trained on real data and synthetic data, and, finally, tested both of them in the real-world set. Note that when an algorithm involves multiple models to be trained, we only train the association-related model and keep the other models fixed. For instance, if an algorithm employs a Re-ID model, we fix the Re-ID model. During inference, we do not perform domain adaptation.

Benchmark methods. For a comprehensive comparison, we select several typical association methods, including both parametric association models and human-designed association policy. We pay more attention to parametric models as they show superior performance. Details are described as follows: MPNTracker[1] formulates MOT with the classical network flow. A type of GNN named message passing network (MPN) is proposed to predict linkages based on the graph built with appearance features and motion cues. DeepMOT[2] proposes a deep Hungarian net (DHN) as an association module to approximate the Hungarian matching algorithm. GNMOT[9] builds the appearance graph and the motion graph for two conjunctive frames. Then, two graph networks compute the similarities between nodes to achieve association. SORT[32] is a human-crafted association policy. It only employs motion cues. Observations are associated with tracklets in a hierarchical manner by comparing IoU distances. We mainly tune the key hyperparameter, IoU threshold, in the training set and use it in the test set. StrongSORT[33] proposes an appearance-free link model to associate short tracklets with complete trajectories. We use synthetic data to train this link model and then directly deploy it in the real-world testing environment.

Evaluation metric. For evaluation, we employ the widely used the widely used the clear mot metrics (CLEAR)[34] metrics. The main metrics include MOTA (MOT accuracy), IDF1 (ID F1-measure), IDSwR (identity switch rate), MT (mostly tracked target percentage), and ML (mostly lost target percentage). Among them, IDF1 and IDs are the most relevant ones to evaluate association accuracy.

6 Results and analysis

6.1 Evaluation on benchmark datasets

In this section, we show that the association knowledge learned from synthetic data works well on real-world test sets. Specifically, we use the test set of MOT-15/16/17/20[3, 4, 35]. For real data training, we use the corresponding train split of the target set, e.g., train on the MOT16 train and test on the MOT16 test. For synthetic data training, we build a synthetic training set, use this single set for training, and evaluate in all test sets. We name the synthetic dataset MOTX-S. MOTX-S is synthesized using the MOTX engine, consisting of 22 videos in total. Videos are generated by roughly simulating the scene dynamics (camera moving, camera view, person density, person velocity, etc.) of videos in the MOT15-17 dataset. As shown in Section 3.1, the resulting synthetic videos yield consistently good results even when the parameters of some scene dynamics vary in a relatively large range.

Association knowledge from the synthetic

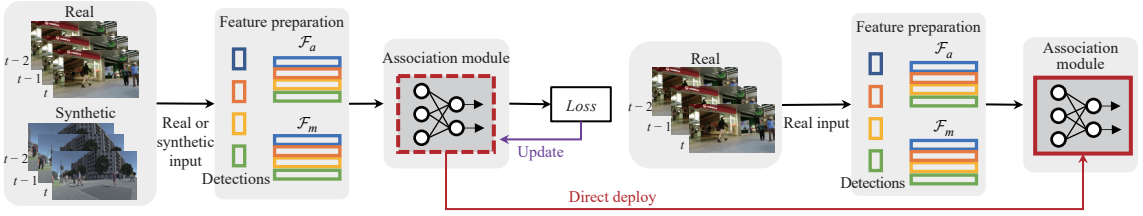


Fig. 5 Comparison pipeline. (a) For each association module, we train or tune two association modules with synthetic data and real data, respectively. (b) We then directly deploy the two association modules on a real-world test set and compare their performance. $\mathcal{F}_a$ and $\mathcal{F}_m$ are appearance features and motion features, respectively.

world is competitive. For the generality of the results, we test with multiple different association algorithms. Results are reported in Table 1. The major observation is that each association method trained on synthetic data can have a similar performance to that trained on real-world training data in terms of all metrics. Note that when training MPNTracker, MOTX-S shows its advantage over MOT15. Specifically, MOTX-S improves IDF1, MT, and ML with 0.5%, 3.1%, and 0.7%, respectively. It suggests that the association scenarios in MOTX give better supervision on association knowledge learning than MOT15. For all comparisons, we do not observe a noticeable performance drop when trained on MOTX-S. In most cases, the performance gap between MOTX and the real-world data is less than 1% for all evaluation indexes. In the experiment of StrongSORT, where the learned association model is pure motion dependent and appearance-free, it achieved slightly better testing accuracy by using MOTX-S than real-world videos. On the one hand, the above observations suggest that the association knowledge learned from synthetic data can achieve similar performance compared with that trained on real-world data. On the other hand, such competitiveness of synthetic data can not be seen in "Appearance-centered tasks" if the deep system is only learned from the synthetic data. Because of the superior performance and run-time efficiency of MPNTracker, experiments in Section 3.1 are conducted on it.

Association domain gap exists. We train MPNTracker on the training set of MOT15, MOT17, and their combination, respectively. Testing results on the MOT17 test set are shown in Table 2. Both MOT15 and the combined set are worse than using MOT17 alone. Specifically, MOT15 gets 3% lower IDF1 and about 25% higher ID switches.

A similar degeneration trend can also be found when deploying the association knowledge from MOT17 into the MOT15 domain. This suggests that there is a domain gap between association scenarios in MOT15 and MOT17.

Table 1 Comparing synthetic data (MOTX-S) and real data in association knowledge learning on real-world test sets. The numbers in bold denote that association knowledge learned from synthetic data is superior or equal to that learned from real data, while underlined mean that the performance gaps are less than 1.0.

Test	Train	MOTA ↑	IDF1 ↑	IDSw (%/#) ↓	MT ↑	ML ↓
			DeepMOT			
MOT16	MOT16-train	54.8	53.4	11.4	19.1	37.0
	MOTX-S	54.4	53.2	12.1	**19.2**	37.2
MOT17	MOT17-train	53.7	53.8	34.7	19.4	36.6
	MOTX-S	53.4	52.9	36.4	**19.7**	**36.6**
			GNMOT			
MOT16	MOT16-train	58.4	54.8	23.3	27.3	23.2
	MOTX-S	**58.4**	54.5	23.6	**27.3**	23.3
MOT17	MOT17-train	56.9	53.9	72.2	25.9	25.6
	MOTX-S	56.8	53.6	73.0	**26.1**	25.7
			MPNTracker			
MOT15	MOT15-train	51.5	58.6	5.8	31.2	25.9
	MOTX-S	51.3	**59.1**	**5.8**	**34.3**	**25.2**
MOT17	MOT17-train	58.8	61.7	6.0	28.3	33.5
	MOTX-S	58.4	61.0	6.1	28.1	33.8
			SORT			
MOT15	MOT15-train	42.6	50.8	7.27	11.2	37.6
	MOTX-S	**42.6**	50.4	7.25	**11.2**	**37.6**
MOT17	MOT17-train	43.1	39.8	–	12.5	42.3
	MOTX-S	**54.8**	**59.5**	–	**16.4**	**40.3**
			StrongSORT			
MOT17	MOT17-train	79.6	79.5	1 194	53.6	13.9
	MOTX-S	**79.6**	79.2	**1 176**	**54.3**	**14.0**
MOT20	MOT20-train	73.8	77.0	770	62.1	14.9
	MOTX-S	**73.9**	**77.0**	**753**	**62.2**	**14.8**

Table 2 Cross domain evaluation. Bold numbers denote the best results.

Test	Train	MOTA ↑	IDF1 ↑	# IDS ↓
MOT15-test	MOT15	**51.5**	58.6	**375**
	MOT17	50.9	58.8	381
	MOT15+MOT17	51.3	58.9	382
	MOTX-S	51.3	**59.1**	377
MOT17-test	MOT17	**58.8**	**61.7**	**1 185**
	MOT15	57.9	58.7	1 481
	MOT17+MOT15	58.3	60.8	1 267
	MOTX-S	58.4	61.0	1 214

Appearance domain adaptation is not necessary. We attempt to reduce the appearance domain gap between synthetic data and real-world data by converting the appearance of detections in MOTX-S into the real-world style by using a generative network using the similarity preserving generative adversarial network (SPGAN)[36]. SPGAN is trained on data provided by VisDA2020[3], which has both Unity-based synthetic persons and real-world persons. The results in Table 3 show that MOTX-S is still competitive without a domain adaptation on appearance.

6.2 Ablation study on appearance and motion features

It is worthwhile to investigate why the competitive results in Table 1 can be achieved by only using synthetic data with a considerable domain gap in image-style. We conduct the ablation study on the input of the association model. Specifically, we eliminate the effect of appearance features $\mathcal{F}_a$ or motion features $\mathcal{F}_m$ in (2) by replacing them with dummy vectors $\mathbf{1} = (1, \cdots, 1)^{\mathrm{T}}$.

Videos {2, 10, 13} in MOT17 are divided as the validation set, and the rest videos in MOT17 make up the training set. We repeat each training on MPNTracker five times and report their means. We also perform hypothesis testing to validate the statistical significance of the results. The results are shown in Fig. 6.

Effectiveness of appearance features and motion features. The tracking performance degenerates when we eliminate either appearance features or motion features. It shows that both appearance features and motion features contribute to association knowledge learning. When training on both appearance & motion features, MOTX-S achieves similar performance on the MOT17 validation set. This is consistent with the conclusion in Section 6.1.

Synthetic VS. real on motion features. When only motion features are used (w/o A), MOTX-S shows a considerable advantage over real data. In detail, the ID switch for MOTX-S is only half of that for real data. IDF1 score also leads by over 6%. This performance gap is not observed in experiments "A+M" and "w/o M". This phenomenon suggests that motion scenarios generated with MOTX can simulate the real-world association scenarios well.

[3] http://ai.bu.edu/visda-2020/

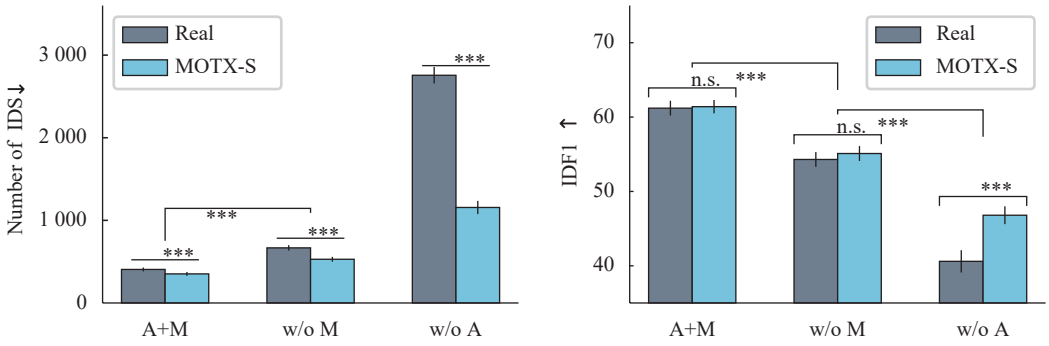
Table 3 Impact of appearance domain adaptation for pedestrians. The best results are in bold.

Test	Train	MOTA ↑	IDF1 ↑	# IDS ↓
MOT17-train	MOTX-S	**64.1**	**68.9**	**551**
	MOTX-S + SPGAN	63.7	68.2	604
MOT15-train	MOTX-S	**53.1**	**67.9**	**78**
	MOTX-S + SPGAN	52.5	66.6	**78**

Synthetic VS. real on appearance features. Intuitively, it is highly possible that the domain gap of the appearance feature harms association learning. This is because appearance models are trained on real-world Re-ID datasets, but in training association models, they are used to extract features of synthetic person images. Moreover, the final test set consists of real-world videos. However, we do not observe the expected performance drop due to the appearance domain gap. See results "w/o M" in Fig. 6, with appearance cues only, trained on real data, and synthetic data perform almost equally with similar IDs and IDF1. This suggests a somehow surprising finding: The appearance domain gap hardly harms the learning of association knowledge.

Quantitative analysis of using MOTX-S as supplements. We consider using MOTX-S to augment real-world training data. Quantitative experiments are conducted to analyze the impact of appearance features and motion features provided by MOTX-S in association knowledge learning. We gradually add videos in MOTX-S into the MOT17 training set. The results on the MOT17 validation set are shown in Fig. 7. We have two observations. The major one is that both appearance and motion features from MOTX-S are effective in augmenting real-world features. The IDF1 scores are improved by 1.4% and 6.5%, respectively. Second, when only motion features are available to learn association, using MOTX-S as supplements can boost the testing tracking accuracy. We can conclude that MOTX has expertise in providing motion-related association domain knowledge.

Discussion. The above insightful findings suggest that it is not necessary to perform additional appearance adaptation techniques when we deploy the learned association knowledge in the real-world test sets. Also, we observe that our synthetic data show stronger competitiveness in Fig. 6 than that in Table 1. The possible reason is that the training set and the test set in the MOTChallenge benchmark overlap in association scenarios, i.e., videos in the test set are collected at the same location as the training set where the camera-related and pedestrian-related factors are very close.



(a) MOT17 validation set (MPNTracker) (b) MOT17 validation set (MPNTracker)

Fig. 6 Ablation study on appearance features & motion features. A+M: With both appearance and motion features; w/o A: Without appearance features; w/o M: Without motion features. "n.s." means that the difference is {not statistically significant} (i.e., p-value > 0.05). ** and * * * mean is {statistically significant} (i.e., 0.001 < p-value < 0.01) and {statistically extremely significant} (i.e., p-value < 0.001), respectively.

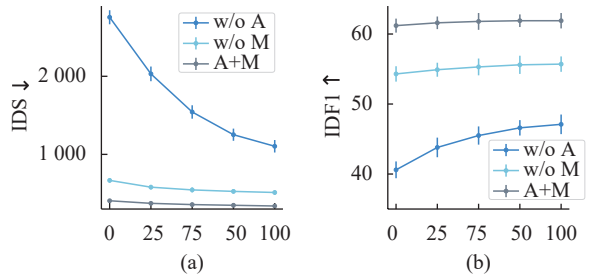


Fig. 7 Using MOTX-S as supplements. The horizontal axis is the percentage of the data in MOTX-S merged with the real data. Experiments are conducted on MPNTracker and the MOT17 validation set. The results of (a) IDS and (b) IDF1 are reported.

6.3 Investigation of controllable factors

Another major advantage of synthetic data is that we can control multiple factors in generating videos. Therefore, it is possible to conduct a thorough investigation into how these controllable factors impact an association algorithm with the help of synthetic data. In this section, we mainly study the influence of four factors, i.e., camera view, camera moving state, pedestrian speed, and pedestrian density. For each factor, we design a group of contrast experiments using different training sets and test sets. A summary of the used datasets is illustrated in Table 4. A principle of these experiments is that we train an identical association model (here we use MPNTracker) with different customized synthetic data (e.g., camera view high VS. low) and test on different real data (both camera view high VS. low). The results are shown in Figs. 8 and 9.

Dataset notation. For clarification, datasets are notated in the format prefix − middle − suffix. The prefix can be "S" and "R", representing synthetic data or real data. The middle word is the controllable factor to be studied, e.g., "Cam" indicates the camera. The suffix is the value of the controllable factor. For instance, "S-Cam-H" represents this dataset consisting of synthetic videos with high camera views.

Camera view. The association models are trained on S-Cam-H, S-Cam-L, and their compound version {S-Cam-H, S-Cam-L}, respectively. Then the trained association models are tested on real-world videos with high camera views (R-Cam-H) or low camera views (R-Cam-L). According to Figs. 8(a)–8(d), a major observation is that association knowledge learning is sensitive to camera view. Specifically, when testing on R-Cam-H, the association model trained on S-Cam-H can achieve a close ID switch rate and IDF1 score compared with the model trained on the compound data. However, the accuracy of only using S-Cam-L decreases noticeably in this case shown in Figs. 8(a) and 8(b). We observe a similar trend when testing on videos with low camera view in Figs. 8(c) and 8(d). This suggests that the knowledge learned from high camera views can not be deployed in low camera view test environment successfully, and vice versa. In other words, there is an obvious association domain gap between high camera view scenarios and low camera view scenarios.

Camera moving state. We learn association knowledge from static cameras (S-Cam-S) and moving cameras (S-Cam-M) and their combination ({S-Cam-S, S-Cam-M}). The real-world videos in two test sets are selected from MOTchallenge. One constrains cameras in videos as static (R-Cam-S), and cameras in another are moving (R-Cam-M). The results are shown in Figs. 8 (e)–8(h). The same trend with camera view is that the camera moving state can also bias the association knowledge learning. For instance, Figs. 8(e) and 8(f) show that S-Cam-S has the advantage in testing videos with static cameras. Similarly, the model trained on S-Cam-M obtains better results on R-Cam-M than that trained on S-Cam-S Figs. 8(g) and 8(h). So, we can conclude that the moving state of cameras can cause the domain gap in association scenarios. Also, we observe an obvious performance increase in IDS and IDF1 if we combine S-Cam-M with S-Cam-S in Figs. 8(e) and 8(f). However, such a trend is not observed in Figs. 8(g) and 8(h), where the test set is R-

Table 4 Notations for four groups of data to study motion factors. The prefix "S" and "R" represent synthetic data and real data, respectively. The suffix "H", "L", "S", and "M" stand for high, low, static, and moving.

#		Notation	Description
1	Train	S-Cam-H	Camera view: High (surveillance view)
		S-Cam-L	Camera view: Low (vehicle view)
	Test	R-Cam-H	Video #04 in MOT17
		R-Cam-L	Video #02, #09 in MOT17
2	Train	S-Cam-S	Camera state: Static
		S-Cam-M	Camera state: Moving
	Test	R-Cam-S	Video #02, #04, #09 in MOT17
		R-Cam-M	Video #10, #11, #13, in MOT17
3	Train	S-Speed-n	Pedestrian speed: n m/s, $n \in \{1, 2, 4, 6\}$
	Test	R-Speed-H	KITTI-17, KITTI-13, PETS09-S2L1 in MOT15
		R-Speed-L	Venice-2, ADL-Rundle-8, ADL-Rundle-6 in MOT15
4	Train	S-Density-n	The number of persons in a frame, $n \in \{10, 20, 40\}$
	Test	R-Density-L	PETS09-S2L1, TUD-Stadtmitte, TUD-Campus, KITTI-17, KITTI-13 in MOT15
		R-Density-H	Video #02, #04 in MOT17

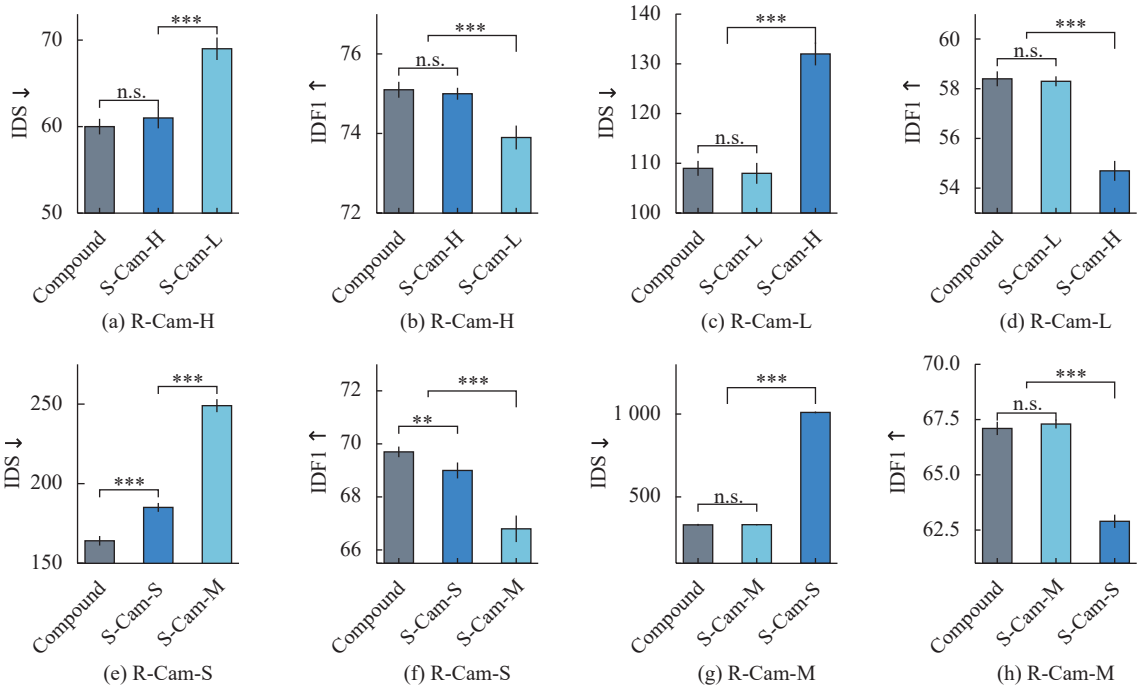


Fig. 8 Impact of motion factors related to the camera. (a)–(d) illustrate how the learning system reacts to the change of camera view. Similarly, (e)–(h) give the result of changing the camera's moving state. All experiments are tested on selected real-world videos. Notations "n.s.", **, and *** have the same meaning as those in Fig. 6.

Cam-M. This insightful discovery implies that the association knowledge learned from moving cameras has stronger compatibility than that learned from static cameras.

Pedestrian speed. Association models are trained on S-Speed-n, $n \in \{1, 2, 4, 6\}$, which means pedestrian speed is n m/s. The test sets are R-Speed-L and R-Speed-H. In detail, the frame rate of videos in R-Speed-H ranges from 7 fps to 10 fps. It means that the moving speed of the same identity between two conjunctive frames is almost 3–4 times that in R-Speed-L, where the video frame rate

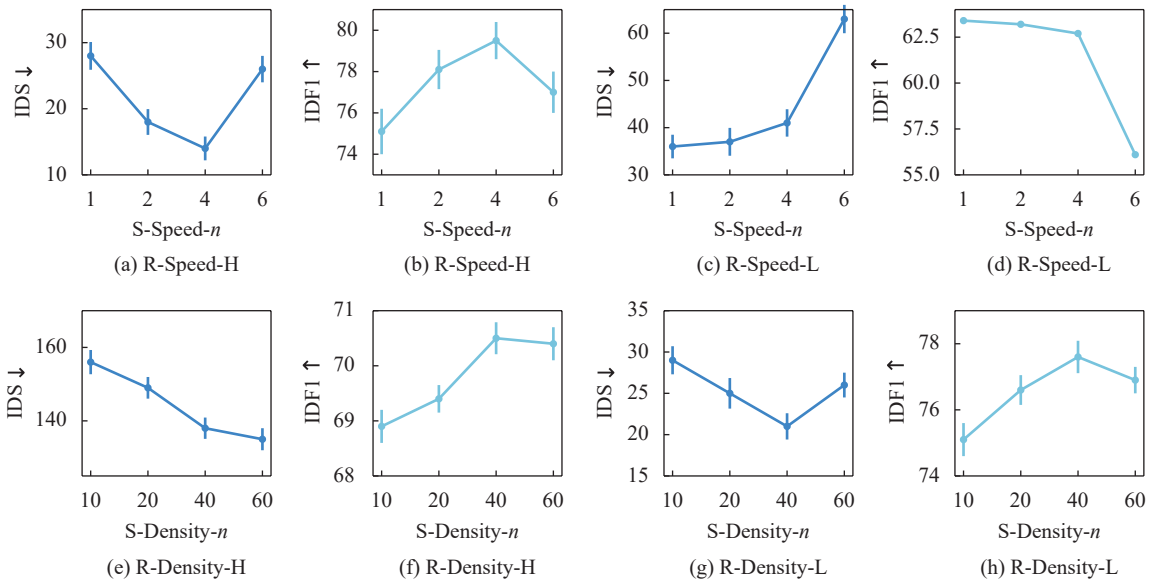


Fig. 9 Impact of pedestrian-related motion factors. (a)– (d) illustrate how the learning system reacts to the pedestrian speed. (e)–(h) gives the result of changing the pedestrian density. All testing videos are real-world videos.

is around 30 fps. According to Figs. 9(a)–9(d), we have two observations. The major one is that if the pedestrian speed in the training set mismatches speeds in the test environment, the performance decreases. When testing on R-Speed-H, the number of IDs doubled (14.2 → 28.6) if the pedestrian speed is changed from 4 m/s to 1 m/s. The IDF1 score also degenerates obviously (79.5% → 75.1%). A similar trend can be found when testing on R-Speed-L. Accelerating the pedestrians' speed to large values (1 m/s → 6 m/s) can significantly increase the number of IDs and decrease IDF1s.

Pedestrian density. Figs. 9(e)–9(h) show results for testing on real videos with different pedestrian densities. According to the official statistics of MOTChallenge, the average pedestrian density of all videos in our build R-Density-L is less than 10. However, for R-Density-L, the optimal density in the training set is 40 according to Figs. 9(g) and 9(h). It suggests that the gap in pedestrian density does not cause the gap in tracking performance if the testing environment has low pedestrian density. For example, S-Density-60 is better than S-Density-10 when testing on R-Density-L. However, association knowledge gained from low-density videos is not very effective in high-density environments (Figs. 9(e) and 9(f)).

Discussion. Our synthetic dataset is manually configured in MOTX. We do so by setting the motion-related parameters to roughly match the real training videos. The above experiment also serves as a confirmation that this manual configuration process is stable. For example, when the pedestrian speed is set between 1–2 m/s, the IDS scores remain stable. The same observation also goes for other factors like pedestrian density. Therefore, in practice, we advise giving a possibly best manual estimation of the motion parameters of the testing environment. Relatively small errors can be well-tolerated, but large errors (e.g., the camera speed is estimated to be 1 m/s but is actually static) should be avoided.

6.4 Tuning human-designed policy w.r.t. scenes

Our synthetic dataset can also benefit the hyper-parameter search for the given scenes. We take the SORT[32] algorithm as an example. When R-Speed-H is the testing scenario, we synthesize a dataset according to the roughly estimated motion factors in R-Speed-H and search for the hyper-parameter IoU threshold. As shown in Table 5, compared with using MOT15 for the hyper-parameter search, the IoU threshold searched from the synthetic data is closer to that searched from fully labeled R-Speed-H.

Table 5 Hyper-parameter tuning for the SORT algorithm. We report the best hyper-parameter for the given dataset.

Dataset	S-Speed-H	MOT15-train	R-Speed-H
IoU threshold (0–1)	0.25	0.2	0.3

7 Conclusions

This paper studies the role of synthetic data in multi-object tracking. Crediting to the proposed MOTX engine, we make two contributions. First, we show that association knowledge obtained from synthetic data can be directly deployed in the real-world environment without domain adaptation, even if the image-style discrepancy between synthetic data and real-world data exists. Second, with the help of the MOTX engine, we thoroughly investigate how association knowledge reacts to

changes in camera-related and pedestrian-related motion factors. Experimental results lead to intriguing finds giving new insights into understanding the impact of data in association knowledge learning.

Declarations

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgements

This work was supported by the ARC Discovery Early Career Researcher Award, China (No. DE200101283) and the ARC Discovery Project, China (No. DP210102801).

Open Access

References

[1] G. Brasó, L. Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6246–6256, 2020. DOI: 10.1109/CVPR42600.2020.00628.

[2] Y. H. Xu, A. Ŝep, Y. T. Ban, R. Horaud, L. Leal-Taixé, X. Alameda-Pineda. How to train your deep multi-object tracker. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6786–6795, 2020. DOI: 10.1109/CVPR42600.2020.00682.

[3] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. [Online], Available: https://arxiv.org/abs/1504.01942, 2015.

[4] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler. MOT16: A benchmark for multi-object tracking. [Online], Available: https://arxiv.org/abs/1603.00831, 2016.

[5] S. Bąk, P. Carr, J. F. Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 193–209, 2018. DOI: 10.1007/978-3-030-01261-8_12.

[6] H. Z. Dou, W. H. Zhang, P. Z. Zhang, Y. H. Zhao, S. Y. Li, Z. Q. Qin, F. Wu, L. Dong, X. Li. VersatileGait: A large-scale synthetic gait dataset with fine-grained attributes and complicated scenarios. [Online], Available: https://arxiv.org/abs/2101.01394, 2021.

[7] Z. F. Xue, W. J. Mao, L. Zheng. Learning to simulate complex scenes. [Online], Available: https://arxiv.org/abs/2006.14611, 2020.

[8] Y. Yao, L. Zheng, X. D. Yang, M. Naphade, T. Gedeon. Simulating content consistent vehicle datasets with attribute descent. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 775–791, 2020. DOI: 10.1007/978-3-030-58539-6_46.

[9] J. H. Li, X. Gao, T. T. Jiang. Graph networks for multiple object tracking. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, Snowmass, USA, pp. 708–717, 2020. DOI: 10.1109/WACV45572.2020.9093347.

[10] Z. D. Wang, L. Zheng, Y. X. Liu, Y. L. Li, S. J. Wang. Towards real-time multi-object tracking. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 107–122, 2020. DOI: 10.1007/978-3-030-58621-8_7.

[11] N. Wojke, A. Bewley, D. Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of IEEE International Conference on Image Processing*, Beijing, China, pp. 3645–3649, 2017. DOI: 10.1109/ICIP.2017.8296962.

[12] Y. F. Zhan, C. Y. Wang, X. G. Wang, W. J. Zeng, W. Y. Liu. A simple baseline for multi-object tracking. [Online], Available: https://arxiv.org/abs/2004.01888v1, 2020.

[13] Z. W. Zhou, J. L. Xing, M. D. Zhang, W. M. Hu. Online multi-target tracking with tensor-based high-order graph matching. In *Proceedings of the 24th International Conference on Pattern Recognition*, IEEE, Beijing, China, pp. 1809–1814, 2018. DOI: 10.1109/ICPR.2018.8545450.

[14] J. Zhu, H. Yang, N. Liu, M. Kim, W. J. Zhang, M. H. Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 379–396, 2018. DOI: 10.1007/978-3-030-01228-1_23.

[15] Q. C. Wang, Y. H. Gong, C. H. Yang, C. H. Li. Robust object tracking under appearance change conditions. *International Journal of Automation and Computing*, vol. 7, no. 1, pp. 31–38, 2010. DOI: 10.1007/s11633-010-0031-9.

[16] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, 1955. DOI: 10.1002/nav.3800020109.

[17] G. Welch, G. Bishop. An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Chapel Hill, USA, 1995.

[18] I. Papakis, A. Sarkar, A. Karpatne. GCNNMatch: Graph convolutional neural networks for multi-object tracking

via Sinkhorn normalization. [Online], Available: https://arxiv.org/abs/2010.00067, 2020.

[19] X. C. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Q. Wang, K. Saenko. VisDA: The visual domain adaptation challenge. [Online], Available: https://arxiv.org/abs/1710.06924, 2017.

[20] X. C. Peng, B. Usman, N. Kaushik, D. Q. Wang, J. Hoffman, K. Saenko. VisDA: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, Salt Lake City, USA, pp. 2021–2026, 2018. DOI: 10.1109/CVPRW.2018.00271.

[21] Y. Cabon, N. Murray, M. Humenberger. Virtual KITTI 2. [Online], Available: https://arxiv.org/abs/2001.10773, 2020.

[22] A. Gaidon, Q. Wang, Y. Cabon, E. Vig. Virtual Worlds as proxy for multi-object tracking analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 4340–4349, 2016. DOI: 10.1109/CVPR.2016.470.

[23] Y. Z. Hou, L. Zheng, S. Gould. Multiview detection with feature perspective transformation. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 1–18, 2020. DOI: 10.1007/978-3-030-58571-6_1.

[24] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, R. Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 450–456, 2018. DOI: 10.1007/978-3-030-01225-0_27.

[25] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, R. Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 3752–3761, 2018. DOI: 10.1109/CVPR.2018.00395.

[26] C. Doersch, A. Zisserman. Sim2real transfer learning for 3D human pose estimation: Motion to the rescue. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, Vancouver, Canada, 2019.

[27] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. K. Zhu, A. Kembhavi, A. Gupta, A. Farhadi. AI2-THOR: An interactive 3D environment for visual AI. [Online], Available: https://arxiv.org/abs/1712.05474, 2017.

[28] A. Kar, A. Prakash, M. Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, S. Fidler. Meta-Sim: Learning to generate synthetic datasets. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 4550–4559, 2019. DOI: 10.1109/ICCV.2019.00465.

[29] A. Juliani, V. P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, D. Lange. Unity: A general platform for intelligent agents. [Online], Available: https://arxiv.org/abs/1809.02627, 2018.

[30] X. X. Sun, L. Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 608–617, 2019. DOI: 10.1109/CVPR.2019.00070.

[31] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.

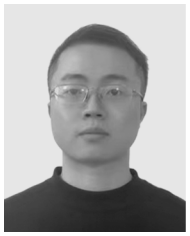
[32] A. Bewley, Z. Y. Ge, L. Ott, F. Ramos, B. Upcroft. Simple online and realtime tracking. In *Proceedings of IEEE International Conference on Image Processing*, Phoenix, USA, pp. 3464–3468, 2016. DOI: 10.1109/ICIP.2016.7533003.

[33] Y. H. Du, Y. Song, B. Yang, Y. Y. Zhao. StrongSORT: Make deepSORT great again. [Online], Available: https://arxiv.org/abs/2202.13514, 2022.

[34] K. Bernardin, R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, vol. 2008, Article number 246309, 2008.

[35] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, L. Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. [Online], Available: https://arxiv.org/abs/2003.09003. 2020.

[36] W. J. Deng, L. Zheng, Q. X. Ye, G. L. Kang, Y. Yang, J. B. Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 994–1003, 2018. DOI: 10.1109/CVPR.2018.00110.



Yuchi Liu received the B.Eng. degree in software engineering from Australian National University, Australia in 2018. He is currently a Ph. D. degree candidate in computer science at Australian National University, Australia.

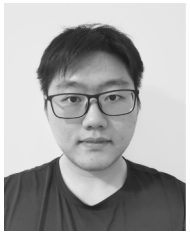
His research interests include video object tracking, learning from synthetic data, and weakly supervised learning.

E-mail: yuchi.liu@anu.edu.au
ORCID iD: 0000-0001-9061-6180

Zhongdao Wang received the B. Sc. degree in physics from Department of Physics Tsinghua University, China in 2017. He is currently a Ph. D. degree candidate in electronic engineering at Department of Electronic Engineering, Tsinghua University, China.

His research interests include perception algorithms for autonomous driving, including but not limited to 3D object detection/tracking, network architecture/learning algorithm/pre-training for multi-modal fusion, and 4D Auto-labeling.

E-mail: wcd17@mails.tsinghua.edu.cn
ORCID iD: 0000-0002-4483-8783



Xiangxin Zhou received the B.Sc. degree in electronic engineering from Department of Electronic Engineering, Tsinghua University, China in 2021. He is currently a Ph. D. degree candidate in artificial intelligence at School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS), China, and Institute of Automation, Chinese Academy of Sciences (CASIA), China.

His research interests include geometric deep learning, graph neural networks for drug design, causal inference, and multimodal machine learning.

E-mail: xx-zhou16@mails.tsinghua.edu.cn
ORCID iD: 0000-0002-1526-0548

Liang Zheng received the B. Eng. degree in life science from Tsinghua University, China in 2010, and the Ph. D. degree in electronic engineering from Tsinghua University, China in 2015. He is a lecturer and a computer science futures fellowship in School of Computer Science, Australian National University, Australia.

His research interests include computer vision, machine learning, object re-identification and dataset-centered vision.

E-mail: liang.zheng@anu.edu.au (Corresponding author)
ORCID iD: 0000-0002-1464-9500