A Two-Stage Minimum Cost Multicut Approach to Self-Supervised Multiple Person Tracking

Kalun Ho[1,2,3], Amirhossein Kardoost[3], Franz-Josef Pfreundt[1,2], Janis Keuper[4], and Margret Keuper[3]

[1] Fraunhofer Center Machine Learning, Germany
[2] CC-HPC, Fraunhofer ITWM, Kaiserslautern, Germany
[3] Data and Web Science Group, University of Mannheim, Germany
[4] Institute for Machine Learning and Analytics, Offenburg University, Germany

Abstract. Multiple Object Tracking (MOT) is a long-standing task in computer vision. Current approaches based on the tracking by detection paradigm either require some sort of domain knowledge or supervision to associate data correctly into tracks. In this work, we present a self-supervised multiple object tracking approach based on visual features and minimum cost lifted multicuts. Our method is based on straightforward spatio-temporal cues that can be extracted from neighboring frames in an image sequences without supervision. Clustering based on these cues enables us to learn the required appearance invariances for the tracking task at hand and train an AutoEncoder to generate suitable latent representations. Thus, the resulting latent representations can serve as robust appearance cues for tracking even over large temporal distances where no reliable spatio-temporal features can be extracted. We show that, despite being trained without using the provided annotations, our model provides competitive results on the challenging MOT Benchmark for pedestrian tracking.

1 Introduction

The objective of multiple object tracking is to find a trajectory for each individual object of interest in a given input video. Specific interest has been devoted to the specific task of multiple person tracking [1–5]. Most successful approaches follow the *Tracking-By-Detection* paradigm. First, an object (pedestrian) detector is used in order to retrieve the position of each person within each frame. Secondly, the output detections of same persons across video frames are associated over space and time in order to form unique trajectories. Since objects might get occluded during the video sequence or the detector might simply fail on some examples, successful approaches are usually based not solely on spatial but also on appearance cues. These are learned from annotated data, for example using Siamese networks for person re-identification [4].

Motivation. Supervised approaches for person re-identification require large amounts of sequence specific data in order to achieve good performance. For this

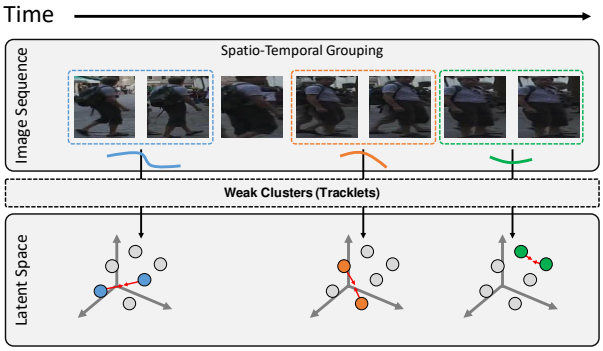

Fig. 1: Given an image sequence, many data associations can be made reliably from pure spatio-temporal cues such as the intersection over union of bounding boxes. These associations are injected into a convolutional AutoEncoder to enforce detections with the same, spatio-temporally determined label to be close to one-another in the latent space. Thus, the learned appearance features will generalize over viewpoint and pose variations.

reason, multiple object tracking benchmarks such as MOT [6] are providing a training sequence recorded in a sufficiently similar setting for every test sequence. The results of our experiments in table 1 confirm this dependency and show the high variance in the quality of supervised approaches, depending on the data used for training. The standard approach to solve this problem is to incorporate additional annotated training data. For example, [7, 8] showed that additional data is key to improving the overall tracking performance.

Thus, publicly available, annotated training data currently seems not to be sufficient for training reliable person re-identification networks. Furthermore, recording and labeling sufficient data in a setting close to a final test scenario usually comes at a high price. Hence, the need for methods with a low amount of supervision becomes obvious and motivates us to propose a multiple object tracking method based on self-supervision.
While self-supervised learning methods [9] have been successfully exploited in other vision tasks [10–15], a direct application to tracking is non-trivial: Learning suitable object appearance metrics for object tracking in a self-supervised way is challenging since, compared to classical clustering problems, visual features of the same person may change over time due to pose and viewpoint changes and partial occlusion. Other issues, such as frequent and long range full occlusion or background noises, makes pedestrian tracking even more challenging.

In this paper, we propose an approach for learning appearance features for multiple object tracking without utilizing human annotations of the data. Our approach is based on two observations: I) given an image sequence, many data associations can be made reliably from pure spatio-temporal cues such as the intersection over union (IoU) of bounding boxes within one frame or between neighboring frames. II) Resulting tracklets, carry important information about

		Train (Supervised)						
		MOT-02	MOT-04	MOT-05	MOT-09	MOT-10	MOT-11	MOT-13
Test	MOT-02	**100.0**	-0.3	-0.3	-19.2	-9.1	-12.5	-9.5
	MOT-04	0.0	**100.0**	0.0	-19.3	-4.9	-11.5	-4.9
	MOT-05	-0.6	-1.2	**100.0**	-3.2	-5.1	-3.4	-5.1
	MOT-09	-0.2	-0.4	-0.2	**100.0**	-2.9	-0.5	-2.5
	MOT-10	0.8	0.6	1.2	0.6	**100.0**	0.6	0.4
	MOT-11	0.0	-0.2	-0.2	0.2	-1.2	**100.0**	-1.4
	MOT-13	0.4	-1.1	-0.4	-3.8	-0.8	-2.7	**100.0**

Table 1: Results for training with one training sequence using *GT annotations*[1] for the tracklet generation, and evaluating on other training sequences with different viewpoints and resolutions. This table shows the relative MOTA changes for non-matching sequences on MOT17, FRCNN in comparison to the baseline (bold). Columns represent the training sequence, rows the test sequence. The tracking performance heavily depends on the employed training data and can become unstable across domains.

the variation of an object's appearance over time, for example by changes of the pose or viewpoint. In our model, we cluster the initial data based on simple spatial cues using the recently successful minimum cost multicut approach [3]. The resulting clustering information is then injected into a convolutional AutoEncoder to enforce detections with the same, spatio-temporally determined label to be close to one-another in the latent space (see Fig.1). Thus, the resulting latent data representation is encoding not only the pure object appearance, but also the expected appearance variations within one object ID. Distances between such latent representations can serve to re-identify objects even after long temporal distances, where no reliable spatio-temporal cues could be extracted. We use the resulting information in the minimum cost lifted multicut framework, similar to the formulation of Tang [4], whose method is based on Siamese networks trained in a fully supervised way.

To summarize, our contributions are:

- We present an approach for multiple object tracking, including long range connections between objects, which is completely supervision-free in the sense that no human annotations of person IDs are employed.
- We propose to inject spatio-temporally derived information into convolutional AutoEncoder in order to produce a suitable data embedding space for multiple object tracking.
- We evaluate our approach on the challenging MOT17 benchmark and show competitive results without using training annotations.

The rest of the paper is structured as follows: Section 2 discusses the related work on multiple object tracking. Our self-supervised approach on multiple object tracking is explained in Section 3. In Section 4, we show the tracking performance of the proposed method in the MOT Benchmark [6] and conclude in Section 5.

[4] Specifically, we mine GT tracklets from the detections with IoU > 0.5 with the GT as e.g. done in [16].

2 Related Work

Multiple Object Tracking. In Multiple Object Tracking according to the *Tracking by Detection* paradigm, the objective is to associate detections of individual persons, which may have spatial or temporal changes in the video. Thus re-identification over a long range remains a challenging task. Multiple object tracking by linking bounding box detections (*tracking by detection*) was studied, e.g., in [17–23, 23–25]. These works solve the combinatorial problem of linking detections over time via different formulations e.g. via integer linear programming [26, 27], MAP estimation [17], CRFs [28], continuous optimization [18] or dominant sets [29]. In such approaches, the pre-grouping of detections into tracklets or non-maximum suppression are commonly used to reduce the computational costs [19–24, 30, 31]. For example Zamir et al. [22] use generalized minimum clique graphs to generate tracklets as well as the final object trajectories. Non-maximum suppression also plays a crucial role in disjoint path formulations, such as [32–34]. In the work of Tang et al. [3], local pairwise features based on DeepMatching are used to solve a multicut problem. The affinity measure is invariant to camera motion and thus makes it reliable for short term occlusions. An extension of this work is found in [4], where additional long range information is included. By introducing a lifted edge in the graph, an improvement of person re-identification has been achieved. Similarly, [35] uses lifted edges as an extension to the disjoint path problem. [36] exploits the tracking formulation using a Message Passing Networks (MPNs). In [37], low-level point trajectories and the detections are combined to jointly solve a co-clustering problem, where dependencies are established between the low-level points and the detections. Henschel et al. [38] solves the multiple object tracking problem by incorporating additional head detecion to the full body detection while in [39], they use a body and joint detector to improve the quality of the provided noisy detections from the benchmark. Other works that treat Multiple Object Tracking as a graph-based problem can be found in [2], [40–42] and [1]. In contrast, [43] introduces a tracklet-to-tracklet method based on a combination of Deep Neural Networks, called *Deep Siamese Bi-GRU*. The visual appearance of detections are extracted with CNNs and RNNs in order to generate a tracklet of individuals. These tracklets are then split and reconnected such that occluded persons are correctly re-identified. The framework uses spatial and temporal information from the detector to associate the tracklets. The approach in [44] exploits the bounding box information by learning from detectors first and combined with a re-identification model trained on a siamese network. While the state of the art approaches in MOT17 Challenge are all based on supervised learning [38, 45–47], there are similar works in [48–50], which attempt to solve person re-identification (ReID) problems in an unsupervised manner.

Self-supervised learning aims to generate pseudo labels automatically from a pretext task, and then employs these labels to train and solve for the actual downstream task. This is especially useful when no labeled data is available. Thus self-supervised approaches can be applied to many specific real-world problems.

An extensive review of recent methods is presented in [51]. For instance [52] uses a motion-based approach to obtain labels to train a convolutional neural network for semantic segmentation problems. Another work on self-supervision based on motion can be found in [11] The idea of Doersch et al. [53] is to predict the position of eight spatial configurations given an image pair. In [54] semantic inpainting task is solved using a context encoder to predict missing pixels of an image. Hendrycks et al. [12] use a self-supervised method to improve the robustness of deep learning models. Lee et al. [55] propose an approach to improve object detection by recycling the bounding box labels while Ye et al. [56] use a progressive latent model to learn a customized detector based on spatio-temporal proposals.

3 AutoEncoder-Based Multicut Approach

The proposed approach is based on the idea to learn, from simple spatial data associations between object detections in image sequences, which appearance variations are to be expected within one object for the task of multiple object tracking. An overview of our workflow implementing this idea is given in Fig. 2.

Stage 1. The object detection bounding boxes are extracted along with their spatial information such that spatial correspondences between detections in neighboring frames can be computed. Based on these simple spatial associations, detections can be grouped into tracklets in order to obtain cluster labels using clustering approaches such as correlation clustering, also referred to as minimum cost multicuts [57].

Stage 2. A convolutional AutoEncoder is trained to learn the visual features of detections. The objective is to learn a latent space representation which can serve to match the same object in different video frames. Thus, the information about spatial cluster labels from the first stage is used as the centroid of latent features. Distances between latent representations of data samples and their centroids are minimized in the convolutional AutoEncoder using a clustering loss.

Lastly, the data are transformed into the latent space of the trained AutoEncoder to extract pairwise appearance distances which are expected to encode the desired invariances. Such pairwise appearance distances are used to not only provide additional grouping information between nearby detections, but also for detections with long temporal distance. The final detection grouping is computed using minimum cost lifted multicuts [58].

This section is divided into three subsections: Section 3.1 describes the minimum cost (lifted) multicut approach employed for obtaining the initial spatial cluster labels (e.g. tracklets), as well as for the generation of the final tracking result. Section 3.2 describes the feature learning process using a convolutional AutoEncoder and cluster labels, and section 3.3 describes the computation of the joint spatial and appearance metrics used in the final data association step within the minimum cost lifted multicut framework.

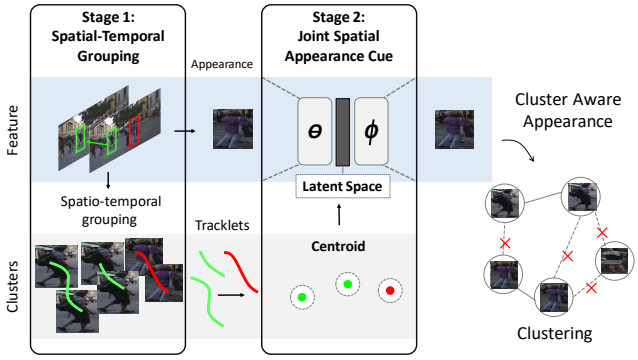

Fig. 2: Summary of our approach in two steps: 1. First, weak cluster labels (tracklets) are obtained from spatio-temporal vicinity using minimum cost multicuts [59]. 2. Then, visual features are learned by an AutoEncoder, with an additional data association loss within the tracklets. The AutoEncoder provides a stable appearance embedding while the additional loss forces detections within one tracklet to have similar embeddings. This facilitates to extract affinities between detections to compute the final tracking with re-identification using lifted multicuts [4].

3.1 Multicut Formulation

We follow Tang [4] and phrase the multiple target tracking problem as a graph partitioning problem, more concretely, as a minimum cost (lifted) multicut problem. This formulation can serve as well for an initial tracklet generation process, which will help us to inject cues learned from spatial information into the appearance features, as it can be used to generate the final tracking result using short- and long-range information between object detections.

Minimum Cost Multicut Problem. We assume, we are given an undirected graph $G = (V, E)$, where nodes $v \in V$ represent object detections and edges $e \in E$ encode their respective spatio-temporal connectivity. Additionally, we are given real valued costs $c : E \to \mathbb{R}$ defined on all edges. Our goal is to determine *edge* labels $y : E \to \{0, 1\}$ defining a graph decomposition such that every partition of the graph corresponds to exactly one object track (or tracklet). To infer such an edge labeling, we can solve instances of the minimum cost multicut problem with respect to the graph G and costs c, defined as follows [57, 59]:

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e y_e \tag{1}$$

$$s.t. \quad \forall C \in cycles(G) \quad \forall e \in C : y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \tag{2}$$

Here, the objective is simply to cut those edges with negative costs c_e such that the resulting cut is a decomposition of the graph. This condition is formalized by the cycle inequalities in Eq. (2), which make sure that, for every cycle in G, if one of its edges is cut, so is at least one other. Thus, no two nodes can remain connected via some path of the graph if an edge is cut between them along any other path. In [59], it was shown to be sufficient to enforce Eq. (2) on all *chordless* cycles, i.e. all cycles.

Typically, if cut probabilities between pairs of nodes are available, the costs are computed using the *logit* function $\text{logit}(p) = \log \frac{p}{1-p}$ to generate positive and negative costs. With these costs set appropriately, the optimal solution of minimum cost multicut problems not only yields an optimal cluster assignment but also estimates the number of clusters (e.g. objects to track) automatically.

While the plain minimum cost multicut problem has shown good performance in multiple object tracking scenarios with only short range information available [3], the cost function actually has a rather limited expressiveness. In particular, when we want to add connectivity cues between temporally distant bounding boxes, we can only do so by inserting a direct edge into the graph. This facilitates solutions that directly connect such distant nodes even if this link is not justified by any path through space and time. This limitation is alleviated by the formulation of minimum cost *lifted* multicuts [58].

Minimum Cost Lifted Multicut Problem. For a given, undirected graph $G = (V, E)$ and an additional edge set $F \subseteq \binom{V}{2} \setminus E$ and any real valued cost function $c : E \cup F \to \mathbb{R}$, the 01 linear program written below is an instance of the *Minimum Cost Lifted Multicut Problem (LMP)* w.r.t. G, F and c [58]:

$$\min_{y \in Y_{EF}} \quad \sum_{e \in E \cup F} c_e y_e \tag{3}$$

with $Y_{EF} \subseteq \{0,1\}^{E \cup F}$ the set of all $y \in \{0,1\}^{E \cup F}$ with

$$\forall C \in \text{cycles}(G)\ \forall e \in C :\ y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \tag{4}$$

$$\forall vw \in F\ \forall P \in vw\text{-paths}(G) :\ y_{vw} \leq \sum_{e \in P} y_e \tag{5}$$

$$\forall vw \in F\ \forall C \in vw\text{-cuts}(G) : 1 - y_{vw} \leq \sum_{e \in C} (1 - y_e) \tag{6}$$

The above inequalities Eq. (4) make sure that, as before, the resulting edge labeling is actually inducing a decomposition of G. Eq. (5) enforces the same constraints on cycles involving edges from F, i.e. so called *lifted* edges, and Eq.(6) makes sure that nodes that are connected via a lifted edge $e \in F$ are connected via some path along original edges $e' \in E$ as well. Thus, this formulation allows for a generalization of the cost function to include long range information without altering the set of feasible solutions.

Optimization. The minimum cost multicut problem (1) as well as the minimum lifted multicut problem (3) are NP-hard [60] and even APX-hard [57, 61]. Nonetheless, instances have been solved within tight bounds, e.g. in [62] using a branch-and-cut approach. While this can be reasonably fast for some, easier problem instances, it can take arbitrarily long for others. Thus, primal heuristics such as the one proposed in [58] or [63] are often employed in practice and show convincing results in various scenarios [4, 58, 64, 65].

Spatio-Temporal Tracklet Generation. Since the proposed approach is self-supervised in a sense that no annotated labels from the dataset are used in the training process, it is challenging to effectively learn such probabilities. To approach this challenge, we first extract reliable point matches between neighboring frames using DeepMatching [66] as done before e.g. in [3,4,37]. Instead of learning a regression model on features derived from the resulting point matches, we simply assume that the intersection over union (IoU) of retrieved matched within pairs of detections (denoted by $\mathrm{IoU}_{\mathrm{DM}}$) is an approximation to the true IoU. Thus, when $\mathrm{IoU}_{\mathrm{DM}} > 0.7$, we can be sure we are looking at the same object in different frames. While this rough estimation is not suitable in the actual tracking task since it clearly over-estimates the cut probability, it can be used to perform a pre-grouping of detections that definitely belong to the same person. The computation of pairwise cut probabilities used in the lifted multicut step for the final tracking task is described in section 3.3.

3.2 Deep Convolutional AutoEncoder

A convolutional AutoEncoder takes an input image, compresses it into a *latent space* and reconstructs it with the objective to learn meaningful features in an unsupervised manner. It consists of two parts: the encoder $f_\theta(.)$ and a decoder $g_\phi(.)$, where θ and ϕ are trainable parameters of the encoder and decoder, respectively. For a given input video, there are in total n detections $x_i \in X_{i=1}^n$, the objective is to find a meaningful encoding z_i, where the dimension of z_i is much lower than x_i. The used convolutional AutoEncoder first maps the input data into a latent space Z with a non-linear function $f_\theta : X \to Z$, then decodes Z to its input with $g_\phi : Z \to X$. The encoding and reconstruction is achieved by minimizing the following loss equation:

$$\min_{\theta,\phi} \sum_{i=1}^{N} L(g(f(x_i)), x_i) \tag{7}$$

where L is the least-squared loss $L(x, y) = \|x - y\|^2$. Similar to the work of [67], we add an additional clustering term to minimize the distance between learned features and their cluster center $\tilde{c_i}$ from the spatio-temporal tracklet labels.

$$\min_{\theta,\phi} \sum_{i=1}^{N} L(g(f(x_i)), x_i)\lambda + L(f(x_i), \tilde{c_i})(1 - \lambda) \tag{8}$$

Fig. 3: Nearest neighbor of the query detection (left most detection) within 46 frames with a step size of 5 frames of the sequence MOT17-09-SDP without (top) and with (bottom) the self-supervised clustering loss. Without this loss, the detections on the girl are spread over several clusters and a false association is made by the nearest neighbor. These mistakes are corrected by the clustering loss.

The parameter $\lambda \in [0, 1]$ balances between reconstruction and clustering loss. When choosing $0 < \lambda < 1$, the reconstruction part (Eq. (7)) can be considered to be a data-dependent regularization for the clustering. To compute the centroid c_i, the whole dataset is passed through the AutoEncoder once:

$$\tilde{c_i} = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{9}$$

We use a deep AutoEncoder with five convolutional and max-pooling layers for the encoder and five up-convolutional and upsample layers for the decoder, respectively. Furthermore, batch normalization is applied on each layer and initialized using Xavier Initialization [68]. The input image size is halved after each layer while the number of filters are doubled. The size of latent space is set to 32. The input layer takes a colored image with dimension 128×128 in width and height and we applied ReLu activation functions on each layer.

3.3 AutoEncoder-based Affinity Measure

We use the trained AutoEncoder to estimate the similarity of two detections x_i and x_j of a video sequence based on the Euclidean distance in the latent space:

$$d_{i,j} = \|f(x_i) - f(x_j)\| \tag{10}$$

Figure 3 shows the nearest neighbor of a selected frame t (left box marked in red) from the sequence MOT17-09 and frame $t+5 \cdot k$. The example illustrates that the location of detections with the same ID are close to one another in the latent space even over a long distance of up to 40 frames. Yet, false positives can appear. The example also shows that change in appearance affects the AutoEncoder

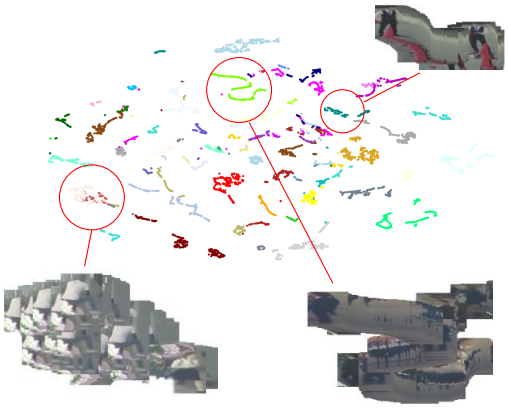

Fig. 4: TSNE Visualization of the latent space of the trained AutoEncoder for the sequence MOT17-04 FRCNN. The colors represent the assigned person IDs. As the appearance changes for example due to pose changes, the latent representations vary smoothly.

distance, further denoted d_{AE}. For instance in the first row, frame 1 and frame 6 are very similar due to the same detection position of the person within the bounding box as well as the direction the girl is looking to. At frame 41, the girl (in Fig. 3) slightly turned towards another person. Although the correct nearest neighbour was retrieved, the distance d_{AE} almost doubled (in blue: distance 4.83 compared to 2.96 at frame 6). Another observation is that the position of the bounding box influences the latent space distance. Such behavior easily allows for false positive associations. In the second row, in the first detection from the left (frame 5), the detection of the person is slightly shifted to the left. At frame 15, 20 or 25, the position is slightly zoomed and d_{AE} increases. Yet, it is overall more stable and less false positive associations are made.

Visualization of Latent Space. Figure 4 shows the TSNE-Visualization [69] of the latent space from the sequence MOT17-04-FRCNN. Our proposed AutoEncoder learned the visual features without supervision. The different colors represent the cluster labels. As shown in the example circled on the bottom left, similar looking persons are very closed to one another in the latent space: The sitting person in white shirt and the lady, wearing a white shirt (example in bottom left). The visualization also shows that the same person may change the appearance over time (example on the bottom right). In the latent space, the *snake*-like shape may indicate that the viewpoint or pose of a person may have changed over time, causing a continuous appearance change. When standing still, the change is minimal, which is also observed in the example on the top right corner. While for nearby frames, we can compute pairwise cues based on the distance between latent feature representation (d_{AE}), as well as on spatial cues (IoU_{DM}), spatial information can not be used to associate detections over

longer temporal distances. However, to facilitate the re-identification of objects after being fully or partly occluded, such long-range information is needed. In these cases, we have to purely rely on the learned latent space distance d_{AE}. The distance is directly cast to a binary logistic regression to compute the cut probability of the respective edge in graph G. The label that is used for the regression comes from the DeepMatching IoU. If $\mathrm{IoU}_{\mathrm{DM}}(x_i, x_j) < T_{low}$ for a threshold T_{low}, x_i and x_j most certainly belong to different objects. If $\mathrm{IoU}_{\mathrm{DM}}(x_i, x_j) > T_{high}$ for a different threshold T_{high}, they are very likely to match. Formally, we estimate a probability $p_e \in [0, 1]$ between two detections using a feature vector $f^{(e)}$ by regressing the parameters β of a logistic function:

$$p_e = \frac{1}{1 + \exp(-\langle \beta, f^{(e)} \rangle)} \tag{11}$$

Thus, the costs c_e can intuitively be computed by the logit. To robustly estimate these probabilities, we set T_{low} and T_{high} most conservatively to 0.1 and 0.7, respectively. From this partial, purely spatially induced labeling, we can estimate cut probabilities for all available features combinations, i.e. possible combinations of $\mathrm{IoU}_{\mathrm{DM}}$ and d_{AE} within nearby frames and only d_{AE} for distant frames.

4 Experiments and Results

We evaluate the proposed method on the MOT17 Benchmark [6] for multiple person tracking. The dataset consists of 14 sequences, divided into train and test sets with 7 sequences each. For all sequences, three different detection sets are provided, from the detectors SDP [70], DPM [71] and FRCNN [72], thus yielding 21 sequences in both data splits. While SDP and FRCNN provide reliable detections, the performance of the DPM detector is relatively noise and many detections are show poor localization.

The settings between the training and testing scenes are very similar such as moving/static camera, place of recording or view angle, such that learning-based methods usually train on the most similar training sequence for every test sequence. For the evaluation, we use the standard CLEAR MOTA metric [73]. We reported Tracking Accuracy (MOTA), Precision (MOTP), number of identity switches (IDs), mostly tracked trajectories ratio (MT) and mostly lost trajectories (ML).

Implementation Details. Our implementation is based on the Tensorflow Deep Learning Framework. We use a convolutional AutoEncoder in order to extract features by optimizing the equation (8). Thus no pre-training or any other ground truth is required. Furthermore, our pre-processing step is only limited to extracting the provided detections from all sequences and resizing them to the corresponding size of the AutoEncoder input layer. Thus the detections from the MOT17 dataset are directly fed to the AutoEncoder. For each sequence from the dataset (MOT17-01 to MOT17-14 with the detector SDP, FRCNN and

Table 2: Tracking Performance using different features on the MOT17 Training Dataset. The third column refers to the frame distance over which bounding boxes are connected in the graph. d_{AE} represents the AutoEncoder latent space distance while $d_{\mathrm{AE+C}}$ includes the clustering term, respectively. Our proposed approach includes lifted edges [4] between frames of distance 10, 20 and 30.

No	Features	Distance	MOTA	MOTP	IDs	MT	ML	FP	FN
1	$\mathrm{IoU}_{\mathrm{DM}}$	1-3	47.2	83.8	3,062	311	657	7,868	167,068
2	d_{AE}	1-3	35.2	83.9	4,378	138	743	10,213	203,868
3	$d_{\mathrm{AE+C}}$	1-3	37.6	84.0	3,830	162	745	8,951	197,308
4	Combined (1+2)	1-3	49.4	83.5	1,730	381	593	7,536	161,057
5	Combined (1+3)	1-3	49.4	83.4	1,713	380	594	7,786	161,084
6	$\mathrm{IoU}_{\mathrm{DM}}$	1-5	47.2	83.5	2,731	337	642	12,195	163,055
7	d_{AE}	1-3	35.8	84.2	4,623	129	755	6,867	204,697
8	$d_{\mathrm{AE+C}}$	1-5	35.2	83.9	4,378	138	743	10,213	203,868
9	Combined (6+7)	1-5	49.7	83.3	1,567	389	578	9,067	158,788
10	Combined (6+8)	1-5	49.8	83.3	1,569	388	580	8,869	158,715
11	**Proposed**	**1-5**	**50.2**	**83.3**	**1,458**	**391**	**582**	**8,466**	**157,936**

Table 3: Tracking result compared to other methods on the MOT17 dataset. The best performance is marked in bold.

Sequence	Method	**MOTA**	MOTP	IDs	MT	ML	FP	FN
Lif_T [35]	Supervised	**60.5**	78.3	1,189	27.0	33.6	**14,966**	**206,619**
MPNTrack [36]	Supervised	58.8	**78.6**	**1,185**	**28.8**	**33.5**	17,413	213,594
eHAF17 [74]	Supervised	51.8	77.0	1,834	23.4	37.9	33,212	**236,772**
AFN17 [75]	Supervised	51.5	77.6	2,593	20.6	35.5	22,391	248,420
jCC [37]	Supervised	51.2	75.9	1,802	20.9	37.0	25,937	247,822
Proposed	Self-Supervised	48.1	76.7	2,328	17.7	39.8	17,480	272,602

DPM), one individual model is trained with the same setup and training parameters. However, it is important to note that the number of detections for each individual person varies significantly: while some pedestrians are staying in the scene for a long time, others are passing by quickly out of the scene. This results different cluster sizes. To balance this, randomized batches of detections are applied during the training, where each batch contains only images from one single frame. This way, one iteration of training contains only detections from unique persons. The initial learning rate is set to $\alpha = 0.001$ and decays exponentially by a factor of 10 over time. The balancing parameter between reconstruction and clustering loss is set to $\lambda = 0$ at the beginning in order to first learn the visual features of the video sequences. After five epochs, the cluster information is included in the training, e.g. λ is set to 0.95 to encode the appearance variations from the spatio-temporal clusters into the latent space of the AutoEncoder.

From Clusters to Tracklets. To transform detection clusters into actual tracks, we follow the procedure proposed in [3], i.e. from all detections within one cluster, we select the one with the best detection score pre frame. Clusters containing less than 5 detections are completely removed and gaps in the resulting tracklets are filled using bilinear interpolation.

4.1 Ablation Study

We investigated feature setups in the minimum cost multicut framework. The cut probability between pairs of nodes are computed using a logistic regression function. Adding new features directly affects the edge cost between pairs thus resulting in different clustering performances. Here, we investigate the extent to which our proposed appearance model improves the tracking performance.

Comparison of different setups. Table 2 shows the evaluated setups and the resulting tracking performance scores. The column *Features* lists the added features to the logistic regression model. The temporal distances over which bounding boxes are connected in the graph are marked in the column *Distances*. The tracking accuracy of experiment 1 and 6, which uses IoU_{DM} only, is 47.2%. Experiment 2+3 and 7+8 compare the different AutoEncoder models: the Euclidean distance (d_{AE}) from the AutoEncoder latent space is computed in order to estimate the similarity of each pair detections. Here, d_{AE} denotes the latent space distance before adding the clustering loss while $d_{\text{AE+C}}$ denotes the distance after training of the AutoEncoder with the clustering loss, i.e. our proposed appearance method.

Best performance with proposed method. The benefit from using the clustering loss on the model training is obvious: for both distances (1-3 and 1-5 frames), the performance is significantly higher. For distance 1-3, $d_{\text{AE+C}}$ has a tracking accuracy of 37.6 compared to d_{AE} (35.2) and for distance 1-5, the MOTA scores are 35.2 and 35.8 for $d_{\text{AE+C}}$ and d_{AE}, respectively. Although the scores are lower than using IoU_{DM}, combining them both together increases the performance further. This is shown in experiment 4+5 and 9+10, where the best score is achieved with in experiment 10 (proposed method). We also observe that the number of identity switches (IDs) is reduced with our setup. Finally, we add lifted long range edges and solve the resulting minimum cost lifted multicut problems on G. Our best performance is achieved using the setup of experiment 11 with a MOTA of 50.2% using all model components.

4.2 Results

Tracking Performance on test data. Here, we present and discuss our final tracking results on the MOT17 test dataset. Compared to the performance on the training dataset, the MOTA score of our proposed approach is slightly lower (Training: 50.2% vs. Testing: 48.1%), which is within the observed variance between different sequences, neglecting excessive parameter tuning. The best

Table 4: Tracking results on the recent MOT20 dataset. Our proposed method is closed to the current state-of-the-art method given the fact that ours is based on self-supervised learning.

Sequence	Method	**MOTA**	MOTP	IDs	MT	ML	FP	FN
SORT20 [76]	Supervised	**42.7**	78.5	**4,470**	**16.7**	**26.2**	**27,521**	**264,694**
Proposed	Self-Supervised	41.8	**78.6**	5,918	15.9	27.0	28,341	266,672

performance is achieved in conjunction with the SDP-detector while the performance on the noisier DPM detections are weaker (detailed tables are provided in the supplementary material). While supervised approaches can also train their models w.r.t. the overlap of provided detections with the ground truth and thus compensate for poor detector quality, our self-supervised approach depends on reasonable object detections.

Comparison with other tracking approaches. We compare our method with five other reported tracking methods Lif_T [35], MPNTrack [36], eHAF17 [74], AFN17 [75] and jCC [37]. We consider a tracking method as supervised when ground truth data is used (for example label data for learning a regression function) or if any pre-trained model is included in the approach. Table 3 gives an overview of the scores in different metrics that is being evaluated. The best on each category is marked in bold.

When comparing more closely the average MOTA scores we achieve per detector over all sequences, our proposed method reaches 46.9% on the SDP detector while [44] reach 47.1%. For a state-of-the-art detector, our method performs thus competitive with supervised one. Yet, on the noisy DPM detections, our approach is outperformed by 10% (49.0 [44] vs. 34.3 (Ours)), decreasing the total average significantly.

Evaluation on MOT20. We evaluated our approach on the recent MOT20 dataset. The current state-of-the-art method [76] achieves a MOTA score of 42.7% while ours 41.8% (see table 4).

5 Conclusion

We present a two stage approach towards tracking of multiple persons without the supervision by human annotations. First, we group the data based on their spatial-temporal features to obtain weak clusters (tracklets). Combining the visual features learned from an AutoEncoder with these tracklets, we are able to automatically create robust appearance cues enabling multiple person tracking over a long distance. The result of our proposed method achieves a tracking accuracy of 48.1% and 41.8% on the MOT17 and MOT20 benchmark, respectively.

Acknowledgement Margret Keuper and Amirhossein Kardoost receive funding from the German Research Foundation (KE 2264/1-1).

References

1. Zamir, A.R., Dehghan, A., Shah, M.: Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In: Computer Vision–ECCV 2012. Springer (2012) 343–356
2. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Improvements to frank-wolfe optimization for multi-detector multi-object tracking. arXiv preprint arXiv:1705.08314 (2017)
3. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Multi-person tracking by multicut and deep matching. In: European Conference on Computer Vision, Springer (2016) 100–111
4. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person reidentification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3539–3548
5. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., Kim, T.K.: Multiple object tracking: A literature review. arXiv preprint arXiv:1409.7618 (2014)
6. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs] (2016) arXiv: 1603.00831.
7. Yoon, Y.c., Boragule, A., Song, Y.m., Yoon, K., Jeon, M.: Online multi-object tracking with historical appearance matching and scene adaptive detection filtering. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE (2018) 1–6
8. Feng, W., Hu, Z., Wu, W., Yan, J., Ouyang, W.: Multi-object tracking with multiple cues and switcher-aware classification. arXiv preprint arXiv:1901.06129 (2019)
9. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2019) 1920–1929
10. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: CVPR. (2017)
11. Mahendran, A., Thewlis, J., Vedaldi, A.: Cross pixel optical-flow similarity for self-supervised learning. In: Asian Conference on Computer Vision, Springer (2018) 99–116
12. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. In: Advances in Neural Information Processing Systems. (2019) 15637–15648
13. Ye, Q., Zhang, T., Ke, W., Qiu, Q., Chen, J., Sapiro, G., Zhang, B.: Self-learning scene-specific pedestrian detectors using a progressive latent model. (2017) 2057–2066
14. Lee, W., Na, J., Kim, G.: Multi-task self-supervised object detection via recycling of bounding box annotations. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
15. Vondrick, C.M., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. (2018)
16. Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: Siamese cnn for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2016) 33–40
17. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. (2011)

18. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: CVPR. (2012)
19. Huang, C., Wu, B., Nevatia, R.: Robust object tracking by hierarchical association of detection responses. In: ECCV. (2008)
20. Andriluka, M., Roth, S., Schiele, B.: Monocular 3d pose estimation and tracking by detection. In: CVPR. (2010)
21. Fragkiadaki, K., Zhang, W., Zhang, G., Shi, J.: Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In: ECCV. (2012)
22. Zamir, A.R., Dehghan, A., Shah, M.: GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In: ECCV. (2012)
23. Henschel, R., Leal-Taixe, L., Rosenhahn, B.: Efficient multiple people tracking using minimum cost arborescences. In: GCPR. (2014)
24. Tang, S., Andriluka, M., Schiele, B.: Detection and tracking of occluded people. IJCV (2014)
25. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Improvements to frank-wolfe optimization for multi-detector multi-object tracking. CoRR **abs/1705.08314** (2017)
26. Shitrit, H.B., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: ICCV. (2011)
27. Wang, X., Turetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: ECCV. (2014)
28. Kumar, R., Charpiat, G., Thonnat, M.: Multiple object tracking by efficient graph partitioning. In Cremers, D., Reid, I., Saito, H., Yang, M.H., eds.: Computer Vision – ACCV 2014, Cham, Springer International Publishing (2015) 445–460
29. Tesfaye, Y.T., Zemene, E., Pelillo, M., Prati, A.: Multi-object tracking using dominant sets. IET Computer Vision **10** (2016) 289–297
30. Wojek, C., Roth, S., Schindler, K., Schiele, B.: Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In: ECCV. (2010)
31. Wojek, C., Walk, S., Roth, S., Schindler, K., Schiele, B.: Monocular visual scene understanding: Understanding multi-object traffic scenes. IEEE TPAMI (2013)
32. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. (2011)
33. Wang, X., Turetken, E., Fleuret, F., Fua, P.: Tracking interacting objects optimally using integer programming. In: ECCV. (2014)
34. Chari, V., Lacoste-Julien, S., Laptev, I., Sivic, J.: On pairwise costs for network flow multi-object tracking. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 5537–5545
35. Hornakova, A., Henschel, R., Rosenhahn, B., Swoboda, P.: Lifted disjoint paths with application in multiple object tracking. arXiv preprint arXiv:2006.14550 (2020)
36. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 6247–6257
37. Keuper, M., Tang, S., Andres, B., Brox, T., Schiele, B.: Motion segmentation & multiple object tracking by correlation co-clustering. IEEE transactions on pattern analysis and machine intelligence (2018)
38. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: Computer Vision and Pattern Recognition Workshops (CVPRW). (2018)

39. Henschel, R., Zou, Y., Rosenhahn, B.: Multiple people tracking using body and joint detections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0
40. Keuper, M., Levinkov, E., Bonneel, N., Lavoué, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1751–1759
41. Keuper, M., Tang, S., Zhongjie, Y., Andres, B., Brox, T., Schiele, B.: A multi-cut formulation for joint segmentation and tracking of multiple objects. arXiv preprint arXiv:1607.06317 (2016)
42. Kumar, R., Charpiat, G., Thonnat, M.: Multiple object tracking by efficient graph partitioning. In: Asian Conference on Computer Vision, Springer (2014) 445–460
43. Ma, C., Yang, C., Yang, F., Zhuang, Y., Zhang, Z., Jia, H., Xie, X.: Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. arXiv preprint arXiv:1804.04555 (2018)
44. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. arXiv preprint arXiv:1903.05625 (2019)
45. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4696–4704
46. Sheng, H., Chen, J., Zhang, Y., Ke, W., Xiong, Z., Yu, J.: Iterative multiple hypothesis tracking with tracklet-level association. IEEE Transactions on Circuits and Systems for Video Technology (2018) 1–1
47. Chen, J., Sheng, H., Zhang, Y., Xiong, Z.: Enhancing detection model for multiple hypothesis tracking. In: Conf. on Computer Vision and Pattern Recognition Workshops. (2017) 2143–2152
48. Li, M., Zhu, X., Gong, S.: Unsupervised person re-identification by deep learning tracklet association. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 737–753
49. Lv, J., Chen, W., Li, Q., Yang, C.: Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7948–7956
50. Karthik, S., Prabhu, A., Gandhi, V.: Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609 (2020)
51. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. arXiv preprint arXiv:1902.06162 (2019)
52. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2701–2710
53. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1422–1430
54. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2536–2544
55. Lee, W., Na, J., Kim, G.: Multi-task self-supervised object detection via recycling of bounding box annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4984–4993
56. Ye, Q., Zhang, T., Ke, W., Qiu, Q., Chen, J., Sapiro, G., Zhang, B.: Self-learning scene-specific pedestrian detectors using a progressive latent model. In: Proceed-

ings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 509–518
57. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. Theoretical Computer Science **361** (2006) 172–187
58. Keuper, M., Levinkov, E., Bonneel, N., Lavoue, G., Brox, T., Andres, B.: Efficient decomposition of image and mesh graphs by lifted multicuts. In: ICCV. (2015)
59. Chopra, S., Rao, M.: The partition problem. Mathematical Programming **59** (1993) 87–115
60. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning **56** (2004) 89–113
61. Horňáková, A., Lange, J.H., Andres, B.: Analysis and optimization of graph decompositions by lifted multicuts. In: ICML. (2017)
62. Andres, B., Kröger, T., Briggman, K.L., Denk, W., Korogod, N., Knott, G., Köthe, U., Hamprecht, F.A.: Globally optimal closed-surface segmentation for connectomics. In: ECCV. (2012)
63. Beier, T., Kroeger, T., Kappes, J., Kothe, U., Hamprecht, F.: Cut, glue, & cut: A fast, approximate solver for multicut partitioning. In: CVPR. (2014)
64. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schieke, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In: European Conference on Computer Vision (ECCV). (2016)
65. Kardoost, A., Keuper, M.: Solving minimum cost lifted multicut problems by node agglomeration. In: ACCV 2018, 14th Asian Conference on Computer Vision, Perth, Australia (2018)
66. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Deep convolutional matching. CoRR **abs/1506.07656** (2015)
67. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. arXiv preprint arXiv:1610.04794 (2016)
68. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. (2010) 249–256
69. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9** (2008) 2579–2605
70. Yang, F., Choi, W., Lin, Y.: Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2129–2137
71. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32** (2010) 1627–1645
72. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. (2015) 91–99
73. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 (2015)
74. Sheng, H., Zhang, Y., Chen, J., Xiong, Z., Zhang, J.: Heterogeneous association graph fusion for target association in multiple object tracking. IEEE Transactions on Circuits and Systems for Video Technology **29** (2018) 3269–3280
75. Shen, H., Huang, L., Huang, C., Xu, W.: Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. arXiv preprint arXiv:1808.01562 (2018)

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