

# SPAMming Labels: Efficient Annotations for the Trackers of Tomorrow

Orcun Cetintas<sup>1,2†</sup>, Tim Meinhardt<sup>2</sup>, Guillem Brasó<sup>1,2†</sup>, and Laura Leal-Taixé<sup>2</sup>

<sup>1</sup> Technical University of Munich

<sup>2</sup> NVIDIA

**Abstract.** Increasing the annotation efficiency of trajectory annotations from videos has the potential to enable the next generation of data-hungry tracking algorithms to thrive on large-scale datasets. Despite the importance of this task, there are currently very few works exploring how to efficiently label tracking datasets comprehensively. In this work, we introduce SPAM, a video label engine that provides high-quality labels with minimal human intervention. SPAM is built around two key insights: i) most tracking scenarios can be easily resolved. To take advantage of this, we utilize a pre-trained model to generate high-quality pseudo-labels, reserving human involvement for a smaller subset of more difficult instances; ii) handling the spatiotemporal dependencies of track annotations across time can be elegantly and efficiently formulated through graphs. Therefore, we use a unified graph formulation to address the annotation of both detections and identity association for tracks across time. Based on these insights, SPAM produces high-quality annotations with a fraction of ground truth labeling cost. We demonstrate that trackers trained on SPAM labels achieve comparable performance to those trained on human annotations while requiring only 3 – 20% of the human labeling effort. Hence, SPAM paves the way towards highly efficient labeling of large-scale tracking datasets. We release all models and code.

**Keywords:** Multi-object tracking · Data labeling · Active learning

## 1 Introduction

Detecting and tracking objects in time is at the core of many challenging vision applications. To address these problems, modern multi-object tracking (MOT) approaches [44, 66, 72, 76, 78] rely on increasingly larger amounts of annotated data. While annotating image datasets is already costly, temporal component of videos further increases the annotation difficulty and data scale requirements. This is due to the fact that redundancies between frames cause the information density to scale poorly with the amount of data, making the overall annotation task more challenging and resource-intensive.

---

† Work done during an internship at NVIDIA.

To mitigate these issues, researchers are spending more effort in operating with limited human annotations. Self- and semi-supervised learning (SSL) and pseudo-labeling (PL) approaches are often used to leverage unlabeled data whenever possible [34,35,40], while active learning (AL) strategies aim at selecting the most informative samples to label. All these strategies are commonly applied in various image understanding tasks, such as classification [11,14] or object detection [17,67]. Even if video tasks do not have any lower requirement for data, there are only a handful of works that target efficient labeling of tracking data in the video domain [39,59–61]. Existing works fail to provide *comprehensive* solutions to labeling by either ignoring dense temporal component of videos [39,61] or working under a limited single-object setup [71]. An effective video labeling engine should operate across multiple objects and frames, considering the intricate dependencies between tracks inherent in the labeling problem.

Videos have unique characteristics that can be leveraged to unleash the full potential of a modern label engine. Firstly, frame-wise similarities in videos yield data redundancies. In the context of MOT, this implies that some associations between detections are relatively easy to solve [7]. This setup provides an ideal opportunity to leverage state-of-the-art models pre-trained to pseudo-label new data at zero cost. Secondly, the dependencies of objects across time and space cause annotations in one frame of the video to have cascading effects in other frames. For example, solving an association for one track can resolve associations for neighboring tracks. In order to maximize the efficiency of the annotations, we frame the labeling task within a model that takes into account such dependencies. Thus, we adopt a track-centric labeling approach rather than a frame-centric one [39].

We present the **SPAM** video label engine, which chains **S**ynthetic pre-training, **P**seudo-labeling, and **A**ctive learning with a graph-based **M**odel for optimal tracking performance with minimal human effort. In the image domain, there are plenty of large-scale annotated datasets that a model-driven approach can leverage for pre-training [34]. However, this is not the case for multi-object tracking datasets, which are typically much smaller in scale; therefore, we advocate for a **S**ynthetic pre-training. We further analyze which components of a tracking method are more affected by the synthetic-to-real gap. This first experiment gives us an important insight: labeling should be focused on detections and associations, and we can directly use the synthetically trained reID model. With our synthetic pre-trained model, we then propose a model-driven approach to produce **P**seudo-labels on real data and retrain the model with its own pseudo-labels. We show that with this recipe we can even reach a performance comparable to ground truth without ever training our labeler on manually annotated data.

Synthetic pre-training and pseudo-labeling allow us to handle easy decisions and defer only uncertain or complex decisions to an annotator. We propose an **A**ctive learning strategy where the labeler makes decisions at the track-level instead of frame-level, thereby utilizing the annotation budget more efficiently. Towards this goal, we solve the labeling problem with a **M**odel, based on graph

hierarchies [12]. This model captures long-range spatio-temporal dependencies, which means our annotations will have an impact across frames, and not only on the frame where the annotation actually happens. We extend the model with a detection filtering layer so that we can naturally label detection and association within the same framework.

In summary, we propose **SPAM**, a video label engine that significantly reduces annotation cost by dealing with easy scenarios with a strong MOT model pre-trained on (i) **S**ynthetic data, followed by (ii) **P**seudo-labeling real data for its own re-training. Harder detection and association decisions are carefully selected by an (iii) **A**ctive learning scheme that works on a (iv) **M**odel based on graph hierarchies. This allows us to efficiently use the annotator’s budget and account for spatio-temporal dependencies in the labels. The resulting model generates labels comparable to ground truth performance, requiring only 3 – 20% of manual annotation effort across three diverse datasets.

## 2 Related Work

**Pre-training for tracking.** Over the years, the multi-object tracking (MOT) community has annotated many challenging benchmark datasets [18, 19, 57]. However, due to the significant costs associated with manually labeling temporal data, most benchmarks only provide few annotated video sequences. To obtain top performance, modern data-hungry tracking methods are usually pre-trained on additional video and tracking datasets [20, 22, 68, 74, 77]. Many works also rely on relevant image recognition [56] datasets to boost their per-frame object detection performance. An alternative pre-training direction avoids human annotations entirely by relying on simulated synthetic data [23, 24, 26, 29, 58].

Our label engine also pre-trains the labeling model synthetically, minimizing its total annotation footprint. To this end, we demonstrate the effectiveness of synthetic pre-training [23] for generating pseudo-labels for self-supervision.

**Self-supervision in the temporal domain.** To benefit not only from already annotated datasets, many works explore the utilization of available unlabeled data sources, *i.e.*, how to perform self-supervised learning for videos. Notable examples leverage cycle-consistency-based correspondences [30, 64], color propagation [63] or augmentation-based consistencies [4, 41]. Furthermore, [45] uses contrastive learning over re-identification (reID) embeddings to perform unsupervised association learning. More recently, several works [33–35, 40] in the image domain demonstrate the effectiveness of training on pseudo-labels generated by strong underlying labeling models. In the video domain, this type of self-supervision is mostly unexplored.

We demonstrate the effectiveness of pseudo-labels for the temporal domain by improving the label quality of our SPAM labeler via self-supervision on its own output. The necessary signal-to-noise ratio for this training loop is obtained by pre-training a strong graph-based model on synthetic data.

**Efficient labeling of video datasets.** While significant effort has been devoted towards increasing the labeling efficiency of image datasets [1, 34], similar

approaches for the video and tracking domain are still relatively limited. Earlier works reduce the number of frames to annotate within individual videos but still rely heavily on human annotators [62]. Other solutions are limited to only a single object per sequence [16, 60] or still rely on human bounding box annotations [59] to label segmentation masks across time. The authors of [39] select key frames based on their expected impact on tracking performance. While effective within its scope, this approach only yields sparse frame annotations, which are insufficient for training all modern trackers. To reduce the human annotation effort for trajectories, [43] proposes an efficient video interface that still requires labeling all trajectories.

In contrast, our label engine performs active learning for multiple objects across all video frames and is the first of its kind to tackle long-term tracking problems via a hierarchical labeling approach. The uncertainty measures provided by our graph-based model allow us to focus expensive annotation efforts on individual tracks instead of labeling entire frames in a brute-force manner.

**Multi-object tracking models.** Tracking objects across time in a video is an established problem with a long research history. The most dominant tracking-by-detection (TbD) paradigm splits the task into two steps: (i) detecting objects in every frame and (ii) associating them to object trajectories, also referred to as data association. Modern trackers have explored performing (i) and (ii) jointly [7, 25, 44, 78], or providing solutions for (ii) relying on motion models [2, 3, 38, 47, 49, 52, 53, 69, 75], appearance models [13, 15, 36, 46, 51, 54, 70, 76] or more complex techniques involving transformers [10, 72]. An alternative direction of work relies on graph formulations [6, 31, 37, 48, 73] to utilize the TbD paradigm. In this formulation, nodes model object detections, and edges, association hypotheses. This long-studied formulation has recently been revived with remarkable success with the help of learnable graph neural network (GNN) architectures [8, 9, 12, 65].

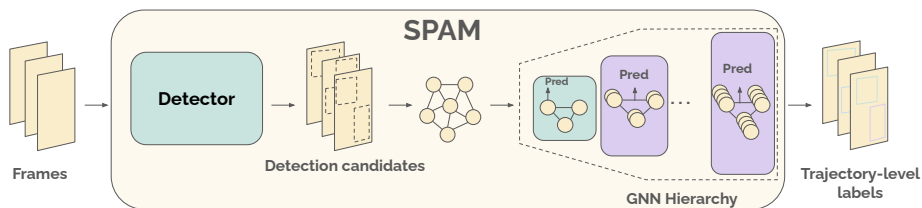
For our label engine to generate long-term pseudo-labels and uncertainty estimations, we rely on a strong graph-based tracker [12]. To incorporate both detection and association labeling within a unified graph framework, we introduce a novel hierarchy level dedicated to detection filtering.

### 3 SPAM

#### 3.1 Overview

Our main goal is to label multi-object tracking datasets *with minimal annotation effort while enabling optimal tracking performance*. Our key observation is that most association cases can be solved easily by models pre-trained on synthetic data. Therefore, no human effort is needed, and we can instead use our model’s predictions as pseudo-labels. Decisions that are more complex or uncertain are selected by an active learning scheme to be annotated manually.

In order to perform pseudo-labeling and active learning, we require a model that can capture long-term spatio-temporal dependencies between tracks. We leverage a hierarchical graph neural network formulation and cast the annotation



**Fig. 1:** Overview of the SPAM model. We first generate a set of detection candidates with our detector. Hierarchical GNNs then classify these candidates into valid and invalid objects via node classification, and assign identities through edge classification.

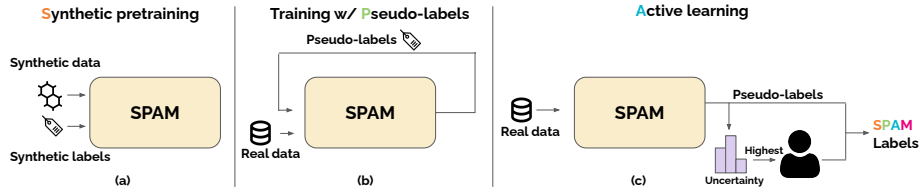
task over a graph. Our architecture (Fig. 1) draws inspiration from SUSHI [12], which demonstrated remarkable association performance. Our model’s scope, however, goes beyond association. Given the constraints of the labeling task, we need to equip our model with the ability to reason about detections before performing association. To address this broader goal, our GNN hierarchy, also includes the capability to accept or reject detections by classifying nodes into correct and incorrect object hypotheses.

Our overall model is pre-trained on synthetic data [23] and further fine-tuned on its own predicted pseudo-labels over the target dataset that we aim to annotate. This recipe allows for strong labelling performance *without any human annotations*, and provides the basis for a scalable video label engine. After this training loop, we use our updated model’s predictions as pseudo-labels to annotate a large portion of the target dataset. Decisions that are more complex or uncertain are selected by an active learning scheme to be annotated manually. The resulting set of labels achieves near-GT performance with significantly fewer manual annotations, thereby vastly enhancing the efficiency of the overall process. We provide an overview of our training and annotating recipe in Fig. 2.

### 3.2 Graph Hierarchies as a Model for Labeling

**Problem formulation.** Our model builds upon the standard graph formulation [73] where, given a set of object candidates  $\mathcal{O}$ , our goal is to obtain a set of valid objects  $\mathcal{O}_v \subset \mathcal{O}$  and their corresponding trajectories  $\mathcal{T}$ . Each trajectory  $T_k \in \mathcal{T}$  consists of a set of objects sharing the same identity or, equivalently, the set of edges connecting those nodes. Specifically, we model each object candidate  $o_i \in \mathcal{O}$  as a node and each association hypothesis among candidates as an edge in an undirected graph  $G = (V, E)$ . Within this formulation, nodes  $u \in V$  could be classified into valid objects if  $u \in \mathcal{O}_v$ . Analogously, edges can be classified into correct or incorrect hypotheses depending on whether they belong to some trajectory or not. This formulation can be generalized to a hierarchical setting by splitting video sequences into non-overlapping subsequences, *i.e.*, disjoint subgraphs, and progressively merging them [12].

**Background on graph hierarchies for MOT.** Our model consists of a hierarchy of GNNs operating over the graph formulation described above. Intuitively,

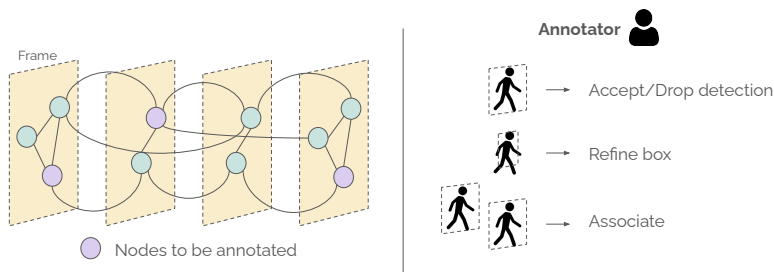


**Fig. 2:** Overview of the SPAM training and annotation pipeline. (a) Initial model training on synthetic data. (b) Application of SPAM to generate pseudo-labels without incurring manual annotation costs on a real dataset, followed by self-training on pseudo-labels. (c) Real dataset labeling using pseudo-labels and an uncertainty-based active learning approach.

starting from detection candidates, at each hierarchy level, GNN hierarchy subsequently merge tracklets from the previous level into longer ones. Nodes and edges are represented by initial embeddings and message passing is used to share information within the graph in order to update node and edge embeddings with richer information. The GNNs are trained to classify edges into active and inactive association hypotheses and they share the same features, network architecture and learnable weights across hierarchy levels. In addition to the edge features reported in [12], for each object candidate  $o_i$  we concatenate its timestamp  $t_i$ , normalized bounding box coordinates and dimensions  $(x_i, y_i, h_i, w_i)$  and corresponding confidence from the detector  $c_i$  and embed it as node features.

**Classifying nodes.** We aim to minimize the annotation effort; therefore, we ask the annotator to operate on a fixed graph of detection candidates, *i.e.* annotator does not label new detections after the initial set given by detector. This is why we aim to create an initial graph that is over-complete. In order to do so, we use a low confidence threshold for our detector and obtain a large number of detection candidates. While this ensures a high detection recall, it also introduces a larger number of false positive detections that our model needs to address. As we show in experiments, the original SUSHI architecture [12] cannot handle such false positives, which is why we propose a new layer in the hierarchy,  $GNN_{node}$ . Intuitively,  $GNN_{node}$  leverages temporal information to validate object candidates. We use its output node embeddings and feed them to a lightweight classifier to enable false positive detection filtering. By doing so, we can consider a large number of input proposals achieving high detection recall by leveraging the spatio-temporal reasoning ability of our GNN for detection candidates.

**Pre-training and annotation-free fine-tuning.** We pre-train all of our model components on the large-scale synthetic dataset MOTSynth [23]. To boost our model’s performance and bridge the domain gap with our target dataset for annotation, we conduct further fine-tuning on the target dataset after pre-training. We achieve this by utilizing the raw predictions of our model on the target dataset and employing them as pseudo-labels for *self-training*. As we show in



**Fig. 3:** Our graph-based labeling pipeline begins with the selection of nodes for annotation. For each node to be annotated, the annotator could be asked to validate the *detection*, improve the *localization* by refining the box or perform *association*.

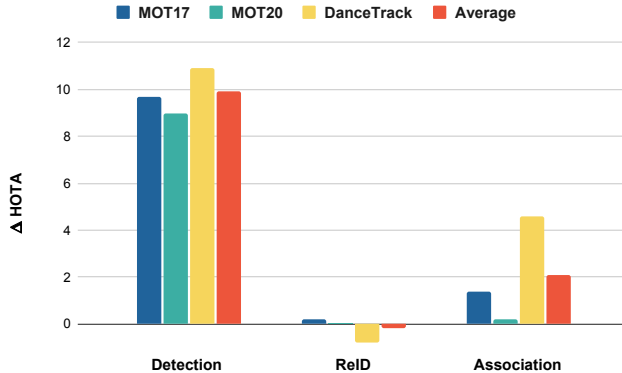
our experimental section, this strategy yields a significant performance improvement without incurring any human annotation cost.

### 3.3 How to Label on Graph Hierarchies?

**Manual labeling.** Labeling video data at the box level involves several steps: identifying objects of interest, localizing their boundaries with bounding boxes, and assigning them unique identities. Typically, annotators achieve this by examining each frame, detecting and localizing objects individually, and then assigning identities based on temporal context from neighboring frames. In terms of annotation effort in clicks, *detecting* and *localizing* objects typically require two clicks (one for the top-left corner and one for the bottom-right corner of the bounding box), while *associating* objects across time requires one click (identifying the next occurrence of the object in subsequent frames), as reported in [18, 19]. This manual process is labor-intensive, costly, and severely limits the availability of annotated video data.

**Graph-based labeling.** Utilizing a graph formulation for detection and tracking enables explicit modeling of objects in the scene and their interactions. In this framework, each node of the graph represents an object, allowing a manual annotation of an object to be framed as selecting a node on this graph. Given a set of nodes to be annotated, possible annotator decisions are: (i) accepting/rejecting a *detection* as we are working on a large set of detection candidates (one click), (ii) refining the bounding-box to improve *localization* (two clicks, one for the top-left corner and one for the bottom-right corner of the bounding box) and (iii) *associating* a node across time (one click) as shown in Fig. 3.

**Annotation acquisition.** The core idea of our approach is to use our model decisions to guide the annotation process. For the vast majority of scenarios, we directly use our model’s predictions as pseudo-labels. For a small subset of more challenging decisions where our model exhibits uncertainty, we validate our decisions with an annotator as described above. Within our graphs,



**Fig. 4:** Analysis of performance gap between training a model on synthetic and real data for the three most common tracking components: detection, association, re-identification.

we adopt an uncertainty-based approach to identify which nodes require annotations. Specifically, for each node  $v \in V$ , we compute its uncertainty as  $\text{uncert}(v) := \max_{u \in N_v} H(\hat{y}_{(v,u)})$ , where  $N_v$  denotes the set of neighboring nodes of  $v$  and  $\hat{y}_{(v,u)}$  is the edge prediction for edge  $(v, u)$ :

$$H(\hat{y}_{(v,u)}) := -(\hat{y}_{(v,u)} \log \hat{y}_{(v,u)} + (1 - \hat{y}_{(v,u)}) \log(1 - \hat{y}_{(v,u)})) \quad (1)$$

At the  $GNN_{node}$  level, where we have node predictions instead of edge predictions, we simply compute  $\text{uncert}(v)$  as  $H(\hat{y}_{(v)})$  instead. Subsequently, we refer the nodes with the highest uncertainty scores to the annotator for manual annotation and use the remaining predictions as pseudo-labels for the rest of the nodes.

**Hierarchical label generation.** To fully leverage our hierarchical graph formulation, we apply our label acquisition approach described above in a hierarchical manner. Intuitively, we distribute our annotation budget  $B$  across  $L$  hierarchical levels as  $B_1, \dots, B_L$ , with  $B_1 + \dots + B_L = B$ . In deeper hierarchy levels, nodes represent tracklets instead of single detections. This allows us to propagate annotator decisions for the entire cluster instead of individual detections, resulting in solving multiple uncertainties with a single annotation. As a result, we utilize the annotation budget more effectively while also mitigating error propagation within our pseudo-labels.

## 4 Experiments

### 4.1 Datasets and Metrics

**Datasets.** We conduct experiments on three public benchmarks:



- **MOT17** [18] is an established benchmark containing 14 challenging video sequences, exhibiting diverse characteristics, including variations in camera motion, viewpoint, and pedestrian densities.
- **MOT20** [19] contains a total of 8 video sequences, all featuring extremely crowded scenes. Due to their high pedestrian density, the emphasis is on addressing highly challenging scenarios characterized by frequent and prolonged occlusions.
- **DanceTrack** [57] contains 100 video sequences. These sequences feature dancing individuals with similar appearances, facilitating the evaluation of tracker robustness against highly challenging motion patterns where appearance information has a relatively minor influence. This non-standard setup often leads to a performance drop in trackers [57].

**Metrics.** We assess the performance of trackers using three established metrics:

- **MOTA** [32] assesses trajectory coverage, by combining object detection measurements with identity switches, with the latter playing a relatively minor role.
- **IDF1** [50] focusses on identity preservation over the sequences and provides a measure of association performance.
- **HOTA** [42] is a recently introduced metric designed to strike a balance between association and object detection performance within a single score.

## 4.2 Implementation Details

**Architecture and training.** Our model consists of three main components: (i) a detector network, based on YOLOX [27], (ii) a ResNet-based ReID network [28], and (iii) a GNN hierarchy, all pretrained on the synthetic sequences of MOTSynth [23]. After pretraining, we use the resulting model to obtain pseudo-labels for our target MOT dataset, and then fine-tune our detector and GNN on them. Note that our **Model** is *never trained on human annotations*, it is only trained on synthetic annotations and its own pseudo-labels.

**Experimental setup.** Online trackers are real-time tools suitable for deployment on robots or autonomous vehicles, whereas offline trackers are better suited for labeling tasks. Therefore, we structure our experiments to utilize **SPAM** for labeling and use these labels with online trackers. To evaluate the quality of **SPAM** labels, we train two state-of-the-art online trackers with these labels: ByteTrack [75] and GHOST [54]. For validation, we follow the standard half-split setting [54, 75, 76] over MOT17 sequences. Annotation budgets are computed based on number of clicks as described in Sec. 3.3.

## 4.3 Synthetic Pretaining

Several works have shown that synthetic pre-training is a strong starting point for multi-object tracking methods [23, 75]. In this section, we aim to analyze the gap between training a model on synthetic data and training it with real

Model	High Conf. Box	Low Conf. Box	$GNN_{node}$	MOTA $\uparrow$	IDF1 $\uparrow$	HOTA $\uparrow$	DetA $\uparrow$
A	✓	✗	✗	64.4	74.7	59.9	55.9
B	✓	✓	✗	60.6	71.4	58.5	54.9
C	✓	✓	✓	<b>65.4</b>	<b>75.1</b>	<b>60.4</b>	<b>56.6</b>

**Table 1:** Ablation study investigating the impact of incorporating low confidence bounding boxes with and without the  $GNN_{node}$ .

data for the three most common tracking components: detection, association, re-identification. For each setting, only the specific component that we are analysing is trained on synthetic data while other components are all trained on real data. We perform this experiment on three datasets and plot the relative HOTA difference (a positive score indicates superior performance with real data training) when training on real data vs. synthetic data in Fig. 4. Our first observation, is that the detector is the component that suffers the domain gap the most, with a performance drop of 9.9 HOTA points. Re-identification, on the other hand, has a similar performance when trained on synthetic or real data. Lastly, association performance is 2.1 HOTA higher with real data training. This first experiment gives us an important insight: labeling and retraining should be focused on detection and association components, and we can directly use the synthetically trained reID model.

#### 4.4 Towards a Model for Labeling

**Graph-based validation of detection candidates.** In our framework, annotators have the capability to eliminate detections, but we refrain from requesting them to annotate new detections in order to maintain minimal annotation effort. Instead we start from an initial graph formed by an over-complete set of detections. Towards this end, we use a low confidence threshold for our detector, thereby reducing the number of false negatives in our initial graph and ensuring a high recall. As a result, this leads to an increase in the number of false positives incorporated into the graph, a tendency that is further accentuated by the fact that our detector is exclusively trained on synthetic data. We propose to leverage the graph structure itself to validate the detections based on temporal consistency. Specifically, we add an additional GNN-layer in our hierarchy,  $GNN_{node}$ , which intuitively examines the spatial and temporal consistency of the bounding box candidates and filters out implausible ones. In Tab. 1, we assess the effect of this design choice. Model A uses only high confidence bounding boxes, model B creates a graph with also low confidence bounding boxes but without  $GNN_{node}$ , while C includes  $GNN_{node}$ . As we can see, simply adding low confidence boxes results in a drastic drop in performance on all metrics. Only by adding our  $GNN_{node}$  we are able to take full advantage of the low confidence boxes without penalizing our performance with too many errors.

**SPAM is a SOTA tracker.** Before moving to labeling with our model, we want to verify its overall performance and generalization capabilities. Only by using a state-of-the-art model we will be able to reduce annotation effort as much as

Method	HOTA $\uparrow$	MOTA $\uparrow$	IDF1 $\uparrow$	HOTA $\uparrow$	MOTA $\uparrow$	IDF1 $\uparrow$	HOTA $\uparrow$	MOTA $\uparrow$	IDF1 $\uparrow$
	MOT17			MOT20			DanceTrack		
MeMOT [10]	56.9	72.5	69.0	54.1	66.1	63.7	–	–	–
TrackFormer [44]	57.3	74.1	68.0	54.7	65.7	68.6	–	–	–
MOTR [72]	57.8	68.6	73.4	–	–	–	54.2	51.5	79.7
FairMOT [76]	59.3	73.7	72.3	54.6	61.8	67.3	39.7	82.2	40.8
QDTrack [46]	63.5	78.7	77.5	60.0	74.7	73.8	54.2	87.7	50.4
ByteTrack <sup>†</sup> [75]	62.8	78.9	77.1	60.4	74.2	74.5	47.7	<b>89.6</b>	53.9
GHOST [54]	62.8	78.7	77.1	61.2	73.7	75.2	56.7	91.3	57.7
UTM [70]	64.0	<b>81.8</b>	78.7	62.5	<b>78.2</b>	76.9	–	–	–
MotionTrack [49]	65.1	81.1	80.1	62.8	78.0	76.5	–	–	–
SUSHI [12]	66.5	81.1	83.1	64.3	74.3	79.8	63.3	88.7	<b>63.4</b>
<b>SPAM (Ours)</b>	<b>67.5</b>	80.7	<b>84.6</b>	<b>65.8</b>	76.5	<b>81.9</b>	<b>64.0</b>	89.2	<b>63.4</b>

**Table 2:** Test set results on MOT17, MOT20 and DanceTrack when using **SPAM** solely as a tracker.

possible. We therefore pre-train our model on the synthetic dataset, and finetune it on the specific real dataset using groundtruth. For a fair comparison, we obtain object detections from a YOLOX detector [27] trained following [12, 54, 75]. Our model outperforms state-of-the-art by 1.0 HOTA point on MOT17, 1.5 HOTA points on MOT20, and 0.7 HOTA points on DanceTrack as shown in Tab. 2. Furthermore, on MOT17 and MOT20, there is also a significant improvement in IDF1, showing the identity-preserving capabilities of the model, which is especially important for our video annotation setting. Even if being state-of-the-art is not the main goal of this paper, having a strong performing model allows us to have an excellent starting point for producing pseudo-labels. We would like to emphasize that, we use ground truth annotations only for this experiment just to compare our model fairly with other trackers. From this point onward, we will not use ground truth labels. We will use our synthetic pretraining weights and focus on the pseudo-labeling and active learning aspects of our method to show how to annotate multi-object tracking datasets.

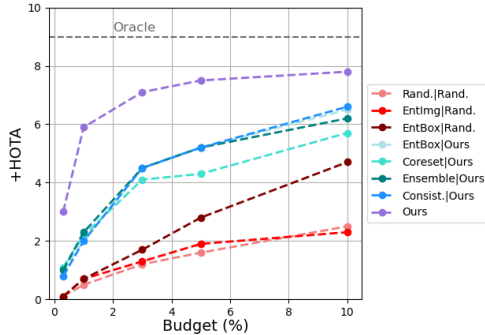
#### 4.5 Enhancing Annotations with Pseudo-labels and Active Learning

**Retraining our Model on Pseudo-labels.** In this experiment, we compare two models. The first one, **SPAM**, is only trained on synthetic data. The second one, **SPAM**, is trained in three stages: (i) pre-training on synthetic data, (ii) pseudo-labeling the real dataset with our model, and (iii) re-training our model on pseudo-labels. In Sec. 4.5, we observe a significant improvement when using pseudo-labels to re-train our model: an increase of 4-6 HOTA, 4-6 MOTA, and 4-7 IDF1 points on MOT17, MOT20, and DanceTrack, respectively. Note that these improvements come at zero human annotation effort as this process requires no manual annotations.

**Active learning on graph hierarchies.** We now test the effectiveness of our proposed active learning labeling on graph hierarchies, and show a comprehensive comparison with several baselines in Fig. 5. We directly measure the accuracy of the output labels given an increasing percentage of manual annotations selected

Model	Pseudo-label	HOTA $\uparrow$	MOTA $\uparrow$	IDF1 $\uparrow$
MOT17				
SPAM	$\times$	60.0	65.3	75.1
SPAM	$\checkmark$	<b>63.8</b>	<b>69.2</b>	<b>79.7</b>
MOT20				
SPAM	$\times$	52.2	65.4	70.3
SPAM	$\checkmark$	<b>58.7</b>	<b>70.2</b>	<b>77.1</b>
DanceTrack				
SPAM	$\times$	41.8	65.2	43.4
SPAM	$\checkmark$	<b>48.1</b>	<b>71.0</b>	<b>50.2</b>

**Table 3:** Performance boost obtained by our model when retraining with its own pseudo-labels incurring no manual annotation cost.



**Fig. 5:** Impact of various active learning strategies on the quality of labeling.

by different acquisition functions on MOT17. The vertical axis shows the increase in HOTA over the baseline performance, *i.e.*, the performance without active learning. *Oracle* is our upper bound and shows the performance if all possible labels are set to their correct value by the annotator. Every method has two different labeling schemes: one for detections (nodes), and one for associations (edges) as explained in Sec. 3.3. In addition to our method, we explore active learning strategies commonly used in object detection, including Coreset [55], Ensemble [5], and Consistency over augmented inputs [21]. The naming is based on the active learning strategy for detection in the first part and the strategy for association in the second part. For instance, *EntImg/Rand.* indicates image level entropy-based selection for detection and random selection for association.

The methods shown in red shades and circle markers compare three detection labeling methods while keeping association label selection random. **Box-level labeling** is far superior to commonly-used **image-level labeling**, or **random labeling**. This highlights the potential of performing active learning at the object level rather than the frame level. The methods in blue shades and triangle markers all use our proposed active learning selection on graph hierarchies for the association, and compare several detection selection schemes. The **Coreset** [55], which prioritizes diversity of the data over uncertainty, emerges as the poorest performing approach. This observation underlines that, when accurate predictions are available, an uncertainty-based approach proves to be a more effective complement to pseudo-labels. **Box-level entropy**, **ensembles**, and **consistency** all perform similarly. Finally, **our method**, with both detection and association label selection on graph hierarchies is superior to all compared baselines, staying at only 1.2 points from Oracle performance. Notably, we also unify the AL strategy for detection and association with our approach.

#### 4.6 Putting Everything Together: SPAM Labels in Action

**Training trackers with our labels.** In this experiment, we ablate the effects of our **SPAM** pipeline on the actual usecase of training a model with our pro-

Model	Labels	Pseudo-labels	Active Learn.	Ann. Effort ↓	HOTA ↑	MOTA ↑	IDF1 ↑
ByteTrack [75]	Ground Truth	–	–	100%	52.6	60.4	65.7
ByteTrack [75]	SPAM	✗	✗	✗	49.6	57.8	63.1
ByteTrack [75]	SPAM	✓	✗	✗	50.5	59.5	64.5
ByteTrack [75]	SPAM	✓	✓	3.3%	52.5	61.8	66.2

**Table 4:** Effect of (i) Synthetic training, (ii) Retraining our model with our Pseudo-labels, and (iii) Active learning-based acquisition and annotation of a small percentage of real data, on our label quality on the MOT17 validation set.

vided labels. We ablate label quality under the following settings: (i) Synthetic training, (ii) Retraining our models on our Pseudo-labels, (iii) Active learning based acquisition and annotation of a small percentage of real data. We train state-of-the-art online tracker ByteTrack [75] from scratch with the aforementioned sets of labels and report its performance on the MOT17 validation set in Tab. 4. The goal performance we want to reach is the one achieved by the model trained on groundtruth annotations (row 1). Row 2 shows our model trained only on synthetic data, which is 3 HOTA points below our goal. Using pseudo-label based training of our model increases performance by almost a HOTA point, and the ground truth level performance is achieved by also using our active learning method to select which samples to label. By annotating only 3.3% of the data, we can achieve the same results as the model trained on the ground truth data with 100% annotation effort. This shows the potential of our SPAM in reducing annotation effort for multi-object tracking datasets.

**How much data do we really need to label?** In this final experiment, our goal is to assess the impact of our entire pipeline by answering the question: How much data do we actually need to label in order to achieve the same performance as a model trained on the entire groundtruth training dataset? For that, we choose two well-established trackers, ByteTrack [75] and GHOST [54], and train them from scratch using our SPAM labels. We report their performance on MOT17, MOT20, and DanceTrack test sets in Tab. 5. We first confirm that our results on the validation set also generalize to the test set, and that we can reach GT performance with 3.3% of annotation effort on MOT17. Note that for MOT17 our annotations yield a slight improvement over GT, which can be attributed to observed annotation noise in the original GT, particularly in moving sequences (further details can be found in the supplementary material). MOT20 and DanceTrack are known to be more challenging datasets [57], which is confirmed by the fact that the trackers still miss a few points when trained on only 3.3% of manually annotated data. Given that we cannot do a full analysis of different percentages because we are working on the test sets, we choose two percentages of manually annotated data on which to train our trackers: 10% for MOT20, and 20% for DanceTrack. In both cases, we achieve on-par performance with the trackers trained on full groundtruth data. This clearly shows the potential in properly leveraging synthetic data, pseudo-labels from a strong model, and active learning on graph hierarchies in order to label multi-object tracking

Method	Labels	Ann. Effort ↓	HOTA ↑	MOTA ↑	IDF1 ↑
MOT17 Test					
ByteTrack [75]	GT	100%	50.6	60.4	61.1
ByteTrack [75]	SPAM	3.3%	<b>51.6</b>	<b>64.0</b>	<b>63.0</b>
GHOST [54]	GT	100%	49.5	58.0	59.0
GHOST [54]	SPAM	3.3%	<b>51.3</b>	<b>61.9</b>	<b>62.1</b>
MOT20 Test					
ByteTrack [75]	GT	100%	<b>48.7</b>	60.5	60.0
ByteTrack [75]	SPAM	3.3%	47.9	57.6	61.4
ByteTrack [75]	SPAM	10%	48.4	<b>61.0</b>	<b>62.8</b>
GHOST [54]	GT	100%	<b>48.3</b>	<b>59.7</b>	60.2
GHOST [54]	SPAM	3.3%	47.0	58.2	60.7
GHOST [54]	SPAM	10%	46.9	59.0	<b>60.8</b>
DanceTrack Test					
ByteTrack [75]	GT	100%	41.0	79.1	46.0
ByteTrack [75]	SPAM	3.3%	39.5	76.4	45.0
ByteTrack [75]	SPAM	20%	<b>41.3</b>	<b>80.9</b>	<b>48.1</b>
GHOST [54]	GT	100%	<b>45.8</b>	80.6	46.3
GHOST [54]	SPAM	3.3%	41.0	76.3	44.8
GHOST [54]	SPAM	20%	43.9	<b>81.3</b>	<b>47.7</b>

**Table 5:** Evaluation of SPAM labels on MOT17, MOT20, and DanceTrack test sets with varying annotation budgets by training ByteTrack [75] and GHOST [54] using annotations generated by our method.

data, where we can save 96.7% of the annotation effort on MOT17, 90% of the annotation effort on MOT20 and 80% of the annotation effort on DanceTrack.

## 5 Conclusion

We have introduced SPAM, a labeling engine to efficiently obtain trajectory annotations from videos. We have shown that SPAM is able to produce annotations that yield state-of-the-art tracking performance with a significantly reduced human annotation effort. Through our extensive experiments we have shown that (i) our synthetic pretraining + pseudo-label finetuning recipe yields a strong, annotation-free, labeler; (ii) our combination of active learning + pseudo-labeling for annotation generation yields high quality and efficient annotations; and (iii) that graphs are an excellent tool to provide comprehensive modelling for trajectory annotation. We believe that SPAM paves the way to make large-scale cheap annotation of tracking datasets a reality, which will in turn power the next generation of data-hungry tracking algorithms.

## References

1. Acuna, D., Ling, H., Kar, A., Fidler, S.: Efficient interactive annotation of segmentation datasets with polygon-rnn++. In: CVPR (2018) [3](#)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016) [4](#)
3. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In: CVPR. pp. 1265–1272 (2011). <https://doi.org/10.1109/CVPR.2011.5995311> [4](#)
4. Bastani, F., He, S., Madden, S.: Self-supervised multi-object tracking with cross-input consistency. *Advances in Neural Information Processing Systems* **34**, 13695–13706 (2021) [3](#)
5. Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. In: CVPR. pp. 9368–9377 (2018) [12](#)
6. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. *IEEE TPAMI* **33**(9), 1806–1819 (2011) [4](#)
7. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV. pp. 941–951 (2019) [2](#), [4](#)
8. Brasó, G., Cetintas, O., Leal-Taixé, L.: Multi-object tracking and segmentation via neural message passing. *IJCV* **130**(12), 3035–3053 (2022) [4](#)
9. Braso, G., Leal-Taixe, L.: Learning a neural solver for multiple object tracking. In: CVPR (2020) [4](#)
10. Cai, J., Xu, M., Li, W., Xiong, Y., Xia, W., Tu, Z., Soatto, S.: Memot: Multi-object tracking with memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8090–8100 (2022) [4](#), [11](#)
11. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS* (2020) [2](#)
12. Cetintas, O., Brasó, G., Leal-Taixé, L.: Unifying short and long-term tracking with graph hierarchies. In: CVPR. pp. 22877–22887 (June 2023) [3](#), [4](#), [5](#), [6](#), [11](#)
13. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE international conference on multimedia and expo (ICME). pp. 1–6. IEEE (2018) [4](#)
14. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: *IEEE Int. Conf. Mach. Learn.* pp. 1597–1607. PMLR (2020) [2](#)
15. Chu, P., Ling, H.: Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: ICCV (October 2019) [4](#)
16. Dai, K., Zhao, J., Wang, L., Wang, D., Li, J., Lu, H., Qian, X., Yang, X.: Video annotation for visual tracking via selection and refinement. In: ICCV (2021) [4](#)
17. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1601–1610 (June 2021) [2](#)
18. Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., Leal-Taixé, L.: Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision* **129**(4), 845–881 (2021) [3](#), [7](#), [9](#)
19. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J.Q., Cremers, D., Reid, I.D., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. *ArXiv abs/2003.09003* (2020) [3](#), [7](#), [9](#)

20. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR. pp. 304–311. IEEE (2009) [3](#)
21. Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixe, L., Alvarez, J.M.: Not all labels are equal: Rationalizing the labeling costs for training object detection. In: CVPR. pp. 14492–14501 (2022) [12](#)
22. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR. pp. 1–8 (2008). <https://doi.org/10.1109/CVPR.2008.4587581> [3](#)
23. Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., Cucchiara, R.: Motsynth: How can synthetic data help pedestrian detection and tracking? In: ICCV. pp. 10849–10859 (October 2021) [3](#), [5](#), [6](#), [9](#)
24. Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., Cucchiara, R.: Learning to detect and track visible and occluded body joints in a virtual world. In: ECCV (2018) [3](#)
25. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: ICCV (Oct 2017) [4](#)
26. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: CVPR. pp. 4340–4349 (2016) [3](#)
27. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021) [9](#), [11](#)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) [9](#)
29. Hu, H.N., Cai, Q.Z., Wang, D., Lin, J., Sun, M., Krähenbühl, P., Darrell, T., Yu, F.: Joint monocular 3d vehicle detection and tracking. In: ICCV (2019) [3](#)
30. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems* **33**, 19545–19560 (2020) [3](#)
31. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: CVPR. pp. 1–8 (2007). <https://doi.org/10.1109/CVPR.2007.383180> [4](#)
32. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation for face, text and vehicle detection and tracking in video: data, metrics, and protocol. *IEEE TPAMI* (2009) [9](#)
33. Kaul, P., Xie, W., Zisserman, A.: Label, verify, correct: A simple few-shot object detection method. In: IEEE Conference on Computer Vision and Pattern Recognition (2022) [3](#)
34. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV (2023) [2](#), [3](#)
35. Lan, S., Yang, X., Yu, Z., Wu, Z., Alvarez, J.M., Anandkumar, A.: Vision transformers are good mask auto-labelers. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2023) [2](#), [3](#)
36. Leal-Taixe, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: Siamese cnn for robust target association. In: CVPRW (June 2016) [4](#)
37. Leal-Taixe, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: CVPR (June 2014) [4](#)



38. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: *Int. Conf. Comput. Vis. Worksh.* pp. 120–127 (2011). <https://doi.org/10.1109/ICCVW.2011.6130233> 4
39. Li, R., Zhang, B., Liu, J., Liu, W., Zhao, J., Teng, Z.: Heterogeneous diversity driven active learning for multi-object tracking. In: *ICCV* (2023) 2, 4
40. Litrico, M., Del Bue, A., Morerio, P.: Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023) 2, 3
41. Liu, K., Jin, S., Fu, Z., Chen, Z., Jiang, R., Ye, J.: Uncertainty-aware unsupervised multi-object tracking. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 9996–10005 (2023) 3
42. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. *IJCV* 129(2), 548–578 (2021) 9
43. Manen, S., Gygli, M., Dai, D., Van Gool, L.: Pathtrack: Fast trajectory annotation with path supervision. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 290–299 (2017) 4
44. Meinhardt, T., Kirillov, A., Leal-Taixé, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022) 1, 4, 11
45. Meng, S., Shao, D., Guo, J., Gao, S.: Tracking without label: Unsupervised multiple object tracking via contrastive similarity learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 16264–16273 (2023) 3
46. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: *CVPR*. pp. 164–173 (2021) 4, 11
47. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You’ll never walk alone: Modeling social behavior for multi-target tracking. In: *ICCV*. pp. 261–268 (2009). <https://doi.org/10.1109/ICCV.2009.5459260> 4
48. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: *CVPR*. pp. 1201–1208 (2011). <https://doi.org/10.1109/CVPR.2011.5995604> 4
49. Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., Tang, W.: Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 17939–17948 (2023) 4, 11
50. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: *Eur. Conf. Comput. Vis. Worksh.* pp. 17–35. Springer (2016) 9
51. Ristani, E., Tomasi, C.: Features for multi-target multi-camera tracking and re-identification. In: *CVPR* (June 2018) 4
52. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: *ECCV* (2016) 4
53. Scovanner, P., Tappen, M.F.: Learning pedestrian dynamics from the real world. In: *2009 IEEE 12th International Conference on Computer Vision*. pp. 381–388. IEEE (2009) 4
54. Seidenschwarz, J., Brasó, G., Serrano, V.C., Elezi, I., Leal-Taixé, L.: Simple cues lead to a strong multi-object tracker. In: *CVPR*. pp. 13813–13823 (2023) 4, 9, 11, 13, 14

55. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: ICLR (2018), <https://openreview.net/forum?id=H1aIuk-RW> **12**
56. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018) **3**
57. Sun, P., Cao, J., Jiang, Y., Yuan, Z., Bai, S., Kitani, K., Luo, P.: Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In: CVPR (2022) **3, 9, 13**
58. Sun, S., Akhtar, N., Song, X., Song, H., Mian, A., Shah, M.: Simultaneous detection and tracking with motion modelling for multiple object tracking. In: ECCV. pp. 626–643 (2020) **3**
59. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7942–7951 (2019) **2, 4**
60. Voigtlaender, P., Luo, L., Yuan, C., Jiang, Y., Leibe, B.: Reducing the annotation effort for video object segmentation datasets. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3060–3069 (2021) **2, 4**
61. Vondrick, C., Ramanan, D.: Video annotation and tracking with active learning. *NeurIPS* **24** (2011) **2**
62. Vondrick, C., Ramanan, D., Patterson, D.: Efficiently scaling up video annotation with crowdsourced marketplaces. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11. pp. 610–623. Springer (2010) **4**
63. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. In: Proceedings of the European conference on computer vision (ECCV). pp. 391–408 (2018) **3**
64. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: CVPR (2019) **3**
65. Wang, Y., Kitani, K., Weng, X.: Joint object detection and multi-object tracking with graph neural networks. In: IEEE Int. Conf. Robotics and Autom. pp. 13708–13715 (2021). <https://doi.org/10.1109/ICRA48506.2021.9561110> **4**
66. Wang, Z., Zheng, L., Liu, Y., Wang, S.: Towards real-time multi-object tracking. *The European Conference on Computer Vision (ECCV)* (2020) **1**
67. Wei, F., Gao, Y., Wu, Z., Hu, H., Lin, S.: Aligning pretraining for detection via object-level contrastive learning. In: *Advances in Neural Information Processing Systems* (2021) **2**
68. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X.: Joint detection and identification feature learning for person search. In: CVPR. pp. 3415–3424 (2017) **3**
69. Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: CVPR. pp. 1345–1352 (2011). <https://doi.org/10.1109/CVPR.2011.5995468> **4**
70. You, S., Yao, H., Bao, B.K., Xu, C.: Utm: A unified multiple object tracking model with identity-aware feature enhancement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21876–21886 (2023) **4, 11**
71. Yuan, D., Chang, X., Liu, Q., Yang, Y., Wang, D., Shu, M., He, Z., Shi, G.: Active learning for deep visual tracking. *IEEE Transactions on Neural Networks and Learning Systems* (2023) **2**

72. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision. pp. 659–675. Springer (2022) [1](#), [4](#), [11](#)
73. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR (2008) [4](#), [5](#)
74. Zhang, S., Benenson, R., Schiele, B.: Citypersons: A diverse dataset for pedestrian detection. In: CVPR. pp. 3213–3221 (2017) [3](#)
75. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: ECCV. pp. 1–21. Springer (2022) [4](#), [9](#), [11](#), [13](#), [14](#)
76. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. IJCV **129**(11), 3069–3087 (2021) [1](#), [4](#), [9](#), [11](#)
77. Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., Tian, Q.: Person re-identification in the wild. In: CVPR. pp. 1367–1376 (2017) [3](#)
78. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: ECCV. pp. 474–490. Springer (2020) [1](#), [4](#)