SPAMming Labels: Efficient Annotations for the Trackers of Tomorrow *Supplementary Material*

Orcun Cetintas[1,2†], Tim Meinhardt[2], Guillem Brasó[1,2†], and Laura Leal-Taixé[2]

[1] Technical University of Munich
[2] NVIDIA

Abstract. This supplementary material offers additional details (Appendix A) and experimental results (Appendix B) of SPAM. To this end, we provide further information on the architecture of our graph-based model (Appendix A.1) and its application for hierarchical labeling (Appendix A.2). In Appendix A.3, we complete the implementation details of the main paper. The experimental results in Appendix B.1 demonstrate the superiority of our approach to a common frame-based labeling method. We compare labeling performance for static and moving cameras in Appendix B.2 and finally discuss about an alternative evaluation setting in Appendix B.3.

A Additional Details about SPAM

A.1 GNN Model

Our hierarchical graph-neural network (GNN) utilizes the time-aware message passing framework introduced in [1]. We use a hierarchy of 10 GNNs operating over a video clip of 512 frames. At the first level, we employ a GNN_{node}, which considers a temporal window of 10 frames for each node. Subsequent levels, denoted as GNN_{edge}, cover 2, 4, ..., up to 512 frames. We follow a simple and unified approach for all hierarchy levels: GNN_{node} and GNN_{edge} levels share the same input features and network architecture (as presented in Sec 3.3 of the main paper). Moreover, GNN_{edge} levels also share learnable weights across hierarchy levels. Overall, we rely on a lightweight GNN hierarchy with only approximately 65K parameters running at 20 frames per second (FPS).

A.2 Active Hierarchical Labeling

To fully utilize our hierarchical graph framework, we distribute our annotation budget B across L hierarchical levels as $B_1, \ldots, B_L$, with $B_1 + \cdots + B_L = B$. Prioritizing deeper levels enables a more effective allocation of the annotation budget. Intuitively, in deeper hierarchy levels, nodes represent tracklets rather than individual detections. We propagate annotator decisions for entire clusters

rather than individual detections, resolving multiple uncertainties with a single annotation. This makes the annotation pipeline more efficient compared to working at the detection level. For example for MOT17 [6], 50% of the budget B is allocated to the last three levels, 30% to the previous three levels, and the remaining 20% to earlier levels. For MOT17, we do not allocate a budget for refining bounding boxes since our experiments revealed that the bounding boxes generated by our synthetic detector are already well localized. However, for the challenging DanceTrack [8] dataset, we allocate a portion (approximately 30%) of our annotation budget B for refining bounding boxes.

A.3 Implementation Details

Architecture and training. We train our YOLOX [3] detector on MOTSynth for 170 epochs following the training parameters in [9]. We keep augmentations on throughout the training. We train our GNN hierarchy on MOTSynth jointly for 10 epochs, with a learning rate of $3 * 10^{-4}$ and weight decay of 10^{-4} with batches of 4 graphs. We use a focal loss with $\gamma = 1$ and the Adam optimizer [4].
Experimental setup. After obtaining labels with SPAM, we train two well-established online trackers ByteTrack [9] and GHOST [7] to evaluate our label quality in the main paper. Our main goal is to compare our label quality with ground truth label quality, therefore, we start trainings from scratch and only train on the dataset of interest (200 epochs for MOT17, 150 epochs for MOT20 and 100 epochs for DanceTrack). During testing, we follow their default parameters [7, 9].

B Additional Experiments

B.1 Frame-based Labeling and FPS

The most related work to active learning in the MOT domain is [5], a frame-based labeling method. Motivated by the redundancy of nearby video frames, they

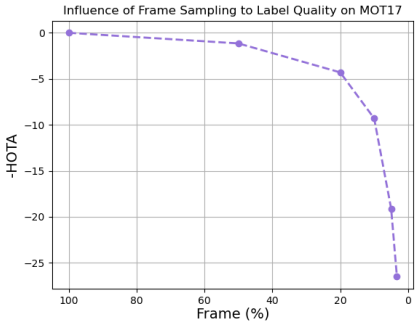

Fig. 1: Annotation quality (reported as relative HOTA drop) on MOT17 by uniformly labeling different percentages of frames, and then interpolating the ground truth boxes.

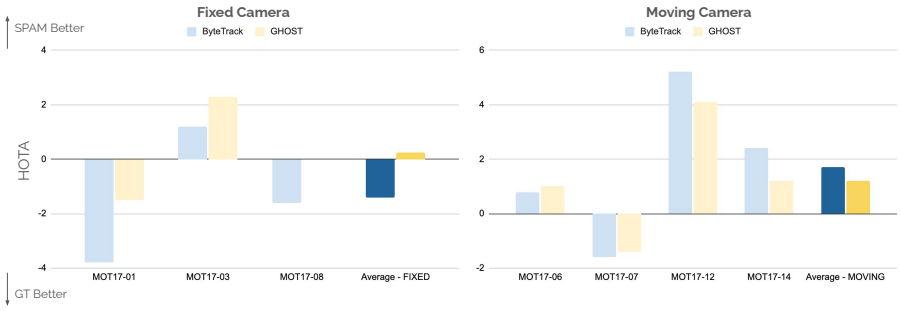

Fig. 2: Comparison of MOT17 results for static and moving camera sequences.

propose to label only a portion of frames. However, their method requires a large labeling budget (50% of the frames) to achieve ground-truth level performance. Furthermore, the authors [5] report that at lower data regimes (20% to 40%) their method is outperformed by a simple uniform sampling of frames.

To demonstrate the inefficiency of such a frame-based labeling method, we show that for high frame rates, labeling such a large portion of the frames will trivially lead to ground-truth performance. This is attributed to the redundancy of data in MOT17 sequences, which are annotated with high frames per second (FPS) of up to 30. To this end, we examine the annotation quality on MOT17 by uniformly labeling different percentages of frames, *i.e.*, reducing the FPS, and then interpolating the ground truth boxes. We report relative HOTA drops compared to labeling all frames in Fig. 1. Notably, we observe that a naive solution of labeling 50% of the frames (every other frame) and interpolating the bounding boxes yields almost perfect labels (1 HOTA drop, leading to 99 HOTA). However, a significant performance drop is observed for lower data regimes, i.e., using lower than 20% of frames equivalent to approximately 6 FPS for MOT17. This highlights that labeling becomes notably more challenging for low data regimes in which [5] fails to operate.

In contrast to frame-based labeling, SPAM performs active learning for multiple objects across all video frames, utilizing the uncertainty measures from our graph-based model. This allows us to invest expensive annotation efforts on *individual tracks* instead of labeling *entire frames* in a brute-force manner. Consequently, we can reach ground-truth level performance with a minimal annotation budget of only 3.3% for MOT17.

B.2 Label Noise for Moving Cameras

In our experiments detailed in Sec 4.6 of the main paper, we found that on MOT17, our annotations with a budget of 3.3% yield slightly better performance compared to the ground truth. We attribute this observation to annotation noise in the original ground truth, particularly in sequences with moving cameras. To illustrate this, we provide detailed results on MOT17 for each sequence with

ByteTrack and GHOST in Fig. 2. We compare relative HOTA scores between trainings on ground truth and SPAM labels. Positive scores indicate superior performance with SPAM labels, while negative scores indicate that ground truth labels result in better performance. Overall, with a 3.3% budget, ground truth exhibits better performance on static camera sequences, while SPAM outperforms ground truth labels on moving camera sequences. We speculate that this phenomenon could be due to the additional noise in ground truth labels on moving sequences, which may be introduced by the interpolation-based labeling strategy reported in [2].

B.3 Alternative Evaluation Setting

As demonstrated in the literature on pseudo-labeling, active learning, and noisy labels, labels do not contribute equally when training models. Therefore, our main goal is not to achieve perfect labels but to label data with minimal annotation effort while enabling optimal tracking performance. Thus, we train SOTA trackers with our labels and compare the results with ground truth training in our experiments similar to an AL evaluation setting. For completeness, we also report tracking scores reached by our labels directly as an alternative setting here. We obtain 84.1 MOTA, 72.2 HOTA, and 90.1 IDF1 with our labels (3.3% annotation effort) on MOT17. This shows that the labels are indeed not completely covering groundtruth. However, results from the Table 5 of the main paper show that training on those labels gives the same results as training on full groundtruth. Thus, even though the labels are not covering the full ground-truth, allocating more annotation resources will not improve the training performance due to diminishing returns. This demonstrates that perfect labels are not necessary to reach optimal tracking performance and our experimental setting is well-suited for our goal.

References

1. Braso, G., Leal-Taixe, L.: Learning a neural solver for multiple object tracking. In: CVPR (2020) 1
2. 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) 4
3. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021) 2
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 2
5. 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, 3
6. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016) 2
7. 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) 2
8. 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) 2
9. 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) 2