SELSAMOT: Sequence-Level Semantic Aggregation for Multi-Object Tracking

Elaine Sui* Serena Yeung
Stanford University, Stanford, CA 94305
{esui, syyeung}@stanford.edu

Abstract

Multi-object tracking (MOT) has become an important research topic in computer vision in recent years due to the rise of robotics and autonomous vehicles and a need for rich scene understanding from RGB video streams. However, tracking faces many challenges including tracking through poor quality frames, large pose variation, occlusion and re-identification of instances. As frames in the same video are temporally dependent on each other, it is a natural extension to use contextual information from videos to generate better object detections to then improve tracking performance. Although existing methods perform very well on benchmark MOT datasets, these methods only use frame-level dependencies in the tracking phase of the algorithm. In this work, we propose SELSAMOT, a method that aggregates region proposal features across frames, a technique explored in the adjacent task of video object detection (VID). Specifically, we add multi-head attention modules to aggregate features from the current and context frames in our object detector before passing the detection results to a state-of-the-art tracking algorithm. We test our proposed model on MOT17, a benchmark tracking dataset for pedestrians and obtain comparable performance to our Faster R-CNN baseline on both object detection and tracking metrics, which we attribute to dataset complexity. The source code is released at https://github.com/elaine-sui/selsamot.

1. Introduction

Multi-object tracking (MOT) is the task of estimating the locations and identities of instances in a video clip. From autonomous robots tracking objects and agents in the real world [16] to assessing a surgeon's skill level from the movement of hands and tools [9], MOT increases our scene-level understanding, which is crucial in many downstream applications. However, MOT is not a trivial task. Rather, trackers must be able to follow instances through obstacles such as occlusion, movement in and out of the frame, and motion blur, just to name a few. Thus, it would make sense to leverage the temporal dependency of frames in a video clip throughout all parts of a tracking algorithm to best mitigate these aforementioned issues.

Current tracking paradigms of tracking-by-detection and tracking-by-attention (i.e. transformer-based detection-by-tracking) are limited in that they do not explicitly enforce temporal consistency in their intermediate (e.g. object detections) predictions. Tracking-by-detection algorithms typically use image object detectors that assume frames to be independent from one another. Similarly, tracking-by-attention methods feed video frames to the encoder sequentially where the detection of "newborn" objects is independent to each frame.

In this paper, we draw inspiration from feature-based video object detectors to propose SELSAMOT, a temporally consistent tracking algorithm. SELSAMOT can be divided into a video object detector and deterministic tracker. The video object detector inspired by [20] takes as input a video frame f and k randomly-sampled frames from the same video clip to output the object detections for f. The object detection results (bounding box predictions and confidence) are then passed to the tracking algorithm to predict the tracks (IDs associated with each detected instance).

We choose to follow the tracking-by-detection paradigm so that we can leverage the well-performing video object detectors that are robust to motion blur and large pose variation, as well as state-of-the-art tracking algorithms which address occlusion and re-identification.

2. Related Work

2.1. Multi-Object Tracking

In general, there are two paradigms used in multi-object tracking (MOT): 1) tracking-by-detection and 2) tracking-by-attention.

Tracking-by-detection Leveraging recent advancements in object detection, tracking-by-detection methods use a deterministic tracking algorithm that matches an object detector's predicted bounding boxes to tracklet identities. These methods are primarily based on using a Kalman filter to predict the location of bounding boxes in the next frame, com-

puting the similarity between the frame's detection bounding boxes and the Kalman filtered prediction, and using a matching strategy to assign the detection boxes to unique identities [2, 24]. ByteTrack [24], the current state-of-the-art, improves upon SORT [2], the classical MOT algorithm, by using all predicted bounding boxes, even low-scoring ones that are typically thrown out, as they claim keeping those bounding boxes would help with tracking through occlusion. However, these methods require careful tuning of thresholds to the given video so that the tracking algorithm will perform well.

Tracking-by-attention Rather than using a two-stage detection then tracking pipeline, these transformer-based tracking-by-attention algorithms directly predict tracklets for each video frame. These methods typically use an encoder-decoder architecture where they encode the current frame and decode the set of tracks that are "alive" in that frame. [13, 22]. MOTR [22], a transformer-based multi-object tracker that builds off of DETR [4], introduces a temporal aggregation network and collective average loss to improve the flow of temporal information through the network. However, as these methods perform object detection and tracking simultaneously, these methods suffer from poor detection of newborn tracks compared to their tracking-by-detection counterparts. This problem may, in part, be due to how the creation of "detect queries," which carry information of detected instances in the current frame, does not leverage useful information that can be found in other frames of the video.

2.2. Video Object Detection

Unlike image object detection, the goal of video object detection is to predict temporally consistent bounding boxes for object instances in a video clip. The main challenges that are being tackled are object detection through motion blur and large pose variation. Most proposed methods include either one or a combination of optical flow-based [8, 26] and spatiotemporal feature aggregation-based techniques [5, 20, 25] to utilize information in different frames.

Optical flow-based methods rely heavily on accurately predicting the motion between consecutive frames. However, in the case of very fast motion, the appearance of objects in the frame and hence, the information that can be gleaned by optical flow predictors, degrades. On the other hand, feature aggregation methods do not need to depend on having consecutive frames of good quality. Rather, features can be aggregated from frames more globally. Specifically, [20] proposed adding Sequence Level Semantic Aggregation (SELSA) modules to the ROI head of a Faster R-CNN to perform a weighted aggregation of region proposal features over frames throughout an entire video clip. These modules are essentially multi-head attention (MHA) modules whose queries are the region proposal features of the current frame and whose keys and values are the features of the randomly sampled "context" frames from the same video. The purpose of these modules is to create more informative features by leveraging other frames with the same instances to form a more discriminative feature embedding that is robust to visual changes. In our method, we will use attention modules similar to [20] that perform a weighted aggregation of region proposal features over frames at the full-sequence level. In contrast to the [20] who test their model on ImageNet VID [14], which comprises of short video clips with single (or few) annotated subjects per clip, we attempt to extend this method to the more complex task of MOT where the number of instances per frame is much higher and the instances themselves vary in size.

3. Methods

3.1. Adding SELSA Modules

Inspired by [20], we propose to inject temporal consistency in the object detection phase of the tracking-by-detection algorithm by using multi-headed attention [18] to aggregate features at the region proposal level from multiple frames of the same video. We perform this feature aggregation in the Region of Interest (ROI) head of the Faster R-CNN meta-architecture.

For each frame f in a video, let $\mathbf{X}^f = [\mathbf{x}_1^f, \mathbf{x}_2^f, \ldots, \mathbf{x}_k^f]^\top$ be the matrix where the rows are embeddings of region proposals generated by the Region Proposal Network (RPN) of the Faster R-CNN framework. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_\ell\}$ be a subset of randomly chosen frames from the same video such that $f \notin \mathcal{F}$. Then, for each query $\mathbf{q} = \mathbf{x}_i^f$, and for the set of keys and values $\mathbf{K} = \mathbf{V} = [\mathbf{x}_1^{f_1}, \ldots \mathbf{x}_{k_1}^{f_1}, \mathbf{x}_1^{f_\ell}, \ldots \mathbf{x}_{k_\ell}^{f_\ell}]^\top$,

$$\tilde{\mathbf{x}}_i^f = [head_1, head_2, \ldots, head_h]\mathbf{W}^O$$

where

$$head_m = Attention(\mathbf{W}_i^{Q\top}\mathbf{q}, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V)$$

The new region proposal features $\tilde{\mathbf{X}}^f = [\tilde{\mathbf{x}}_1^f, \ldots, \tilde{\mathbf{x}}_k^f]$ are then transformed and added to the original features $\mathbf{X}^f$ before being passed through a ReLU non-linearity layer. Thus, the overall structure is a modified version of the canonical transformer block shown in Figure 1.

We stack two attention blocks in the Region of Interest (ROI) head before bounding box regression and classification. At training time, we randomly select 2 context frames within a ± 2.5 second window from the current frame. At test time, we randomly select 5 context frames within a ± 2.5 second window from the current frame.

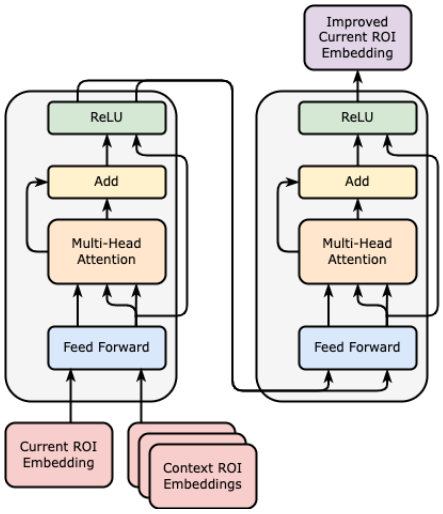

Figure 1. Two stacked SELSA modules.

3.2. SELSA Intuition

As the ROI features are fed through the classification and bounding box regression branches of the R-CNN, these features encode relevant information regarding their respective ROI's appearance and localization. As a result, by aggregating region proposal features across frames, we aggregate semantic similarities that should help discriminate the ROI from others whilst making it more robust to visual differences in appearances coming from artifacts such as motion blur and pose variation. As an oversimplification, one can think of this as forming a feature representation for an instance that is unclear in the current frame by aggregating features of that same object from clearer frames. Mitigating the extent of the effects of visual dissimilarity would then allow for better tracking performance downstream.

However, given that videos capture motion, the locations of region proposals for a particular instance will most probably be different across frames. Thus, as we perform semantic aggregation, we lose information regarding the localization of the instance. To counteract this effect, we add the original region proposal features to its semantically aggregated ones in order to maintain localization information that is important for bounding box regression. Afterwards, we apply a ReLU non-linearity in order to increase the module's expressitivity.

3.3. Our contribution

The Faster R-CNN architecture, as well as training and evaluation pipelines were implementing using detectron2 [21]. The Generalized R-CNN, training and evaluation pipelines were already provided, however, we customized the Faster R-CNN architecture to accommodate for the context frames and passing them through the network, which required modifying the RCNN, RPN, and ROI classes. We implemented, by hand, the SELSA module and its forward pass in the ROI box head. We also created a custom dataloader that randomly samples images from the same video. Data augmentation was also built on top of the pre-existing pipeline in detectron2, and we implemented, again by hand, other augmentations including MixUp [23] and Mosaic [3] (which were not used in the final model). Further, we added a class to report the validation loss and customized the existing COCOEvaluator to run through only a sample of the validation set after at every evaluation step. We also integrated the detectron2 model with ByteTrack [24].

4. Dataset

We use a combination of the MOT17 [6] training and the CrowdHuman [15] training and validation datasets. MOT17 [6] is a benchmark dataset for multi-object tracking of pedestrians (Figure 3. It has a training set consisting of 7 videos of varying frame rates and resolution with 5,316 frames total. The dataset also includes a test set consisting of 7 videos with 5.919 frames total where predictions must be submitted to their leaderboard to get results.

Following [17,24], we split the training set into train_half and val_half, where the first half of each video is in the train_half set (2,664 frames) and the 2nd half in the val_half set (2,652 frames). As we only track pedestrians, we also use the CrowdHuman [15] training and validation datasets for training. The CrowdHuman dataset consists of 20,000 images total of varying resolutions that all contain people in the frame (Figure 4).

For the purposes of this project, we report our results on the MOT17 half validation set.

Note that when we our input image is from CrowdHuman [15], our context images are simply repetitions of the input image.

5. Experimental Details

5.1. Baseline

We use a Faster R-CNN with ResNet101 backbone with a Feature Pyramid Network [11] and 2 fully-connected (FC) layers in the Region of Interest (ROI) head as our baseline to perform object detection. We then use the ByteTrack [24] algorithm for tracking on our detections.

5.2. Model

SELSAMOT is simply a direct extension of the baseline where the 2 FC layers are replaced by the 2 stacked SELSA modules. The proposed model is illustrated in Figure 2. Note that the original implementation of ByteTrack [24] uses a YOLOX [7] backbone and thus, also outputs objectness scores. As this is not part for 2-stage detector outputs, we replace those scores with 1's.

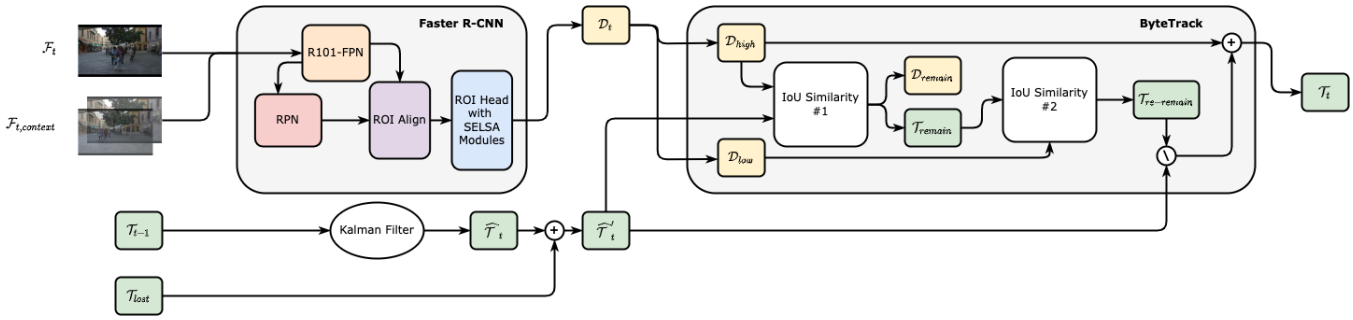

Figure 2. SELSAMOT: $\mathcal{D}_t$ denotes the set of detections for frame $\mathcal{F}_t$. $\mathcal{T}_{t-1}$ denotes the set of tracks carried over from time step $t-1$. $\mathcal{T}_{lost}$ denotes the lost tracks that are still being kept in case of re-identification.

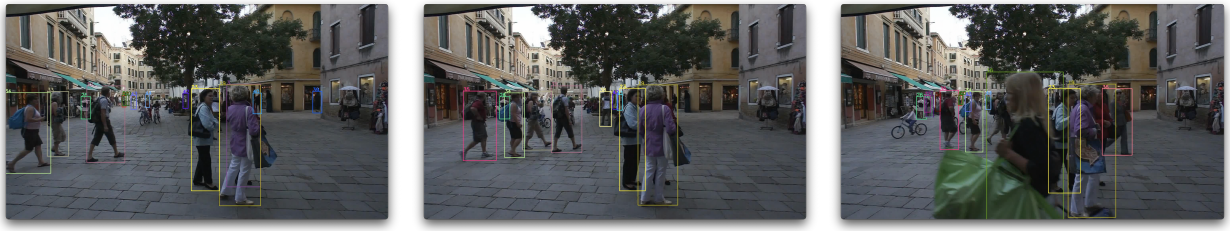

Figure 3. Sequence of frames in MOT17 [6] with ground truth bounding boxes and track IDs

5.3. Implementation Details

For our baseline and proposed object detectors, we started with pre-trained weights on the COCO dataset [12] as published by detectron2 [21]. The models were trained on 1 NVIDIA Tesla V100 GPU with batch size 4 due to memory constraints. We optimize via SGD with weight decay of 1e-4 and momentum of 0.9. Due to training time constraints, we set the initial learning rate to be 1.25e-3 and train for 120k iterations. We decay the learning rate by a factor of 10 at 30k, 45k and 100k iterations.

During training, the shortest side of the input image is randomly re-scaled from 576 to 672 pixels, with a maximum side length of 1000 pixels. During testing, the shortest side is re-scaled to 800 pixels, with a maximum size of 1440 pixels. Basic data augmentation includes random crop, random horizontal flip and random distortion. We then add random blur, saturation, contrast, and rotation. We also implemented MixUp [23] and Mosaic [3], but these proved to be ineffective in combination with our proposed method.

All metrics were evaluated on MOT17 val_half. Note that our models are not comparable to the state-of-the-art due to compute constraints, and thus, were omitted from this paper.

5.4. Metrics

CLEAR MOT The CLEAR MOT [1] metrics are one of the most widely used tracking metrics to date. They comprise of Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP). MOTA encapsulates all errors in object configuration including misses, mismatches (Identity Switches) and false positives whereas MOTP assesses the precision of estimating object locations. We report MOTA, MOTP and IDSW (Identity switches). These metrics are defined as follows:

$$\text{MOTA} = 1 - \frac{|\text{FN}| - |\text{FP}| + |\text{IDS}|}{|\text{gtDet}|} \tag{1}$$

$$\text{MOTP} = \frac{1}{|\text{TP}|} \sum_{c \in \{\text{TP}\}} \mathcal{S}(c) \tag{2}$$

where gtDet refers to the number of ground truth detections and $\mathcal{S}(c)$ refers to the total error in estimated position for all matched detection-track pairs in frame c.

HOTA Recently, [10] claimed that existing MOT metrics, including MOTA as well as ID metrics (e.g. IDF1) do not equally measuring the detection and association of tracks, both of which are essential for proper tracking. Moreover, they claim that these metrics do not capture how well tracks are localized. Thus, they developed Higher Order Tracking Accuracy (HOTA) which measures and balances these three goals. HOTA is comprised of two parts: detection accuracy (DetA) and association accuracy (AssA). DetA is the percentage of aligning detections to tracks and AssA is the average alignment between matched trajectories, averaged over all detections. We report HOTA, DetA and AssA in our results. These metrics are defined as follows:

Figure 4. Sample images from CrowdHuman [15]

$$\text{DetA} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|} \tag{3}$$

$$\text{AssA} = \frac{1}{|\text{TP}|} \sum_{c \in \{\text{TP}\}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|} \tag{4}$$

$$\text{HOTA} = \sqrt{\text{DetA} \cdot \text{AssA}} \tag{5}$$

where the A of TPA, FNA, and FPA refers to association.

6. Results & Discussion

6.1. Object Detection

From Table 1, we see that our baseline and SELSAMOT perform very similarly and thus, the extra contextual information aggregated in the ROI head via the SELSA modules do not seem to be as effective on this more complex dataset than what was reported by [20] on ImageNet VID [14]. Specifically, we hypothesize that this is due to the difference in the distribution of objects across MOT17 [6] versus that of ImageNet VID. MOT17 comprises of a high number of small- and medium-sized instances per frame and a single class of objects whereas ImageNet VID comprises of fewer and larger instances per frame and has detections for 30 classes. As a result, the multi-headed attention modules in SELSAMOT fail to attend to the most relevant region proposals in the context frames, as there are simply too many of them that are also similar-looking to each other. One can argue that despite the difference in data, we should observe significantly higher average precision on large objects. However, even APl is comparable across the baseline and SELSAMOT suggesting that the model's need to accurately detect the many small instances detracts from its ability to detect the "easier" larger instances. However, as all the object detection metrics are relatively comparable despite adding extra information, a lot of which is noise from region proposals of different instances, it is possible that the instances' feature embeddings have been improved with respect to robustness to differences in visual appearance, which can be explored further in future work.

6.2. Tracking

Similarly to object detection, we see comparable performance across most tracking metrics in Table 2. The exceptions are Detection Accuracy (DetA), Multi-Object Tracking Accuracy (MOTA) and Identity Switches (IDSW). The first two place a greater emphasis on detection [10] and hence, directly reflects SELSAMOT's slightly lower detection performance. However, we see that tracking using SELSAMOT detections leads to fewer identity switches. This means that, on average, when matching a predicted track ID to its corresponding ground truth, for the same instance, the predicted track ID remains "matched" to the ground truth ID for longer, and hence, switches occur less frequently. That said, one can also have a lower IDSW if there is a higher false negative (IDFN) count, as a switch would not be counted if an instance was not assigned an ID at all, and/or a high false positive (IDFP) count, as there is no ground truth to "match." Looking more closely, we see that although there are sequences where SELSAMOT detections are less prone to ID switches than the baseline such as in Figure 5, high false negatives indeed account for a portion of the lower IDSW with the baseline having 20,826 total identity false negatives (IDFN) compared to SELSAMOT's 21,286 over the entire MOT17 val_half set (refer to Table 3). Overall, the SELSAMOT's slightly poorer object detection performance is directly reflected in its slightly poorer tracking performance.

6.3. Tracking Per Video

We further look at the per-video tracking metrics in Figure 6. We observe that SELSAMOT performs worse than the baseline specifically for MOT17-11 and MOT17-13 in terms of HOTA, AssA and/or MOTA. For MOT17-13, its lower performance can be entirely attributed to poorer detection, as reflected by its lower DetA and MOTA, yet similar AssA and MOTP scores. On the other hand, MOT17-11's lower performance is attributed to poorer association, which is driven by its high identity false positive count (IDFP) of 2,834 in comparison to the baseline's 1,951 over the entire MOT17 val_half set. This is the highest difference in IDFP between the two models across all videos (refer to Table 3).

Object Detector	AP	AP50	AP75	APs	APm	APl
Baseline	42.294	71.284	45.574	11.271	31.587	55.248
SELSAMOT ($n = 2.5, k = 5$)	41.866	71.062	44.805	10.522	31.603	54.457
SELSAMOT ($n = 2.5, k = 10$)	42.315	71.187	46.058	10.585	31.931	54.935
SELSAMOT ($n = 5, k = 5$)	41.221	69.243	44.419	8.454	30.726	53.989

Table 1. Object detection metrics. APs, APm, APl denote AP for small, medium and large objects, respectively. n and k refer to the $\pm n$ second window around the current frame from which we randomly sample k frames at inference time. All metrics reported on performance on MOT17 val_half.

Figure 5. An example tracking sequence from MOT17-05. First row: baseline. Second row: SELSAMOT. We see that the baseline assigns the original track of ID 295 as ID 322 after occlusion in the third frame shown, whereas SELSAMOT maintains an ID of 311 even after occlusion.

6.4. Number of context frames and size of context window

We also briefly inspect the effect of the number of randomly sampled frames and the size of the sampling window on object detection and tracking performance at test time. Results are reported in Tables 1 and 2. Interestingly, we see that, due to randomness, we can achieve performance on the object detection metrics that very marginally beats the baseline across most metrics when sampling double the frames from the same window. On the other hand, we see a fairly noticeable performance drop when we sample the same number of frames in a window double the size. This is most probably due to the nature of the dataset, where pedestrians are constantly moving, going in and out of the frame. Thus, sampling more frames in a smaller window will increase the likelihood of sampling frames with most instances being the same. Sampling from a larger window more sparsely will result in less of an overlap in detected instances, leading to worse performance.

We observe that increasing the number of frames sampled in a fixed window results in a negligible decrease in performance while sampling frames more sparsely results in a negligible increase. This is perhaps the case as the features aggregated in the same fixed window are limited in diversity by the window size and hence, increasing the number of samples may have marginal effect. On the flip side, the region proposal embeddings generated by aggregating over sparsely sampled frames would be noisier than the baseline's embeddings, possibly leading to lower confidence scores in the object detector's output. Given that ByteTrack [24] matches detections to existing and "lost" tracks in a 2-stage process, even with these lower confidence scores, matches that otherwise would have been made in the 1st similarity procedure would be caught by the 2nd matching procedure. As a result, performance would remain relatively the same in both settings.

That said, before any strong conclusion are to be made, further work to generate these object detection and tracking metrics over multiple runs must be done to determine whether these trends exist on average.

7. Conclusion

In this work, we attempt to improve multi-object tracking performance by leveraging the temporal dependence between frames in a video. Specifically, we incorporate SELSA [20] modules in our Faster R-CNN object detector to aggregate features at the region proposal level in the Region of Interest head. Unlike the success achieved by [20] in Video Object Detection, we are only able to achieve comparable performance to our baseline Faster R-CNN, which may be due to the higher complexity of our MOT dataset. However, it is worth mentioning that we achieve baseline re-

Method	HOTA↑	DetA↑	AssA↑	MOTA↑	MOTP↑	IDSW↓
Baseline	52.617	51.111	54.874	50.169	79.415	405
SELSAMOT ($n = 2.5, k = 5$)	51.389	48.791	54.675	47.645	79.045	383
SELSAMOT ($n = 2.5, k = 10$)	50.968	48.477	54.134	47.269	79.087	366
SELSAMOT ($n = 5, k = 5$)	51.458	48.836	54.782	48.601	79.062	363

Table 2. Tracking metrics. n and k refer to the $\pm n$ second window around the current frame from which we randomly sample k frames at inference time. All metrics reported on performance on MOT17 val_half.

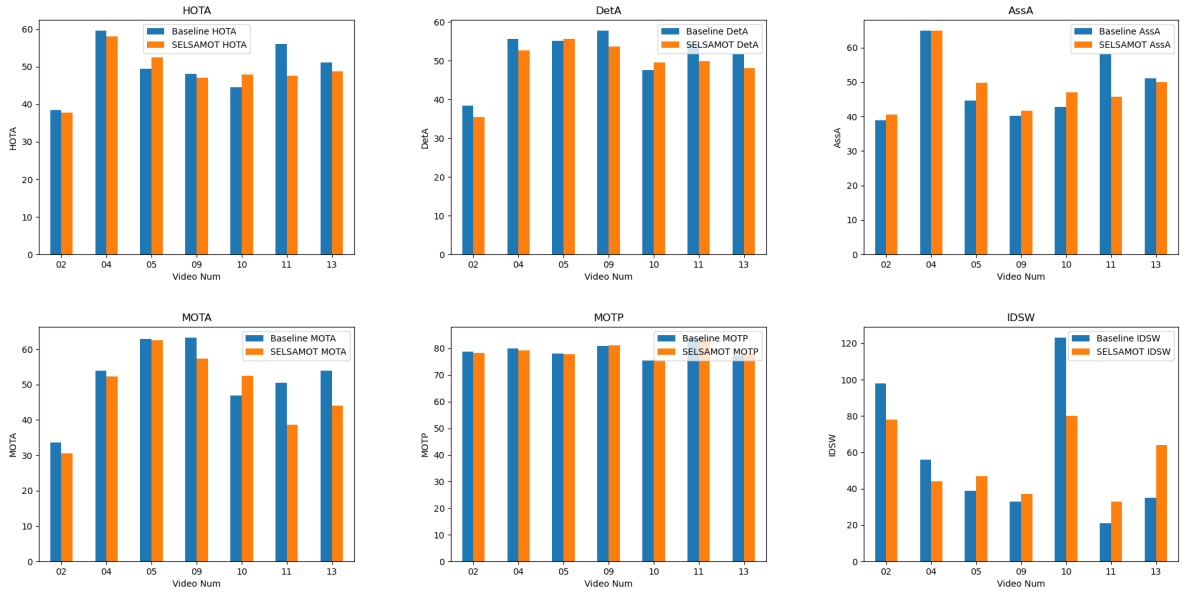

Figure 6. Tracking metrics per video in MOT17 val_half. First row: HOTA. Second row: CLEARMOT

sults after adding noise to our region proposal embeddings. Thus, future work can explore how these SELSA modules have or have not changed the quality of these embeddings. Moreover, researchers may investigate how incorporating these ROI features as appearance features in the tracking algorithm itself, similar to DeepSORT [19] and JRMOT [16], influences tracking.

8. Appendix

Please see Table 3 for per-video identity metrics for tracking mentioned in the paper.

9. Contributions & Acknowledgements

We used and modified code from detectron2[1], ByteTrack[2], and HOTA metrics[3]. We based our implementation of the SELSA module from the original code[4].

We thank Serena Yeung and the rest of the MARVL lab for their guidance. We used the lab's GCP credits to train and evaluate our models.

References

[1] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008. 4

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016. 2

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. 3, 4

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 2

[5] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection, 2020. 2

[1] https://github.com/facebookresearch/detectron2

[2] https://github.com/ifzhang/ByteTrack

[3] https://github.com/nekorobov/HOTA-metrics

[4] https://github.com/happywu/Sequence-Level-Semantics-Aggregation

Video	Model	IDF1	IDR	IDP	IDTP	IDFN	IDFP
MOT17-02	Baseline	46.455	41.184	53.273	4069	5811	3569
	SELSAMOT	44.847	38.381	53.933	3792	6088	3239
MOT17-04	Baseline	69.467	68.901	70.043	16659	7519	7125
	SELSAMOT	70.41	69.038	71.837	16692	7486	6544
MOT17-05	Baseline	63.444	61.186	65.876	2054	1303	1064
	SELSAMOT	69.078	65.982	72.48	2215	1142	841
MOT17-09	Baseline	58.216	56.235	60.343	1619	1260	1064
	SELSAMOT	57.008	56.791	57.228	1635	1244	1222
MOT17-10	Baseline	53.182	57.42	49.527	3401	2522	3466
	SELSAMOT	58.466	60.949	56.178	3610	2313	2816
MOT17-11	Baseline	66.17	70.799	62.109	3198	1319	1951
	SELSAMOT	54.176	60.46	49.075	2731	1786	2834
MOT17-13	Baseline	62.888	65.399	60.563	2064	1092	1344
	SELSAMOT	60.451	61.122	59.795	1929	1227	1297
COMBINED	Baseline	62.07	61.355	62.803	33064	20826	19583
	SELSAMOT	61.934	60.501	63.436	32604	21286	18793

Table 3. Identity metrics for tracking. All metrics reported on performance on MOT17 val_half.

[6] Patrick Dendorfer, Aljoša Ošep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking, 2020. 3, 4, 5

[7] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021, 2021. 3

[8] Julia Gong, F. Christopher Holsinger, and Serena Yeung. Flowvos: Weakly-supervised visual warping for detail-preserving and temporally consistent single-shot video object segmentation, 2021. 2

[9] Emmett D. Goodman, Krishna K. Patel, Yilun Zhang, William Locke, Chris J. Kennedy, Rohan Mehrotra, Stephen Ren, Melody Y. Guan, Maren Downing, Hao Wei Chen, Jevin Z. Clark, Gabriel A. Brat, and Serena Yeung. A real-time spatiotemporal ai model analyzes skill in open surgical videos, 2021. 1

[10] Patrick Dendorfer Philip Torr Andreas Geiger Laura Leal-Taixe Jonathon Luiten, Aljosa Osep and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 2020. 4, 5

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2016. 3

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 4

[13] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers, 2021. 2

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2, 5

[15] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 3, 5

[16] Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel, Amir Sadeghian, Hamid Rezatofighi, Roberto Martín-Martín, and Silvio Savarese. Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset, 2020. 1, 7

[17] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer, 2020. 3

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 2

[19] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017. 7

[20] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. *ICCV 2019*, 2019. 1, 2, 5, 6

[21] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3, 4

[22] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer, 2021. 2

[23] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017. 3, 4

[24] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang

Wang. Bytetrack: Multi-object tracking by associating every detection box, 2021. 2, 3, 6

[25] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection, 2017. 2

[26] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition, 2016. 2