End-to-end Amodal Video Instance Segmentation - Supplementary

Jasmin Breitenstein
j.breitenstein@tu-bs.de
Kangdong Jin
k.jin@tu-bs.de
Aziz Hakiri
a.hakiri@tu-bs.de
Marvin Klingner
m.klingner@tu-bs.de
Tim Fingscheidt
t.fingscheidt@tu-bs.de

Institute for Communications Technology,
Technische Universität Braunschweig
Schleinitzstraße 22, 38106 Braunschweig,
Germany

In this supplementary, we give additional information about our proposed `VATrack` method for end-to-end amodal video instance segmentation.

A Metrics

For our evaluation, we use the commonly used metrics average precision (AP) and video-based average precision (vAP) from instance and video instance segmentation (VIS), respectively. Here, we briefly recall their definition.

AP: Given true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), precision and recall are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{3}$$

In general, AP is defined as the area under the precision-recall curve. The definition used in this work stems from MS-COCO [13] and averages APs at IoU thresholds from 0.5 to 0.95. Metric AP_{50} is AP at an IoU threshold of 0.5. The IoU thresholds are used to determine correctness of a prediction, and are applied to the overlap of the instance masks $\mathbf{m}_{t,n}$ and $\overline{\mathbf{m}}_{t,n}$. Note that the image-based IoU is calculated for an instance n in a single frame t and is defined as:

$$\text{IoU}(\mathbf{m}_{t,n}, \overline{\mathbf{m}}_{t,n}) = \frac{|\mathbf{m}_{t,n} \cap \overline{\mathbf{m}}_{t,n}|}{|\mathbf{m}_{t,n} \cup \overline{\mathbf{m}}_{t,n}|}. \tag{4}$$

To calculate amodal metrics, we use the amodal instance masks $\mathbf{a}_{t,n}$. We simply use the `COCOAPI` [13] for evaluation. This is also the evaluation metric of the SAIL-VOS dataset [11] by considering next to AP and AP_{50} also specific AP_{50} values for small, medium, large objects, as well as partially and heavily occluded objects.

Figure 1: **Qualitative results** of `VATrack` (bottom) compared to `MaskJoint` [1] (top) for a sequence $\mathbf{x}_{t-3}^{t}$ with overlayed colorized amodal predictions $\mathbf{a}_{t-3}^{t}$ on SAIL-VOS-cut. `VATrack` detects and tracks all instances (green, pink and blue masks/bboxes) consistently across frames, while `MaskJoint` cannot exploit temporal context due to the missing tracking method (resulting in random mask/bbox colors) and thus, `MaskJoint` fails to detect severely occluded instances.

vAP: As there is not yet a video-based evaluation defined on the SAIL-VOS dataset, we borrow our proposed evaluation metric from the standard VIS task [19]. Here, the video-based average precision vAP is calculated as for instance segmentation, however, the video-based IoU between a predicted and ground truth instance is defined as [19],

$$\text{vIoU}(\mathbf{m}_{1,n}^{T}, \overline{\mathbf{m}}_{1,n}^{T}) = \frac{\sum_{t=1}^{T} |\mathbf{m}_{t,n} \cap \overline{\mathbf{m}}_{t,n}|}{\sum_{t=1}^{T} |\mathbf{m}_{t,n} \cup \overline{\mathbf{m}}_{t,n}|}, \tag{5}$$

i.e., a predicted instance $\mathbf{m}_{1,n}^{T}$ in a video sequence $\mathbf{x}_{1}^{T}$ is only considered a correct prediction if there is sufficient overlap with *all* the ground truth instance masks $\overline{\mathbf{m}}_{1,n}^{T}$. This is in contrast to the image-based AP where the correctness of a prediction is only based on the IoU between the instance masks in a single frame t. Hence, bad tracking capabilities will result in a low IoU. To calculate our amodal video evaluation, we use the amodal masks $\mathbf{a}_{t,n}$ and $\overline{\mathbf{a}}_{t,n}$, respectively, to calculate vIoU in (5).

B Implementation and Training Details

For comparability reasons, we train all methods using the `ResNext-101` backbone [18]. We use the full available image resolution of 1280×800. We report an ablation for smaller input resolutions in Supplementary Section C.3. For initialization, we use the respective ImageNet-pretrained weights.

Video-based methods: Our training protocols follow largely the `mmtracking` [6] repository and the details of the tracking methods [15, 19]. Using the `ResNext-101` backbone the peak learning rate is set to 0.0025. We have 2000 warm-up iterations with warm-up ratio $\frac{1}{2000}$, and we reduce the learning rate at epoch 8 and 11 by a factor of $\frac{1}{10}$.

Image-based methods: We use the standard training procedure of the `mmdetection` [4] repository.

All methods: We train all methods for 12 epochs on an `NVIDIA A100` GPU using a batch-size of 4. While we monitor results on the validation dataset, we found that the last (12th) model checkpoint performs best for all methods. We adopt hyperparameters for image and video approaches from prior art [4, 6, 15, 19] without further tuning. All our implementations are done in `PyTorch` [16] using version 1.7.0.

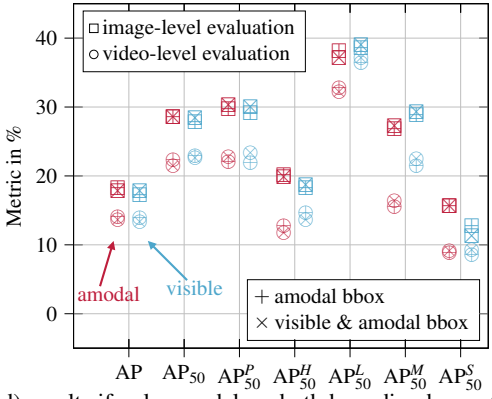

Figure 2: VATrack (QD-based) results if only amodal, or both bounding boxes (bboxes) are used in training the network. We show results for visible VIS (blue) and amodal VIS (red) on the SAIL-VOS-cut dataset for both image- and video-level evaluation.

Equation (2) in the main paper shows the loss function for joint training of amodal and visible VIS. We chose $\lambda_1 = \lambda_2 = 1$ as it is the standard setting used in both instance segmentation and VIS following the mmdetection [4] and mmtracking [6] repositories. Supplementary Section C.2 contains an ablation for various values of λ_1 and λ_2. During inference, we identify instance features $\mathbf{f}_1, \ldots, \mathbf{f}_{N_t}$ in the memory with new ones $\mathbf{f}_{t,n}$ of a current instance n, if the probability (i.e., softmax-based value for the MaskTrack R-CNN and bi-directional softmax value for the QDTrack tracking method) exceeds the value of 0.5, otherwise instance n is considered as a new instance.

C Ablation Studies

In this section, we show results of further ablation studies of our proposed VATrack (QD-based) method.

C.1 Qualitative Analysis and Ablation

Qualitative analysis: Fig. 1 shows qualitative results of QD-based VATrack (bottom) and the image-based MaskJoint method [11] (top). We observe that MaskJoint recovers well partial occlusions, e.g., for frame index $t-3$. However, it fails to detect the left person receiving the hug as the person is heavily occluded in all following frames. VATrack, on the other hand, is able to perceive all three persons in the sequence in all four frames. The person receiving the hug is detected when only partially occluded ($t-3$), heavily occluded ($t-2, t$), and even in the full occlusion case ($t-1$). The results show that image-based methods cannot adequately recover heavy and total occlusions, whereas our video-based VATrack can successfully use the temporal context for this purpose.

Ablation on bounding box heads: Fig. 2 shows results for employing different bounding box heads in the QD-based VATrack method, i.e., using only the amodal bounding boxes, or both bounding boxes. Using both means that the network is trained with two bounding box heads instead of one. For image-level evaluation, Fig. 2 shows that using both bounding boxes can lead to slight improvements in some of the metrics, e.g., visible AP^S_{50}. However, for the amodal image-level metrics as well as for most visible image-level metrics, we do not observe significant improvements when using additional visible bounding boxes, compared to using only the amodal bounding boxes.

	λ_1	λ_2	visible	amodal	AP	AP_{50}	AP^P_{50}	AP^H_{50}	AP^L_{50}	AP^M_{50}	AP^S_{50}
image-based metrics	1.0	1.0		✓	**18.3**	**28.6**	**29.7**	**20.1**	38.1	26.9	**15.7**
	0.5	1.5		✓	18.0	27.3	29.3	19.4	34.4	27.4	11.1
	1.5	0.5		✓	18.0	28.4	29.2	20.0	**40.2**	**28.4**	11.5
	0.75	0.25		✓	17.5	27.7	29.4	19.2	35.0	27.2	10.8
	1.0	1.0	✓		17.3	27.9	29.1	18.3	38.6	28.8	12.7
	0.5	1.5	✓		16.5	26.2	28.1	17.2	35.2	28.6	10.3
	1.5	0.5	✓		**17.9**	**28.3**	**29.3**	**19.0**	**41.3**	**31.0**	**13.4**
	0.75	0.25	✓		17.3	27.9	**29.3**	18.4	37.1	29.4	11.0
	λ_1	λ_2	visible	amodal	vAP	vAP_{50}	vAP^P_{50}	vAP^H_{50}	vAP^L_{50}	vAP^M_{50}	vAP^S_{50}
video-based metrics	1.0	1.0		✓	14.1	22.3	**22.0**	12.8	**32.8**	**15.6**	8.8
	0.5	1.5		✓	**14.6**	**22.8**	21.5	**13.8**	31.1	15.3	**9.2**
	1.5	0.5		✓	11.9	18.8	20.8	8.4	31.4	15.5	8.9
	0.75	0.25		✓	12.6	20.6	21.7	10.1	28.7	14.8	9.0
	1.0	1.0	✓		**14.0**	23.0	**21.9**	14.6	36.4	21.5	8.6
	0.5	1.5	✓		13.8	**23.3**	21.0	**15.5**	**37.0**	21.4	8.7
	1.5	0.5	✓		11.5	19.4	20.6	9.4	35.3	**21.7**	**9.0**
	0.75	0.25	✓		12.7	21.9	21.7	12.7	34.9	20.3	8.8

Table 5: **Image-level** results (%, upper part) and **video-level** results (%, lower part) of QD-based VATrack on the SAIL-VOS-cut validation data for different weights for the visible and amodal mask loss terms λ_1, λ_2 in (2) of the main paper. We show, whether metrics are reported for the visible or amodal masks by checkmarks. Best results are in **bold**, second best are underlined.

For video-level evaluation, we see that using both bounding boxes can improve the results in a few cases, i.e., amodal and visible vAP^P_{50} and vAP^M_{50} in Fig. 2. We do not observe a significant improvement in the video-level visible metrics (blue) when including a second (visible) bounding box head. Hypothetically, the amodal bounding boxes already carry enough localization guidance for the mask prediction so that the additional bounding box branch adds mostly computational complexity to the method. This is also supported by our results from Fig. 2, where using both amodal and visible bounding boxes mostly did not yield improvements. Thus, our proposed VATrack uses only the amodal bounding boxes.

C.2 Loss Weights Ablation for the Mask Heads

Equation (2) of the main paper shows the VATrack loss function J^{joint}. The influence of each mask head loss J^{visible} and J^{amodal} is governed by two pre-selected hyperparameters λ_1, λ_2, respectively. Table 5 reports results of QD-based VATrack for both image-based metrics (blue) and video-based metrics (orange) and for both amodal and visible masks. For the main results of our paper, we chose an equal weighting of both loss terms by 1.0 per default. We see in Table 5 that this leads to balanced results, e.g., the best amodal image-level AP of 18.3%, second best visible image-level AP of 17.3%, second best amodal video-level vAP of 14.1% and the best visible video-level vAP of 14.0%. In the image-based metrics, we observe that a larger weighting of the visible loss term ($\lambda_1 = 1.5$) compared to the amodal loss term ($\lambda_2 = 0.5$) improves clearly all visible image-based metrics (AP= 17.9% etc.). Interchanging this weighting to $\lambda_1 = 0.5, \lambda_2 = 1.5$ does not have the same effect on the amodal image-based metrics (no bold numbers in the second row of Table 5). Here, we actually observe performance drops compared to the equal weighting by 1.0, e.g., AP^L_{50} drops from 38.1% to 34.3%. Similarly, this effect is also not as pronounced for the video-based metrics, where favoring the amodal loss term J^{amodal} by setting $\lambda_2 = 1.5$ leads to improved performance for both amodal and visible video evaluation, e.g., amodal vAP_{50}

	input resolution	visible	amodal	AP	AP_{50}	AP^P_{50}	AP^H_{50}	AP^L_{50}	AP^M_{50}	AP^S_{50}
image-based metrics	1280×800		✓	18.3	28.6	29.7	20.1	38.1	26.9	15.7
	640×400		✓	13.8	22.2	23.1	15.2	32.1	19.8	6.5
	320×200		✓	7.8	13.1	13.7	10.3	22.3	7.6	0.2
	1280×800	✓		17.3	27.9	29.1	18.3	38.6	28.8	12.7
	640×400	✓		14.0	22.7	23.0	15.6	34.5	21.6	6.5
	320×200	✓		8.1	13.2	13.6	10.4	24.7	8.4	0.4
	input resolution	visible	amodal	vAP	vAP_{50}	vAP^P_{50}	vAP^H_{50}	vAP^L_{50}	vAP^M_{50}	vAP^S_{50}
video-based metrics	1280×800		✓	14.1	22.3	22.0	12.8	32.8	15.6	8.8
	640×400		✓	12.4	19.3	18.1	10.9	25.5	8.8	6.1
	320×200		✓	6.1	10.6	10.3	7.2	18.6	2.7	0.0
	1280×800	✓		14.0	23.0	21.9	14.6	36.4	21.5	8.6
	640×400	✓		12.2	20.0	18.1	12.1	33.4	14.9	6.0
	320×200	✓		6.4	11.1	10.3	8.6	26.6	7.3	0.0

Table 6: **Image-level** results (%, upper part) and **video-level** results (%, lower part) of QD-based VATrack on the SAIL-VOS-cut validation data for different input resolutions. We show, whether metrics are reported for the visible or amodal masks by checkmarks.

	Method	Backbone	V	A	TC	AP	AP_{50}	AP^P_{50}	AP^H_{50}	AP^L_{50}	AP^M_{50}	AP^S_{50}
image-based	MaskAmodal [13]	?		✓		13.0	23.0	24.3	16.7	36.6	21.5	6.1
	MaskJoint [13]	?	✓	✓		14.1	24.8	24.3	18.9	37.8	21.5	5.7
	MaskAmodal*	RX-101		✓		16.3	25.6	27.4	17.1	35.2	24.2	10.1
	MaskJoint *	RX-101	✓	✓		16.7	25.6	26.9	17.3	33.0	22.3	9.0
video-based	AmodalTrack (MT-based)	RX-101		✓	✓	15.9	25.7	24.9	17.8	36.8	22.8	11.2
	Ours: VATrack (MT-based)	RX-101	✓	✓	✓	16.4	26.0	24.9	18.0	38.6	22.5	10.6
	AmodalTrack (QD-based)	RX-101		✓	✓	17.8	27.4	29.2	18.6	34.7	26.8	11.4
	Ours: VATrack (QD-based)	RX-101	✓	✓	✓	**18.3**	**28.6**	**29.7**	**20.1**	**38.1**	**26.9**	**15.7**

Table 7: **Amodal** instance segmentation **image-level** performance (%) on the SAIL-VOS validation data for image-based methods, and for video-based methods. Marker $*$ denotes resimulated results. Checkmarks indicate whether the method predicts visible (V) and/or amodal (A) masks. The video-based methods take temporal context (TC) into account, shown by checkmarks. Results have been produced using the ResNext-101 (RX-101) backbone, Hu et al. [13] do not report their backbone (marked by: ?). Best results are in **bold**, second best are underlined.

improves to 22.8% compared to equal weighting (22.3%) and in the visible case we observe an improvement of +0.3% to 23.3% compared to equal weighting (23.0%). On video-level, weighting the visible loss term higher than the amodal one, i.e., $\lambda_1 = 1.5, \lambda_2 = 0.5$, does not lead to an improved visible video-level performance, in contrast to our results on image level.

To summarize, regarding our investigated set of hyperparameters, we see that results of course depend on this hyperparameter choice. This is a well known observation in multi-task learning [12]. In contrast to the natural expectation that amodal or visible performance can improve by a higher weighting of the respective loss term, our ablation does not confirm this. However, we can take away that our default equal weighting by 1.0 performs competitively with the other investigated sets of hyperparameters, and that interchanging our VATrack results of the main paper with those of another set of hyperparameters would not affect the experiments significantly. Instead tuning these hyperparameters could lead to additional gains.

Method	Backbone	V	A	vAP	vAP_{50}	vAP^{P}_{50}	vAP^{H}_{50}	vAP^{L}_{50}	vAP^{M}_{50}	vAP^{S}_{50}
`AmodalTrack` (MT-based)	`RX-101`		✓	0.1	0.2	0.3	0.1	1.0	0.1	0.0
Ours: `VATrack` (MT-based)	`RX-101`	✓	✓	2.3	3.1	3.8	**1.7**	3.7	1.5	0.3
`AmodalTrack` (QD-based)	`RX-101`		✓	**4.2**	**5.5**	**6.6**	0.7	4.1	3.1	**5.4**
Ours: `VATrack` (QD-based)	`RX-101`	✓	✓	3.8	5.4	6.5	0.8	**5.7**	**3.7**	**5.4**

Table 8: **Amodal** instance segmentation **video-level** performance (%) on the **SAIL-VOS** validation data. Checkmarks indicate whether the method predicts visible (V) and/or amodal (A) masks. Results have been produced using the `ResNext-101` (`RX-101`) backbone. Best results are in **bold**, second best are underlined.

C.3 Influence of Input Resolution

Here, we regard the influence of the input resolution of the video frames on the performance of `QD`-based `VATrack`. While for all our experiments in the main paper, we considered a resolution of 1280×800, which is the image resolution of the original SAIL-VOS dataset, we are interested in the impact of smaller input resolutions on the performance. Table 6 shows those results. We omit highlighting the best and second best methods as for this ablation, we of course expect performance to drop for smaller input resolutions. We show results for the original resolution 1280×800, 640×400 and 320×200. Our first observation is that the performance drop between 1280×800 and 640×400 is for most metrics (expect image-based AP^{S}_{50}) much smaller than the performance drop from resolution 640×400 to 320×200, e.g., amodal image-based AP_{50} drops from 28.6% (1280×800) to 22.2% (640×400) to 13.1% (320×200). Additionally, vAP^{S}_{50} drops for both amodal and visible evaluation to 0%, and AP^{S}_{50} drops to almost zero as well. From this, we deduce that using half of the original resolution can still lead to reasonable results for the sake of less memory consumption and shorter training times, however, an image resolution below seems no longer able to perform the task of end-to-end amodal VIS as it even fails detecting a certain range of instances at all.

D Results on SAIL-VOS dataset

In this section, we show the evaluation results for all methods on the SAIL-VOS validation data. Note that we trained all methods on the SAIL-VOS-cut training data. For the image-based methods, this has no impact as for them SAIL-VOS = SAIL-VOS-cut holds. The video-based methods cannot be trained on the SAIL-VOS training data. The jump cuts prevent convergence during the training process by significantly increasing all loss terms whenever a jump cut is part of the current batch.

In Table 7, we show amodal image-level results on the SAIL-VOS validation set. As in the main paper, we highlight image-based methods in blue and video-based methods in orange. We observe that the results for the image-based methods (in blue) do not change compared to Table 1 (main paper). Not taking temporal context into account, SAIL-VOS-cut and SAIL-VOS are identical. Interestingly, the image-level results do also not change for the video-based methods, i.e., the longer videos of SAIL-VOS do not provide additional temporal context for improved amodal mask predictions and the jump cuts do not affect the image-level performance. However, as expected, we see a significant drop in amodal video-level performance as can be seen in Table 8 compared to Table 3 (main paper), as the jump cuts of SAIL-VOS naturally hinder the tracking performance.

Method	Backbone	V	A	vAP	vAP_{50}	vAP^{P}_{50}	vAP^{H}_{50}	vAP^{L}_{50}	vAP^{M}_{50}	vAP^{S}_{50}
MaskTrack R-CNN	RX-101	✓		0.1	0.2	0.3	0.1	1.1	0.2	0.0
Ours: VATrack (MT-based)	RX-101	✓	✓	2.3	3.1	3.8	1.6	4.4	1.4	0.3
QDTrack	RX-101	✓		**3.4**	**9.5**	6.2	**5.4**	8.5	3.1	**5.6**
Ours: VATrack (QD-based)	RX-101	✓	✓	3.2	7.1	**6.6**	2.7	**9.0**	**4.6**	5.5

Table 9: **Visible** instance segmentation **video-level** performance (%) on the **SAIL-VOS** validation data. Checkmarks indicate whether the method predicts visible (V) and/or amodal (A) masks. Results have been produced using the ResNext-101 (RX-101) backbone. Best results are in **bold**, second best are underlined..

Table 8 shows the results of amodal video instance segmentation on the SAIL-VOS validation data. In comparison to Table 3 (main paper) these results impressively show the effect of the jump cuts in SAIL-VOS: Performance in all metrics drops significantly as the methods are no longer able to track instances throughout an entire video sequence, and especially small instances can no longer be recognized and tracked (see vAP^{S}_{50} for all methods). For these small instances at jump cuts, identification with previous instances apparently becomes the hardest. In these cases, SAIL-VOS-cut resets the instance set $\mathcal{N}_t$ at jump cuts (by considering this the starting and end point of a sequence), which improves the performance considerably (see main paper Tables 3, 4).

Table 9 shows the results of visible video instance segmentation on the SAIL-VOS validation data. Results follow the impression of Table 8: As for the amodal video-level evaluation, all performance metrics drop significantly compared to the results on SAIL-VOS-cut validation data in Table 4 (main paper). Also for the visible video instance segmentation, the jump cuts prevent concise identification and tracking of instances throughout the video sequences.

E Limitations and Ethical Implications

Limitations: Our proposed QD-based VATrack outperforms the image-based state of the art on the SAIL-VOS (=SAIL-VOS-cut) validation data. Additionally, VATrack establishes a new video-based state of the art on the SAIL-VOS-cut validation data and can be used as reference in future work. However, it needs to be noted that we did not investigate the ideal hyperparameter choice λ_1, λ_2 in Equation (2) of the main paper for all methods. Instead, for our main results we chose per default the hyperparameters according to [19], $\lambda_1 = \lambda_2 = 1.0$, for all investigated methods of our work. Table 5 gives an insight, how results for the proposed QD-based VATrack depend on this parameter choice for a limited subset of values for λ_1, λ_2. It is widely known in the field of multi-task learning, that results might be sensitive to this choice of parameters [5, 12]. So while for QD-based VATrack the default choice $\lambda_1 = \lambda_2 = 1.0$ performs competitively to the other hyperparameter choices (see Table 5), we did not investigate this for the other joint (baseline) methods, i.e., MT-based VATrack and MaskJoint. This study is not typically part of either instance segmentation or VIS [2, 9, 15, 17, 19] and remains for future work.

Additionally, SAIL-VOS, to the best of our knowledge, is the only publicly accessible dataset meeting our label requirements. So to generalize to other datasets, for future work we aim to investigate softening the hard label requirements of our method to weaker supervision and few-shot learning.

Ethical implications: Since SAIL-VOS is the only dataset providing video-level amodal

ground truth labels while also enabling a comparison against the state of the art, it was chosen based on practical availability for training and evaluation of `VATrack`. However, as the SAIL-VOS dataset is derived from the GTA V game engine, it suffers from existing criticism of the GTA V video game. It has faced criticism for its misogynistic portrayal of women and hypermasculine portrayal of men [10], which thus has been transported to the SAIL-VOS video data as well. There have been many investigations into the effect of such stereotypical and violence-prone portrayals of humans in video games [1, 7, 8, 14], which can, e.g., lead to a correlation with higher tolerance to sexual harassment [7]. Moreover, most persons depicted in the SAIL-VOS dataset are men, which will likely result in worse perception rates for women overall [3]. So the SAIL-VOS dataset has to be considered carefully under those ethical controversies. For this work, it was chosen due to its practical availability and comparability to prior art.

While we investigate amodal perception towards making applications of perception methods safer, e.g., automated driving, medical imaging, or general robotic movements, there is of course a second side to this as well. At the moment, amodal perception is already part of first-person shooting games where the player can use it to have an advantage over his or her opponents by being able to observe them while occluded or hidden. While in simulated worlds, this amodal knowledge is given by the scenario generation pipeline and is not learned, learned methods could potentially bring this technology to the real world and provide the means to weaponize it. This is true to the same amount for any environment perception and tracking method.

References

[1] Craig A Anderson, Nicholas L Carnagey, Mindy Flanagan, Arlin J Benjamin, Janie Eubanks, and Jeffery C Valentine. Violent Video Games: Specific Effects of Violent Content on Aggressive Thoughts and Behavior. *Advances in Experimental Social Psychology*, 36:200–251, August 2004.

[2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time Instance Segmentation. In *Proc. of ICCV*, pages 9157–9166, Seoul, Korea, October 2019.

[3] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proc. of FAccT*, pages 77–91, New York, NY, USA, February 2018.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*, pages 1–13, June 2019.

[5] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *Proc. of ICML*, pages 794–803, Stockholm, Sweden, July 2018.

[6] MMTracking Contributors. MMTracking: OpenMMLab Video Perception Toolbox and Benchmark. https://github.com/open-mmlab/mmtracking, 2020.

[7] Karen E. Dill, Brian P. Brown, and Michael A. Collins. Effects of exposure to sex-stereotyped video game characters on tolerance of sexual harassment. *Journal of Experimental Social Psychology*, 44(5):1402–1408, September 2008.

[8] Tobias Greitemeyer and Dirk O Mügge. Video Games do Affect Social Outcomes: A Meta-analytic Review of the Effects of Violent and Prosocial Video Game Play. *Personality and Social Psychology Bulletin*, 40(5):578–589, May 2014.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. of ICCV*, pages 2980–2988, Venice, Italy, October 2017.

[10] Kevin Hoffin and Geraldine Lee-Treweek. The Normalisation of Sexual Deviance and Sexual Violence in Video Games. In *Video Games Crime and Next-Gen Deviance*, pages 151–174, July 2020.

[11] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G. Schwing. SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation – A Synthetic Dataset and Baselines. In *Proc. of CVPR*, pages 3105–3115, Long Beach, CA, USA, June 2019.

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proc. of CVPR*, pages 7482–7491, Salt Lake City, UT, USA, June 2018.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. of ECCV*, pages 740–755, Zurich, Switzerland, September 2014.

[14] Yanling Liu, Zhaojun Teng, Haiying Lan, Xin Zhang, and Dezhong Yao. Short-term Effects of Prosocial Video Games on Aggression: An Event-related Potential Study. *Frontiers in Behavioral Neuroscience*, 9:1–12, July 2015.

[15] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-Dense Similarity Learning for Multiple Object Tracking. In *Proc. of CVPR*, pages 164–173, June 2021.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. of NeurIPS*, pages 8024–8035, Vancouver, BC, Canada, December 2019.

[17] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and Fast Instance Segmentation. In *Proc. of NeurIPS*, pages 17721–17732, virtual, December 2020.

[18] Saining Xie, Ross Girshick, Piotr Dollár, Zhouwen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Proc. of CVPR*, pages 5987–5995, Honulu, HI, USA, July 2017.

[19] Linjie Yang, Yuchen Fan, and Ning Xu. Video Instance Segmentation. In *Proc. of ICCV*, pages 5188–5197, Seoul, Korea, October 2019.