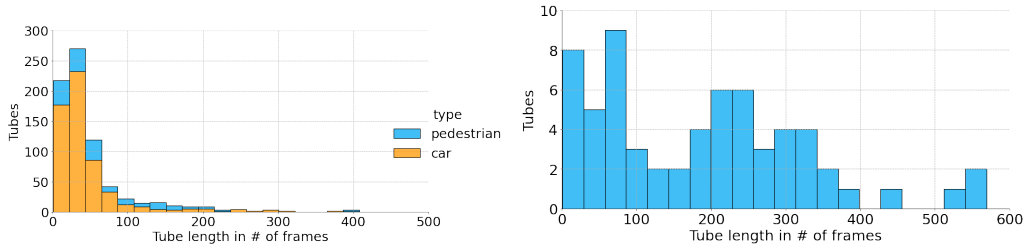

Figure 6: Track length distributions of KITTI-STEP (left) and MOTChallenge-STEP (right).

Cityscapes-VPS (val)	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	All things
Total instances	621	87	791	24	19	2	29	174	1747
Single frame instances	306	48	343	16	8	1	16	80	818
Spanning tracks	315	39	448	8	11	1	13	94	929
Average instance length	3.2	2.9	3.5	2.0	2.5	3.5	3.0	3.6	3.3

Table 4: Per-class statistics of the *validation* set of Cityscapes-VPS [34]. Only half of all instances need to actually be tracked, *i.e.*, appear in more than one frame. On average, instances appear for only 3 frames showing the focus on short trajectories.

A Appendix

In this supplementary material, we provide

B. Benchmark checklist,
C. An extended tracking difficulty discussion (Sec. C),
D. More of our collected dataset statistics, and details about merging our *semantic segmentation* annotations with existing MOTS instance annotations [61] (Sec. D),
E. More discussion about metric design choices (Sec. E),
F. Network architecture details of our proposed unified STEP model, *Motion-DeepLab* (B4) (Sec. F),
G. More details on the experiments described in the main paper (Sec. G),
H. STQ on Cityscapes-VPS (Sec. H),
I. Video of qualitative results: https://youtu.be/NHBAvT0DXVw,
J. Code: https://github.com/google-research/deeplab2.

B Benchmark Checklist

Code. https://github.com/google-research/deeplab2 includes instructions and code to reproduce our baselines.
KITTI. http://cvlibs.net/datasets/kitti/eval_step.php includes links to the dataset, setup instructions and the test server.
MOTCh. https://motchallenge.net/data/STEP-ICCV21/ includes links to the dataset, setup instructions and the test server.
ICCV. The proposed benchmarks are part of the 6th BMTT workshop at ICCV: https://motchallenge.net/workshops/bmtt2021/.
Resp. The authors of this work take full responsibility for the presented work. The maintainers and creators of the original KITTI and MOTChallenge benchmarks and the MOTS datasets did consent to this project. We respect their dataset license and release the test servers in their framework.
Hosting. The benchmarks will be hosted on the long-standing benchmark servers of KITTI and MOTChallenge. The newly provided dataset annotations will be hosted by Google. We take responsibility for the maintenance and ensure that the datasets will be accessible.
License. We release the benchmarks with the corresponding licenses of the original datasets. The code lincense can be found at the corresponding website.
Format. The dataset annotations are released as PNGs.

KITTI-STEP (val)	Person	Car	All things
Total instances	151	68	219
Single frame instances	0	0	0
Spanning tracks	151	68	219
Average instance length	53.2	49.2	51.9

MOTChallenge (val)	Person
Total instances	26
Single frame instances	0
Spanning tracks	26
Average instance length	183.6

Table 5: Per-class statistics of the *validation* set of KITTI-STEP (left) and MOTChallenge-STEP (right).

C Tracking Difficulty Discussion

We provide a per-class breakdown of track numbers for the validation sets of Cityscapes-VPS [34] (Tab. 4), KITTI-STEP and MOTChallenge-STEP (Tab. 5). Furthermore, we show the track length distribution for both datasets in Fig. 6.

Why do only 'pedestrians' and 'cars' have tracking IDs? In Fig. 3 in the main paper, we illustrate the class-wise histograms, *i.e.*, amount of pixels in the whole dataset. As shown in the figure, for KITTI-STEP, both 'cars' and 'pedestrians' contain significantly more pixels compared to the rest of the classes, while for MOTChallenge-STEP, the 'pedestrians' dominate the others. Due to the available annotation budget we therefore decided to focus on tracking the most salient object classes 'pedestrians' and 'cars' for KITTI-STEP, and only 'pedestrians' for MOTChallenge-STEP. This follows the original approach by KITTI-MOTS and MOTSChallange. We note that the number of tracking classes does **not** causally relate to the tracking difficulty. The latter is influenced by simultaneously present objects, occlusions and sequence length. The tracking difficulty and detection difficulty are changed by different factors. As our proposed benchmarks aim to balance segmentation/detection and tracking, we focus on increasing tracking difficulty w.r.t. previous work in this area.

Long-term tracking. In Fig. 6, we show the histograms for tracklet lengths for both datasets. As shown in the figure, our KITTI-STEP and MOTChallenge-STEP present a challenge for long term consistency in segmentation and tracking.

Do more instances or classes lead to harder tracking? Even though, our datasets provide twice the number of masks for the trainval set compared to Cityscapes-VPS, Cityscapes-VPS provides significantly more unique *instances*. Here, instances refer to unique objects not distinguishing for how many frames they are visible. We refrain from calling them tracks, as tracks are usually implied to last more than a single frame. When comparing these numbers directly, Tab. 4 shows that the validation set provides 1747 unique instances, while KITTI-STEP and MOTChallenge-STEP (Tab. 5) provide only 219 and 26 instances. In the following we provide several reasons why more instances does not imply harder tracking:

1. In Cityscapes-VPS, almost half (818) of all instances (1747) only last for a single frame, hence requiring no tracking at all. This shows the strong focus of Cityscapes-VPS on the segmentation aspect.
2. The average length of instances is 3.3 on Cityscapes-VPS, *i.e.*, instances need to be tracked for 2.3 frames after being detected. Hence, Cityscapes-VPS is not suitable to measure tracking performance. On KITTI-STEP and MOTChallenge-STEP, the average length are 51.9 and 183.6 frames, respectively.
3. The validation set of Cityscapes-VPS consists of 50 clips while our datasets contain 10 videos in the validation set. Naturally, every new clip and video always introduces a new set of instances. However, in long videos, instances continue to exist throughout (part of) the video resulting in less new instances. Yet, exactly these instances are the ones that need to be tracked and the ones that make tracking challenging.

We therefore infer that the proposed datasets KITTI-STEP and MOTChallenge-STEP are significantly more suitable when evaluating segmentation and tracking than Cityscapes-VPS. Measuring tracking requires (long) trajectories in the data.

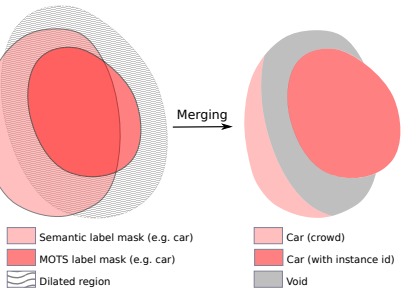

Figure 7: Illustration of how we merge the semantic and MOTS annotations. A large dilation region is chosen for illustration purpose.

D Extended Dataset Discussion

Merging annotation. We merge our new semantic segmentation annotation with the existing tracking instance ground-truth, *i.e.*, instance identity from the KITTI-MOTS and MOTS-Challenge [61]. We refer to their annotations as MOTS annotations for simplicity. Fig. 2 from the main paper gives an example of our annotation process. During the merging process, potential inconsistencies between our annotated semantic labels for the classes 'pedestrian' and 'car' and the MOTS instance annotations need to be carefully handled. For example, following the 'pedestrian' definition in Cityscapes [17], our semantic annotation includes items carried (but not touching the ground) by the person, while MOTS annotations exclude those items. With the aim to build a dataset that is compatible for both Cityscapes and MOTS definitions, we adopt the following strategy to merge the semantic annotations. For the cases where our annotated semantic mask is larger than MOTS annotated mask, we dilate the MOTS instance annotations with a kernel size of 15. The difference between the intersection of our annotation with the the enlarged and with the original MOTS annotations is re-labeled as 'VOID'. In practice, the union regions are detected by dilating the MOTS instance annotations with a kernel size 15. For the cases where our annotated semantic mask is smaller than MOTS annotated mask, the inconsistent regions are overwritten with MOTS annotations. As a result, the consistent 'pedestrian' masks are annotated with both semantic class and tracking IDs. Small regions along the masks are annotated with 'void', while personal items are annotated with only the semantic class (and tracking ID 0). Additionally, the ignored regions in the MOTS annotations are filled with our semantic annotation. For the crowd regions, *e.g.*, a group of 'pedestrians' that could not be distinguished by annotators, semantic labels are annotated but their tracking ID is set to 0. More technically, we denote each MOTS instance mask with semantic label l and instance ID i as M_l^i, and the semantic annotation mask with label k as S_k. We first dilate M_l^i using a kernel with 15 pixels. There are three cases for each S_k:

Case I: S_k intersects with M_l^i. The intersection is overwritten with label l with instance ID = i.

Case II: S_k intersects with the expanded dilated region. The intersection is re-labeled as 'void' if $k = l$.

Case III: The semantic label of the rest without any intersection remains the same.

This process is summarized in Fig. 7. We refer to the merged datasets as KITTI-STEP and MOTChallenge-STEP, respectively.

E Extended Metric Discussion

In addition to the overview of metrics in the main paper, we discuss one more common metric in video instance segmentation [65], and explain why it is unsuitable for STEP. After that, we will discuss our design choices of Segmentation and Tracking Quality (STQ).

E.1 Track-mAP (AP^{track})

For the task of Video Instance Segmentation [65], a variant of AP^{mask} [24, 40] is used to measure the quality of predictions. Like AP^{mask}, AP^{track} allows overlapping predictions, and hence requires

confidence scores to rank instance proposals. These predictions are then matched with an IoU threshold. Moreover, as established in prior work [43], this metric can be gamed by making lots of low-confidence predictions, and the removal of correct detections with wrong track ID can improve scores. We therefore consider this metric unsuitable for our benchmark.

E.2 STQ Design Choices

As stated in the main paper, we define the association quality (AQ) as follows:

$$AQ(g) = \frac{1}{|gt_{id}(g)|} \sum_{p,|p\cap g|\neq\emptyset} TPA(p,g) \times IoU_{id}(p,g),$$
$$AQ = \frac{1}{|\text{gt\_tracks}|} \sum_{g\in\text{gt\_tracks}} AQ(g). \tag{12}$$

Precision and Recall. While it is common for recognition and segmentation tasks to consider both recall and precision, this has not been widely adapted for tracking yet. For example, MOTSA does not consider precision for association. However, precision is important to penalize predicted associations that are false, *i.e.*, *false positive associations*. Consider the following example: All cars in a sequences are segmented perfectly and are assigned the same track ID. As all ground-truth pixels are covered, this gives perfect recall. Yet, the overall prediction is far from being perfect by assigning the same track ID to different cars. Hence, precision is an important aspect of measuring the quality of a prediction. The other aspect to consider is recall. Considering the same example with perfect segmentation, a perfect association precision can be trivially achieved by assigning a different track ID to every pixel. As there are no false positives associations, the overall score is perfect. Yet, this does not fulfill the purpose of measuring the quality of association. Therefore, both aspects, precision and recall, have to be considered for a good metric measuring association.

IoU *vs.* F1. The two most common approaches to combine precision and recall in computer vision are Intersection-over-Union, also known as the Jaccard Index, and F1, also known as the dice coefficient. IoU and F1 correlate positively and are thus both valid measures. We chose IoU for two reasons:

1. We already adopted the IoU metric for measuring segmentation. Using it for association as well leads to a more consistent formulation.
2. When comparing F1 and the IoU score, the F1 score is the harmonic mean of precision and recall and therefore closer to the average of both terms. On the other hand, IoU is somewhat closer to the minimum of precision and recall. Choosing IoU over F1 therefore emphasizes that good predictions need to consider recall and precision for association as well as highlighting innovation better.

Weighting factor *TPA*. A simpler version of equation (12) would compute the average IoU_{id} score without any weighting, and by normalizing with the number of partially overlapping predictions w.r.t. the ground-truth track. However, this formulation has the disadvantage that it does not consider long-term consistency of each track. Given two predicted tracks A and B, whether the IoU to the single ground-truth track are $3/5$ and $2/5$ or $4/5$ and $1/5$, both would achieve the exact same result:

$$\frac{1}{2} \times \left(\frac{3}{5} + \frac{2}{5}\right) = \frac{1}{2} \times \left(\frac{4}{5} + \frac{1}{5}\right) = \frac{1}{2} \tag{13}$$

As our goal is long-term consistency, we weight each IoU_{id} with the TPA. This factor increases the importance of long-term prediction:

$$\frac{1}{5} \times \left(3 \times \frac{3}{5} + 2 \times \frac{2}{5}\right) = \frac{13}{25} \tag{14}$$
$$\frac{1}{5} \times \left(4 \times \frac{4}{5} + 1 \times \frac{1}{5}\right) = \frac{17}{25} \tag{15}$$

Thus, our formulation of AQ fulfills the property of getting a higher score for predictions that have overall higher long-term consistency.

Normalization by ground-truth size. When considering the normalization factor of equation (12), one natural question that could come up is, why do we propose this denominator instead of the sum of all *TPA*. The reason is that otherwise the removal of correctly segmented regions with wrong track ID could achieve a higher score. Consider two predicted car tracks overlapping a ground-truth car track with IoU $4/5$ and $1/5$, respectively. Changing the denominator would lead to the following scores, with and and without the removal of the second track:

$$\frac{1}{5} \times \left(4 \times \frac{4}{5} + 1 \times \frac{1}{5}\right) = \frac{17}{25} \tag{16}$$

$$\frac{1}{4} \times \left(4 \times \frac{4}{5}\right) = \frac{16}{20} = \frac{20}{25} \tag{17}$$

Hence, the removed segment leads to a higher score. In contrast, in our current formulation we achieve the following scores in this scenario:

$$\frac{1}{5} \times \left(4 \times \frac{4}{5} + 1 \times \frac{1}{5}\right) = \frac{17}{25} \tag{18}$$

$$\frac{1}{5} \times \left(4 \times \frac{4}{5}\right) = \frac{16}{25} \tag{19}$$

Therefore, it will always be better to recognize cars (or other objects) than not to detect them. This still holds when looking at the overall metric. Even though the removal of correct segments is already penalized in the segmentation quality, that penalty would would rarely be noticeable when the association quality score would increase in that case. Hence, setting the denominator to the ground-truth size aligns with the importance of not removing predictions. For example, in an autonomous driving scenario, it is critical that correct pedestrian predictions are kept, even though they have a wrong track ID.

Class-aware *vs*. Class-agnostic Association. In previous metrics, VPQ [34] and PTQ [29] tracks must have the correct semantic class assigned to count as *true positives*. Such design couples segmentation and association errors, *e.g.*, a car track mistaken for a van would receive a score of 0 even though is is perfectly tracked throughout a sequence. In our setting, we compare three options to design the association score w.r.t. to semantic classes.

1. Require the *correct* semantic class of predicted tracks to be matched to ground-truth tracks to compute association sores.
2. Require *one* (but any) semantic class of predicted tracks to be matched to ground-truth tracks to compute association sores.
3. Allow *any* semantic thing class to be assigned to pixels of predicted tracks.

Option 1 penalizes wrong semantic segmentation twice and therefore completely couples segmentation and association errors like VPQ and PTQ. The 2nd option has the problem that correcting semantic classes receives a lower score than not correcting them. A prediction that at first mistakes a van for a car should not be penalized, when the semantic class is changed to the correct one. When requiring one semantic class, a prediction that changes the semantic class would create a new track. This would result in an overall reduced score, which violates the goal of not penalizing the correction of mistakes. Therefore, we have chosen the 3rd option for the design of our STQ metric.

Implementation Details. We need to consider two special cases for the implementation of the STQ metric. The first case is the *crowd* region. For far away or highly overlapping objects, it can be impossible for human annotators to distinguish different instances. In those cases, we can still assign the correct semantic class to these pixels. During evaluation, we cannot measure any association quality, but also do not want to penalize (potentially correct) track ID assignment by a network. We therefore consider the semantic class of these regions for measuring the segmentation quality.

	#1	#2	#3	#4	#5
SQ	1.0	1.0	1.0	1.0	0.75
AQ	$\frac{1}{2\times2}(\frac{2\times2}{4}+\frac{2\times2}{4})=0.5$	$\frac{1}{5}(\frac{2\times2}{5}+\frac{3\times3}{5})=\frac{13}{25}$	$\frac{1}{5}(\frac{1\times1}{5}+\frac{4\times4}{5})=\frac{17}{25}$	$\frac{1}{4}(\frac{1\times1}{4}+\frac{3\times3}{4})=\frac{5}{8}$	$\frac{1}{4}(\frac{3\times3}{4})=\frac{9}{16}$
STQ	$\sqrt{1\times0.5}=0.71$	$\sqrt{1\times\frac{13}{25}}=0.72$	$\sqrt{1\times\frac{17}{25}}=0.82$	$\sqrt{1\times\frac{5}{8}}=0.79$	$\sqrt{\frac{3}{4}\times\frac{9}{16}}=0.65$
PTQ	$\frac{4-0}{4+0+0}=1.0$	$\frac{5-1}{5+0+0}=0.8$	$\frac{5-1}{5+0+0}=0.8$	$\frac{4-1}{4+0+0}=0.75$	$\frac{3-0}{3+\frac{1}{2}+0}=0.86$
VPQ†	$\frac{0}{0+\frac{1}{2}+2\times\frac{1}{2}}=0$	$\frac{0.6}{1+\frac{1}{2}+0}=0.4$	$\frac{0.8}{1+\frac{1}{2}+0}=0.53$	$\frac{0.75}{1+\frac{1}{2}+0}=0.5$	$\frac{0.75}{1+0+0}=0.75$

Table 6: Intermediate computation steps for Fig. 3 of the main paper. VPQ† refers to the VPQ evaluation over the complete scene.

For the association quality, these pixel regions are ignored, which means there is no penalty for assigning track IDs to these regions. The second case to consider is the *ignore* label. Ignore labels are commonly used by annotators for regions, which can not be assigned to one of the limited semantic classes and should therefore be ignored during evaluation. However, [36] introduced this concept for predictions, too. Predicted ignore segments do not count as false positives, which lead to common post-processing steps in the field of panoptic segmentation. Specifically, small predicted segments are overwritten with the void label. Since we would not like to encourage such tricks, we adopt the following strategy to handle void label. For the segmentation quality, we allow an additional class void, which is handled like all other classes, except that all ignore regions in the ground-truth will not be considered. Thus, there is no advantage of predicting void labels, but we still allow to evaluate output of methods that require such predictions by design.

Detailed metric scores of illustration. We provide intermediate computation steps to obtain the results of Fig. 3 of the main paper in Tab. 6.

F Network Architecture

Single-frame baselines. Our single-frame baselines build on top of Panoptic-DeepLab [15] by additionally using three different methods to infer the tracking IDs. The adopted separate architectures are therefore the same as the original works [15, 58].

Multi-frame baseline. Motivated by [4, 67], our multi-frame baseline, *'Motion-DeepLab'*, extends Panoptic-DeepLab [15] by adding another prediction head, *Previous Center Regression*, which assists in associating predicted instances between two consecutive frames. Additionally, same as CenterTrack [67], the previous predicted center heatmap and the previous image frame are given as additional inputs to the network. The network architecture is visualized in Fig. 8.

G Experimental Results

Training Protocol. Using Panotic-DeepLab [14] as our base network, we follow closely the same training protocol as in [14]. Specifically, all our models are trained using TensorFlow [1] on 16 TPUs and batch size 32. We use the 'poly' learning rate policy [41], fine-tune the batch normalization parameters [31], adopt random scale data augmentation during training with Adam [35] optimizer. Our model is pretrained on Cityscapes [17] (only image panoptic annotations are exploited) for 60k iterations with an initial learning rate of 2.5e-4. For the single-frame baselines (B1-B3), we fine-tune on KITTI-STEP and MOTChallenge-STEP with an initial learning rate of 1e-5 for 30k and 1.4k iterations, respectively. For our *Motion-DeepLab* baseline (B4), we have to conduct net-surgery on the weights of the first convolution of the ResNet pre-trained checkpoint [26]. The baseline B4 takes 7 channels as input (3 channels for current frame, 3 channels for previous frame, and 1 channel for the previous frame center heatmap). We therefore take the weights of the first 7×7 convolution and duplicate them to get to 6 channels. Finally, the weights of the last channel are obtained by taking another duplicate and average over the channel dimension. With these pre-trained weights, we fine-tune on KITTI-STEP and MOTChallenge-STEP with a learning rate of 1e-5 for 50k and 2k iterations, respectively. For VPSNet [34], we use the default training settings to pre-train on Cityscapes-VPS without the tracking head. Then, we fine-tune the full network on KITTI-STEP

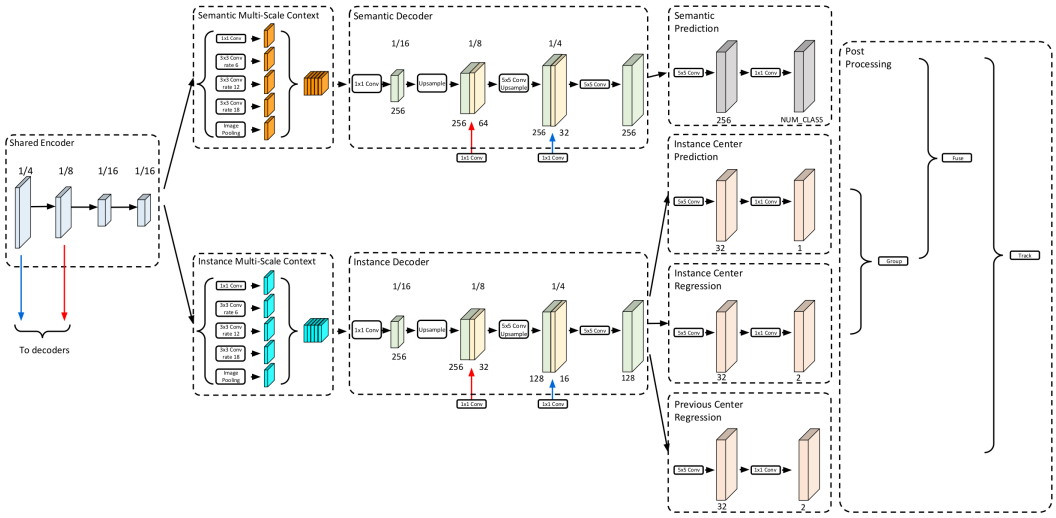

Figure 8: The architecture of the *Motion-DeepLab* baseline (B4).

Baseline	Pretrained	STQ	AQ	SQ	PQ	RQ	SQ
B3: Mask Propagation	✗	0.57	0.59	0.55	0.36	0.46	0.72
B3: Mask Propagation	✓	0.67	0.63	0.71	0.47	0.57	0.79
B4: Motion-DeepLab	✗	0.54	0.55	0.53	0.34	0.43	0.69
B4: Motion-DeepLab	✓	0.58	0.51	0.67	0.43	0.54	0.78

Table 7: **Effect of pretraining** on Cityscapes with results on KITTI-STEP.

and MOTChallenge-STEP again with the optimized default settings. As we observe overfitting on MOTChallenge-STEP, we reduce the training iterations to $1/3$ of the original number.

Qualitative Results. Please refer to the attached files for video visualization of our dataset ground truth and our model predictions (B3).

Effect of pre-training. In Tab. 7, we report the effect of pretraining our networks on Cityscapes before finetuning on KITTI-STEP. As shown in the table, pretraining brings 10%, and 4% improvement of STQ for baselines B3 and B4, respectively. For B4, we observe performance gain in SQ, and slightly degradation in AQ, presenting a challenging research problem to efficiently develop a unified STEP model. When comparing the non-pretrained networks, the unified model B4 has a much smaller gap to the B3 model than with pretraining. We hope our baseline could serve as a strong baseline to facilitate the research along the direction of developing a better unified STEP model.

H STQ on Cityscapes VPS

In Tab. 9 and Tab. 8, we show scores of our metric with VPSNet on Cityscapes-VPS. In the following, we draw insights from these numbers.

Metric insights on Cityscapes-VPS. In Tab. 8, we provide STQ scores of VPSNet on Cityscapes-VPS. Notably, the AQ score is lower than on KITTI-STEP but higher than on MOTChallenge-STEP. In the following, we study the behavior of AQ. For that we also provide per-class scores on Cityscapes-VPS in Tab. 9.

Due to our unified treatment of space and time, STQ can evaluate single frame performance, short clip performance as well as long video performance. Depending on the dataset, the AQ score will

Cityscapes-VPS (val)	STQ	AQ	SQ	VPQ	VPQ^{Th}	VPQ^{St}
VPSNet	0.50	0.35	0.72	0.57	0.44	0.67

Table 8: VPSNet evaluated on Cityscapes-VPS [34]. Scores obtained from official code and models.

Cityscapes-VPS (val)	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	All things	All
VPQ ($K = 4$)	0.32	0.35	0.45	0.32	0.34	0.43	0.27	0.28	0.34	0.52
STQ	0.51	0.48	0.63	0.46	0.38	0.31	0.26	0.44	0.50	0.50
AQ	0.30	0.30	0.42	0.29	0.20	0.18	0.18	0.27	0.35	0.35

Table 9: Per-class breakdown of thing scores of VPSNet evaluated on the validation set of Cityscapes-VPS [34]. Scores obtained from official code and models. VPQ with $K = 4$ annotated frames, is the longest and hardest evaluation of VPQ under the official settings.

therefore adapt to the given characteristics. Naturally, AQ cannot measure tracking when the dataset focus is on segmentation as discussed above. Hence, AQ will measure pixel-precise image and small clip instance segmentation on Cityscapes-VPS. A lower AQ on Cityscapes-VPS does therefore not imply that tracking is harder than on KITTI-STEP. Moreover, all instances contribute equally towards the final score independent of the instance size. With almost 50% instances not being tracks, 50% of the score will be image instance segmentation.

As a reminder, VPQ is averaged over the scores when evaluating PQ on one, two, three, and four frames. In this example, AQ behaves similar as VPQK when evaluated on $K = 4$ frames.