# Scene-Centric Unsupervised Video Panoptic Segmentation

## Supplementary Material

Christoph Reich[*] [1,2,5,6] Oliver Hahn[*] [2,3] Nikita Araslanov [1,5] Laura Leal-Taixé [3]
Christian Rupprecht [4] Daniel Cremers[†] [1,5,6] Stefan Roth[†] [2,6,7]

[1]TU Munich [2]TU Darmstadt [3]NVIDIA [4]University of Oxford [5]MCML [6]ELIZA [7]hessian.AI [*]equal contribution [†]equal advising

https://visinf.github.io/videocups

In this supplement, we first provide additional implementation details, including dataset information, to aid reproducibility (Sec. A). We discuss our evaluation protocol and choice of metric in Section B. Next, we provide additional quantitative and qualitative results and analysis (Sec. C). Finally, we provide a comprehensive discussion on the limitations of VideoCUPS and outline potential future research directions (Sec. D).

## A. Reproducibility

This section provides further information about the datasets used and details on our implementation to ensure reproducibility. To further ensure reproducibility and provide a foundation for future work on unsupervised VPS, our code is available at https://github.com/visinf/cups/tree/main/videocups.

### A.1. Datasets

Here, we provide additional details on the datasets utilized for training and evaluation.

**Cityscapes [23]** is a dataset of urban driving scenes composed of 5 000 high-resolution images at $1\,024 \times 2\,048$ pixels. The dataset is split into 2 975 training, 500 validation, and 1 525 test images, each annotated at the pixel level with panoptic labels. While Cityscapes provides different levels of semantic annotations (27, 19, and 7 semantic categories), the Cityscapes evaluation protocol employs 19 categories for evaluation. These 19 categories are composed of 8 "thing" and 11 "stuff" categories. Every annotated training image is extracted from a 30-frame video clip. Following prior work [41], we utilize these 2 975 training clips (86 275 video frames) for generating pseudo-labels and training.

**Cityscapes-VPS [57]** extends the Cityscapes dataset with panoptic video annotations. In particular, Cityscapes-VPS offers VPS annotations of the 500 Cityscapes validation clips. Cityscapes-VPS provides annotations for every fifth frame of each 30-frame clip using 19 categories matching Cityscapes. The annotations are obtained using a semi-automated annotation process with human correction. Still, multiple works pointed out labeling errors [117, 124]. The 500 Cityscapes-VPS dataset provides a split into 400 training, 50 validation, and 50 test clips. We perform evaluation on the 50 validation video sequences, following the originally proposed setting.

**KITTI-STEP [114]** provides panoptic video annotations for the KITTI-MOTS dataset [107, 128] and comprises 12 training, 9 validation, and 29 test videos. While Cityscapes-VPS provides more video clips and "thing" detections per frame, KITTI-STEP provides significantly longer video clips, on average 381 annotated frames per sequence, and longer tracks (average track length 51 frames). This is significantly longer than the 6 annotated frames of each Cityscapes-VPS clip. The semantic taxonomy of KITTI-STEP matches the 19-class taxonomy of Cityscapes, however, provides fewer "thing" classes—only persons and cars are annotated instance-wise. To compensate for this during evaluation, we perform semantic matching using the "thing" and "stuff" separation of Cityscapes and ignore video instance predictions of semantic classes different than "person" and "car". For evaluation, we use the validation split.

**Waymo [77, 98]** comprises panoramic video panoptic annotations for 2 860 clips, covering a broad range of street scenes under diverse conditions (*e.g.*, night, rain, *etc.*). The dataset provides five camera views and is split into 2 002 training, 286 validation, and 572 test clips. We use the forward-facing view with a resolution of $1\,080 \times 1\,920$ pixels. Following established practice [41], we map Waymo's semantic labels to the Cityscapes taxonomy, resulting in 16 categories, and report results on the validation split. As Waymo includes a substantial number of very small "thing" detections, we remove instances with an average track size below 400 pixels. This mitigates the impact of extremely fine-grained annotations, which current unsupervised approaches cannot segment reliably, and ensures comparability to Cityscapes-VPS.

**MOTS [107]** is used to assess scenes-centric VPS accuracy on videos different from autonomous driving scenarios. We utilize the four MOTChallenge training video sequences, each composed of 2 866 frames, for evaluation. These frames entail a resolution of $1\,080 \times 1\,920$ or $480 \times 640$ pixels. MOTS provides two annotated categories "person" and "background", while providing video instance segmen-

tation for each "person" instance. We consider "person" as a "thing" category and "background" as a "stuff" category.

### A.2. Implementation details

We implement VideoCUPS using PyTorch [131] and train using PyTorch Lightning [127]. We utilized Detectron2 [138] for implementing Panoptic Cascade MaskTrack R-CNN and Kornia [133] for augmentations. Our implementation is partly built upon the code from previous work [41, 42, 91].

**Pre-trained models.** Our full pipeline utilizes SMURF [102], Dynamo-Depth [99], and DepthG [91]. To ensure full compliance with our purely unsupervised and monocular setup, we retrain Dynamo-Depth and DepthG. While SMURF has already been trained using monocular, unlabeled videos, DepthG uses a supervised depth model, and Dynamo-Depth initializes training with an ImageNet [85] supervised backbone. In particular, we retrain Dynamo-Depth [99] with a DINO ResNet-18 [13, 46], instead of an ImageNet-supervised [85] ResNet-18. DepthG is retrained with the monocular depth estimates of our retrained Dynamo-Depth model.

**Pseudo-label generation.** We generate panoptic video pseudo-labels using $c_{\text{p}} = 27$ pseudo-classes on the Cityscapes training sequences, following CUPS [41] and use a thing-stuff threshold of $\psi^{\text{ts}} = 0.01$ (*cf*. Sec. C.1). Semantic pseudo-labeling uses the prediction of our retrained DepthG, and depth-guided semantic inference follows the same setting as proposed in [41]. We post-process pseudo-labels with a CRF [64] using regularized Frank-Wolfe inference [66] and use the original hyperparameters. SMURF and our retrained Dynamo-Depth are used for pseudo-labeling. Our region growing uses $\alpha = 0.15$, $\tau_d = 0.02$, $\tau_f = 0.04$, and $r = 8$ (*cf*. Tab. 8). Instance propagation and tracking uses a sliding window of length three and an IoU-threshold of $\tau_m = 0.4$ in Hungarian matching (*cf*. Tab. 7). Temporal semantic smoothing likewise uses a three-frame sliding window.

**Training and evaluation.** To ensure fairness to our baselines U2Seg [81] and CUPS [41], which employ a Panoptic Cascade Mask R-CNN [10, 58], we use the closest video extension, the Panoptic Cascade MaskTrack R-CNN [10, 58, 121]. Following CutLER [109], U2Seg [81], and CUPS [41], we utilize a ResNet-50 [46] backbone with DINO [13] initialization, pre-trained self-supervised for two epochs on ImageNet [85]. Building on CUPS [41], we train using AdamW [72] with a base learning rate of $2 \times 10^{-5}$, our self-enhanced video copy-paste augmentation, and our Video DropLoss (with $\tau_{\text{IoU}} = 0.5$) for eight epochs. Our self-enhanced video copy-paste augmentation starts using model predictions after one epoch. During the first epoch, we copy-paste pseudo-labels. We paste between one and eight "thing" video detections into another video clip. Training was performed on four NVIDIA A100 GPUs (80 GB) using a batch size of 24. Evaluation of VideoCUPS and our unsupervised baselines is performed using the native resolution of each dataset (*e.g.*, $1\,024 \times 2\,048$ for Cityscapes [23]).

## B. Unsupervised VPS Evaluation Protocol

As we train in a fully unsupervised fashion, our model only predicts pseudo-classes. These need to be mapped to the ground-truth categories. For this, we presented a simple and hyperparameter-free approach in Sec. 3.3. After mapping pseudo-categories to ground-truth categories, we utilize the established Segmentation and Tracking Quality (STQ) [114]. In the following, we provide details on the STQ and discuss other VPS metrics.

### B.1. Segmentation and Tracking Quality

After mapping pseudo-categories to ground-truth categories (*cf*. Sec. 3.3), we are equipped with the VPS predictions per clip $\mathbf{P}_i = (\check{\mathbf{S}}_i, \mathbf{R}_i)$, where $\check{\mathbf{S}}_i \in \{1, 2, \dots, c_{\text{gt}}\}^{\text{T}\times\text{H}\times\text{W}}$ denotes the matched semantic predictions obtained using the pseudo-semantics $\mathbf{S}$; $i$ is the clip index, and $\mathbf{R}_i \in \{0,1\}^{n_p\times\text{T}\times\text{H}\times\text{W}}$ indicates the per-frame presence of $n_p$ predicted "thing" video instances. For evaluation, we use the ground-truth VPS labels $\bar{\mathbf{P}}_i = (\bar{\mathbf{S}}_i, \bar{\mathbf{R}}_i)$, with the semantic ground truth $\bar{\mathbf{S}}_i \in \{1, 2, \dots, c_{\text{gt}}\}^{\text{T}\times\text{H}\times\text{W}}$ and the corresponding $n_{\text{gt}}$ binary video instance masks $\bar{\mathbf{R}}_i \in \{0,1\}^{n_{\text{gt}}\times\text{T}\times\text{H}\times\text{W}}$. The Segmentation and Tracking Quality is computed as

$$\text{STQ} = (\text{AQ} \cdot \text{SQ})^{\frac{1}{2}}, \tag{4}$$

where AQ is the Association Quality, and SQ is the Segmentation Quality. Specifically, the AQ measures how accurately instances were detected and tracked over time, while SQ measures how well pixel semantics were predicted, effectively decoupling segmentation and association.

More specifically, the Segmentation Quality, SQ, is defined as the mean Intersection over Union over the ground-truth classes $c \in \{1, 2, \dots, c_{\text{gt}}\}$ computed as

$$\text{SQ} = \frac{1}{c_{\text{gt}}} \sum_{c\in\{1,\dots,c_{\text{gt}}\}} \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \tag{5}$$

with

$$\text{TP}_c = \sum_{i,t,h,w} \mathbb{1}\big[\check{\mathbf{S}}_{i,t,h,w} = c\big]\ \mathbb{1}\big[\bar{\mathbf{S}}_{i,t,h,w} = c\big], \tag{6}$$

$$\text{FP}_c = \sum_{i,t,h,w} \mathbb{1}\big[\check{\mathbf{S}}_{i,t,h,w} = c\big]\ \mathbb{1}\big[\bar{\mathbf{S}}_{i,t,h,w} \neq c\big], \tag{7}$$

$$\text{FN}_c = \sum_{i,t,h,w} \mathbb{1}\big[\check{\mathbf{S}}_{i,t,h,w} \neq c\big]\ \mathbb{1}\big[\bar{\mathbf{S}}_{i,t,h,w} = c\big], \tag{8}$$

computed over the temporal and spatial dimensions as well as all evaluated clips.

The Association Quality, AQ, is computed using the "thing" video detections $\mathbf{R}_i$ and the ground truth $\bar{\mathbf{R}}_i$. First, the true positive $\mathrm{TPA}_i(g,p)$, false positive $\mathrm{FPA}_i(g,p)$, and false negative $\mathrm{FNA}_i(g,p)$ association areas for the predicted video instances $g \in \{1,2,\dots,n_p\}$ and the ground-truth video instances $p \in \{1,2,\dots,n_{\mathrm{gt}}\}$ are computed per clip $i$ as

$$\mathrm{TPA}_i(g,p) = \sum_{t,h,w} \bar{\mathbf{R}}_{i,g,t,h,w}\, \mathbf{R}_{i,p,t,h,w}, \tag{9}$$

$$\mathrm{FPA}_i(g,p) = \sum_{t,h,w} \mathbf{R}_{i,p,t,h,w} - \mathrm{TPA}_i(g,p), \tag{10}$$

$$\mathrm{FNA}_i(g,p) = \sum_{t,h,w} \bar{\mathbf{R}}_{i,g,t,h,w} - \mathrm{TPA}_i(g,p). \tag{11}$$

Next, the pairwise association Intersection over Union $\mathrm{IoU}_i^{\mathrm{A}}(g,p)$ is computed using

$$\mathrm{IoU}_i^{\mathrm{A}}(g,p) = \frac{\mathrm{TPA}_i(g,p)}{\mathrm{TPA}_i(g,p) + \mathrm{FPA}_i(g,p) + \mathrm{FNA}_i(g,p)}. \tag{12}$$

Finally, using $\mathrm{TPA}_i(g,p)$ and $\mathrm{IoU}_i^{\mathrm{A}}(g,p)$, the Association Quality is computed as

$$\mathrm{AQ} = \sum_i \frac{\sum_{g=1}^{n_{\mathrm{gt},i}} \frac{1}{|\bar{\mathbf{R}}_{i,g}|} \sum_{p=1}^{n_{p,i}} \mathrm{TPA}_i(g,p)\ \mathrm{IoU}_i^{\mathrm{A}}(g,p)}{n_{\mathrm{gt},i}}, \tag{13}$$

where $|\bar{\mathbf{R}}_{i,g}|$ denotes the total area (*i.e.*, the number of pixels) of the ground-truth video instances $g$, $n_{\mathrm{gt},i}$ the number of ground-truth instances of clip $i$, and $n_{p,i}$ the number of predicted instances of clip $i$.

### B.2. Discussion

Existing work in the supervised domain offers alternative metrics for evaluating VPS, including the Video Panoptic Quality (VPQ) [57] and Panoptic Tracking Quality (PTQ) [51]. However, we adopt STQ as our primary metric in our proposed evaluation protocol for several reasons. *First*, STQ provides a clear separation between segmentation and association quality, yielding more interpretable insights into model behavior. *Second*, STQ evaluates entire videos at the per-pixel level and avoids the need for temporal windowing. In contrast, VPQ relies on fixed window sizes, which introduce sensitivity to the chosen hyperparameter and do not scale to full-length videos [114, 130]. *Third*, STQ does not use a threshold-based matching of "thing" predictions, in contrast to both PTQ and VPQ, making it more robust across object scales and crowded scenes. For these reasons, we refrain from incorporating VPQ into our evaluation protocol, as STQ provides the most stable, interpretable, and hyperparameter-free evaluation. In doing so, we follow KITTI-STEP [114], the 2D Video Panoptic Segmentation Challenge at CVPRW 2023, and the Waymo panoramic VPS dataset [77]. For a more detailed discussion of different VPQ metrics, we refer to Weber *et al*. [114].

Table 5. **VPQ *vs*. STQ.** We compare VideoCUPS to our unsupervised VPS baselines using VPQ and STQ (all in %, ↑) on KITTI-STEP. Both in VPQ and STQ, VideoCUPS outperforms the baselines. † denotes CUPS retrained using monocular videos.

| Method | VPQ | STQ |
|---|---|---|
| DepthG [91] + VideoCutLER [110] | 14.3 | 13.2 |
| U2Seg [81] + SORT [9] | 19.0 | 24.0 |
| CUPS [41] + SORT [9] | 20.4 | 34.2 |
| CUPS† [41] + SORT [9] | 20.0 | 32.9 |
| VideoCUPS *(Ours)* | **21.1** | **37.3** |

Nevertheless, we report VPQ for VideoCUPS and all baselines on KITTI-STEP in Tab. 5 for completeness. VideoCUPS consistently outperforms all proposed baselines in both VPQ and STQ. We observe that the relative accuracy gap between the methods is smaller for VPQ than for STQ. We attribute this to the fundamental differences between the two metrics: VPQ is dominated by per-frame/per-window mask quality, whereas STQ explicitly captures both recognition and temporal consistency. Additionally, STQ does not penalize for the recovery/correcting of tracks [see 114, for more details]. Therefore, STQ better reflects progress in unsupervised VPQ.

## C. Additional Results

Here, we provide additional qualitative and quantitative results extending our experiments in the main paper (*cf*. Sec. 4).

### C.1. Pseudo-label thing-stuff threshold analysis

The thing–stuff threshold $\psi^{\mathrm{ts}}$ introduced in Sec. 3.1 partitions the semantic pseudo-classes into pseudo-*thing* and pseudo-*stuff* classes based on their frequency inside the instance masks across the training data. Table 6 reports the effect of varying $\psi^{\mathrm{ts}}$ on pseudo-labels generated for the Cityscapes-VPS validation split. Very low thresholds assign many semantic pseudo-classes (*e.g.*, 10 for $\psi^{\mathrm{ts}} = 0.0025$) to the "thing" subset, leading to a form of over-clustering of true instance categories. Conversely, high thresholds reduce the number of thing pseudo-classes (*e.g.*, 3 for $\psi^{\mathrm{ts}} = 0.03$), which degrades results, as measured by the STQ. The best results are obtained at $\psi^{\mathrm{ts}} = 0.01$, yielding five pseudo-thing classes and an STQ of 12.1 %.

### C.2. Tracking threshold analysis

In Tab. 7, we analyze the influence of the IoU-threshold $\tau_m$ used for tracking and instance propagation (*cf*. Section 3.1).

Table 6. **Pseudo-label thing-stuff threshold analysis.** We evaluate pseudo-labels generated on Cityscapes-VPS val using different values for the thing-stuff threshold $\psi^{\text{ts}}$ using STQ (in %, ↑).

| $\psi^{\text{ts}}$ → | 0.0025 | 0.005 | 0.01 | 0.02 | 0.03 |
|---|---|---|---|---|---|
| **STQ** | 7.5 | 11.5 | **12.1** | 11.3 | 10.7 |

Table 7. **Video pseudo-label instance propagation and tracking threshold $\tau_m$ analysis**, using different IoU thresholds evaluated on pseudo-labels generated on Cityscapes-VPS val, using STQ, AQ, and SQ (all in %, ↑).

| **Pseudo-label configuration** | **STQ** | **AQ** | **SQ** |
|---|---|---|---|
| $\tau_m = 0.3$ | 11.8 | 4.4 | 32.2 |
| $\tau_m = 0.4$ | **12.1** | **4.5** | **32.3** |
| $\tau_m = 0.5$ | 12.0 | 4.4 | **32.3** |

Table 8. **Instance pseudo-labeling hyperparameter analysis.** We analyse our region growing hyperparameters (instance seed threshold $\alpha$, relative depth threshold $\tau_d$, relative flow threshold $\tau_f$, and neighborhood radius r) on Cityscapes val and report STQ (in %, ↑).

| | ← $\alpha$ → | | | ← $\tau_d$ → | | | ← $\tau_f$ → | | | ← r → | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.05 | 0.15 | 0.25 | 0.01 | 0.02 | 0.04 | 0.01 | 0.04 | 0.07 | 2 | 8 | 14 |
| **STQ** | **12.1** | **12.1** | 11.3 | 11.6 | **12.1** | 12.0 | 12.0 | **12.1** | 12.0 | **12.1** | **12.1** | **12.1** |

In particular, we generate pseudo-labels on the Cityscapes-VPS validation split and evaluate the pseudo-labels following the experimental setup from Tab. 3. Overall, our pseudo-labeling is robust to different $\tau_m$ values. Nonetheless, setting $\tau_m = 0.4$ yields a slightly better STQ than 0.3 and 0.5.

## C.3. Instance pseudo-labeling analysis

In Tab. 8, we provide an analysis of our instance pseudo-labeling hyperparameters. We again generate pseudo-labels on the Cityscapes-VPS validation split and evaluate the pseudo-labels following the experimental setup from Tab. 3. We observe a robust behaviour of our pseudo-labeling w.r.t. the relative motion threshold $\tau_f$ and the neighbourhood radius $r$. Both the instance seed threshold $\alpha$ and the relative depth threshold $\tau_d$ still exhibit a relatively robust behaviour, while less robust than $\tau_f$ and $r$.

## C.4. Dynamic vs. static analysis

Table 9 analyses the accuracy of VideoCUPS and our pseudo-labels on dynamic and static "thing" instances only. In particular, we utilize Cityscapes ground-truth motion masks [134] and compute STQ for moving "thing" instances ($\text{STQ}^{\text{D}}$) and static "thing" instances ($\text{STQ}^{\text{S}}$). When computing $\text{STQ}^{\text{D}}$, we ignore all "stuff" regions and static "thing" instances. Similarly, for $\text{STQ}^{\text{S}}$ we ignore all "stuff" regions and dynamic "thing" instances. While our pseudo-labels only capture dynamic objects, VideoCUPS improves accuracy on *both* static and dynamic objects. These results demonstrate the effectiveness of our Video DropLoss in enabling the network to detect and track objects missed by our pseudo-labels. Note that Cityscapes-VPS contains significantly more and smaller static than dynamic instances [134], resulting in lower $\text{STQ}^{\text{S}}$, which is also observed for the supervised upper bound.

Table 9. **Dynamic *vs*. static analysis.** We report STQ for dynamic ($\text{STQ}^{\text{D}}$) and static "thing" objects ($\text{STQ}^{\text{S}}$) only, ignoring stuff pixels. Both metrics in % (↑) on Cityscapes val.

| **Approach** | **$\text{STQ}^{\text{D}}$** | **$\text{STQ}^{\text{S}}$** |
|---|---|---|
| Supervised | 42.8 | 28.6 |
| Pseudo-labels | 16.8 | 5.9 |
| VideoCUPS | **23.9** | **18.8** |

Table 10. **Pseudo-labeling oracle.** We analyze pseudo-labels generated using supervised depth, flow & motion masks and our unsupervised pseudo-labels on Cityscapes val. using STQ, AQ, and SQ (all in %, ↑).

| **Pseudo-labels** | **STQ** | **AQ** | **SQ** |
|---|---|---|---|
| Supervised | **17.3** | **8.7** | **34.3** |
| Unsupervised *(Ours)* | 12.1 | 4.5 | 32.3 |

Table 11. **SSL features for semantic pseudo-labeling analysis.** We compare our modified version of DepthG [91] using DINO [13] and DINOv3 [93], evaluating semantic image segmentation using mIoU (in %, ↑) for unsupervised clustering and supervised linear probing on Cityscapes val.

| **SSL-Features** | **Unsupervised mIoU** | **Supervised mIoU** |
|---|---|---|
| DINO [13] | 23.2 | 28.6 |
| DINOv3 [93] | 22.0 | 41.0 |

## C.5. Pseudo-labeling oracle

In Tab. 10, we provide an oracle experiment by using supervised cues to generate pseudo labels. In particular, we use supervised depth [139], flow [137], and motion masks [134] for pseudo-labeling. These supervised cues significantly improve pseudo-label accuracy, demonstrating the potential benefit of more accurate unsupervised depth, flow, and motion segmentation to improve unsupervised VPS. Note that this only improves the moving-object masks while still using unsupervised semantics.

## C.6. Analysing SSL features for semantic pseudo-labeling

We analyze the effect of different SSL feature representations on the unsupervised semantic segmentation component used for our pseudo-labeling (*cf*. Sec. 3.1). We experiment using our DepthG [91] variant, adapted to the unsupervised and monocular setting by replacing supervised depth with monocular predictions from Dynamo-

Depth [99]. Both VideoCUPS and the original DepthG employ DINO [13] ViT-Base/8 features. We additionally evaluate DINOv3 [93] ViT-Base/16 features under the standard unsupervised semantic image segmentation protocol [20, 40, 42, 56, 88, 91] and report the mean Intersection over Union in Tab. 11.

Despite stronger segmentation results from supervised linear probing, DINOv3 yields inferior unsupervised segmentation mIoU compared to DINO (*cf*. Tab. 11). This aligns with prior observations for DINO compared to DINOv2 [40]. We attribute the drop to the substantially larger patch sizes in DINOv2/v3, which result in a reduced spatial resolution. While the representations become more discriminative in a supervised setting, their coarse spatial granularity appears detrimental for unsupervised clustering. We use DINO(v1) features in our experiments to ensure a fair comparison with U2Seg [81] and CUPS [41].

### C.7. Class-level analysis

Table 12 provides class-wise Segmentation Quality results of VideoCUPS and our baselines. Note that SQ only measures the segmentation accuracy, not detection and tracking accuracy. We observe that rare classes are still a significant challenge for *all* unsupervised approaches. For example, the predictions of VideoCUPS only capture five out of the eight "thing" classes. CUPS + SORT and CUPS$^\dagger$ + SORT only capture four "thing" classes. Notably, while scoring a significantly lower overall SQ and aligning not well with ground-truth instances (*cf*. Tab. 1), U2Seg + SORT predicts all "thing" and "stuff" classes, most likely due to the significant overclustering with 827 pseudo-classes. On average, VideoCUPS outperforms our proposed baselines. In comparison to the supervised upper bound, missed classes account for most of the accuracy gap between supervised and unsupervised approaches, including VideoCUPS. For frequent classes (*e.g.*, "Road", "Sky", or "Car"), our unsupervised VPS almost matches the results of the supervised upper bound. Fine-tuning on just a few VPS annotations (10 % of Cityscapes-VPS train) can adapt VideoCUPS to predict all semantic classes.

### C.8. Qualitative results

In addition to the Cityscapes-VPS qualitative results in Sec. 4.1, we present further visual comparisons. We compare VideoCUPS to the proposed baselines DepthG [91] + VideoCutLER [110], U2Seg [81] + SORT [9], CUPS [41] + SORT, and CUPS$^\dagger$ [41] + SORT, across KITTI-STEP [114], Waymo [77, 98], and MOTS [107]. We also include qualitative out-of-domain (OOD) results on DAVIS [132]. Further, we include a video comparing all methods on the Cityscapes demo video sequences in the supplementary material. Importantly, we apply no post-processing to avoid confounding the evaluation. We deliberately do not filter small masks or discard short-lived instance tracks, as this would introduce additional inference-time hyperparameters.

Figure 7 presents a qualitative comparison on KITTI-STEP. DepthG + VideoCutLER detects only a limited set of instances. U2Seg + SORT increases the number of predicted instances but frequently produces artifact-like instance predictions (*e.g.*, erroneous arrow on the road; top example). CUPS + SORT yields a large number of instances with stable temporal identities, while its monocular variant, CUPS$^\dagger$ + SORT, misses several smaller background instances. In contrast, VideoCUPS consistently discovers both near and far objects, producing accurate masks and temporally robust tracks.

Figure 8 compares all methods on the Waymo dataset. DepthG + VideoCutLER captures only prominent foreground objects and frequently merges distant instances into single masks (*e.g.*, car 1, left). U2Seg + SORT predicts good semantics but continues to merge multiple objects and exhibits noticeable artifacts in the instance predictions. CUPS + SORT achieves strong semantic segmentation results and recovers many instances with stable temporal identities. Under the pronounced domain shift and in cluttered scenes, both CUPS + SORT and VideoCUPS occasionally predict small false instance predictions (*e.g.*, lamppost; right example). The monocular variant, CUPS$^\dagger$ + SORT, detects fewer objects and generates coarser instance masks (*e.g.*, person 7, right). In contrast, VideoCUPS provides accurate semantics and numerous precise instance masks (*e.g.*, person 6 and car 4, right) with consistent tracking across the entire sequence.

A qualitative assessment on the OOD dataset, MOTS, is provided in Fig. 9. DepthG + VideoCutLER recovers many of the foreground pedestrians but frequently merges multiple individuals into a single instance mask. U2Seg + SORT predicts pedestrian instances reliably, yet suffers from artifacts (*e.g.*, store signs in the right example). CUPS + SORT outputs precise instance masks but occasionally fails to maintain tracks (*e.g.*, person 2; left example). The monocular variant, CUPS$^\dagger$ + SORT, yields coarser masks and more artifacts overall. Overall, VideoCUPS delivers the strongest qualitative results among all evaluated methods: it provides accurate instance masks with stable temporal associations, struggling only with very small distant objects.

We further assess the generalization ability of VideoCUPS. Figure 10 shows qualitative results on the DAVIS [132] dataset, using the class assignments learned on Cityscapes-VPS to map pseudo-classes to ground-truth categories for visualization. VideoCUPS generalizes well to this unseen domain and correctly handles unseen semantic concepts, such as forest and mountains (top examples). In addition, VideoCUPS produces accurate instance masks with consistent tracking over time.

Table 12. **Class-level results on Cityscapes-VPS val.** We compare VideoCUPS to the unsupervised VPS baselines, using the class-wise segmentation quality (SQ, in %, ↑). † denotes CUPS retrained using monocular videos. * denotes "thing" classes with spatio-temporal instance annotations. For reference, we also report the class-wise scores of VideoCUPS fine-tuned with 10 % of the Cityscapes-VPS annotations.

| Method | Road | Sidewalk | Building | Wall | Fence | Pole | Traffic Light | Traffic Sign | Vegetation | Terrain | Sky | Person* | Rider* | Car* | Truck* | Bus* | Train* | Motorcycle* | Bicycle* | Mean (SQ) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Supervised [58] | 88.7 | 73.0 | 83.1 | 47.9 | 57.6 | 51.1 | 47.9 | 63.8 | 82.4 | 58.2 | 85.7 | 72.0 | 58.7 | 84.8 | 65.7 | 82.5 | 26.2 | 47.2 | 64.9 | 65.3 |
| DepthG [91] + VideoCutLER [110] | 85.4 | 17.8 | 67.5 | 1.3 | 9.5 | 8.5 | 7.2 | 24.9 | 81.3 | 25.0 | 78.0 | 53.6 | 0.0 | 74.6 | 2.0 | 0.0 | 0.0 | 0.0 | 0.0 | 28.2 |
| U2Seg [81] + SORT [9] | 77.4 | 0.4 | 50.6 | 7.2 | 1.2 | 1.0 | 0.4 | 0.1 | 77.7 | 22.5 | 74.9 | 15.1 | 2.9 | 48.0 | 2.7 | 32.8 | 0.5 | 4.5 | 17.8 | 23.0 |
| CUPS [41] + SORT [9] | 81.1 | 16.2 | 63.2 | 1.3 | 4.9 | 30.6 | 17.6 | 40.7 | 80.9 | 36.3 | 81.9 | 47.9 | 0.0 | 61.1 | 0.0 | 23.3 | 0.0 | 0.0 | 18.2 | 31.8 |
| CUPS† [41] + SORT [9] | 81.3 | 13.2 | 62.2 | 1.5 | 5.0 | 30.3 | 19.7 | 38.1 | 81.1 | 30.1 | 81.9 | 45.4 | 0.0 | 62.8 | 0.0 | 12.4 | 0.0 | 0.0 | 2.3 | 29.9 |
| VideoCUPS *(Ours)* | 81.4 | 17.7 | 67.9 | 1.0 | 9.9 | 27.0 | 8.8 | 39.1 | 80.9 | 32.5 | 83.4 | 53.1 | 0.0 | 67.8 | 0.0 | 10.1 | 0.0 | 0.6 | 31.5 | **32.3** |
| VideoCUPS w/ 10 % VPS ann. *(Ours)* | 85.6 | 43.9 | 83.0 | 16.8 | 35.7 | 37.4 | 37.5 | 52.7 | 83.0 | 19.5 | 88.2 | 70.0 | 44.8 | 82.9 | 24.8 | 5.1 | 21.6 | 18.2 | 56.5 | 47.7 |


$t_0$ $t_1$

Figure 6. **VideoCUPS partial occlusion example** on KITTI-STEP. While VideoCUPS struggles by design with full occlusion, VideoCUPS is still able to track objects through some partial occlusions (*cf*. object 154). Zoom in for details.

## D. Limitations and Future Work

**Moving objects assumption.** We show that unsupervised VPS is feasible by combining self-supervised visual representations with motion and depth cues. A current limitation is the requirement for independently moving objects to obtain initial video instance pseudo-labels. Although this assumption holds in many real-world scenarios, predominantly static objects, *e.g*., a painting mounted on a wall, remain challenging to segment in the unsupervised setting. MaskCut-based approaches such as U2Seg or VideoCutLER can, in principle, discover such objects, but they require object-centric imagery and exhibit poor results on scene-centric data. Integrating motion-based segmentation with MaskCut-based pseudo-labeling may enable the segmentation of predominantly static objects while still scaling to scene-centric videos.

**Dependency on driving scenes.** While Most of our results are reported on driving scenes, VideoCUPS can be applied to non-driving-specific scenarios. Our approach only requires an agent moving through space, and target instances are movable, a common setting in robotics. Still, we require accurate unsupervised depth and motion, as well as VPS annotations for evaluation, which are mostly available for driving scenes (*e.g*., KITTI-STEP [114]). We show domain generalization of VideoCUPS beyond driving scenes on MOTS (*cf*. Tab. 2, Fig. 9) and DAVIS (*cf*. Fig. 10).

**Occlusions.** Motion segmentation, used for pseudo-labeling, can only detect non-occluded objects. Additionally, partial occlusions, *e.g*., a car behind a pole, can lead to two detections of the same object cut by the partial occlusion. Subsequently, VideoCUPS, trained using these pseudo-labels, struggles to detect partially occluded objects correctly and fails to track temporarily fully occluded objects. Still, though our self-enhanced video copy-paste augmentation VideoCUPS can handle some degree of partial occlusions (*cf*. Fig. 6). Enhancing the pasting strategy of video copy-paste augmentations by systematically introducing partial and full occlusions, as well as by extending training to longer clips, might offer potential avenues to mitigate this limitation.

**Unsupervised semantic taxonomy.** Unsupervised segmentation approaches, including VideoCUPS, learn a segmentation taxonomy from unsupervised cues and imposed hyperparameters. While we demonstrate that VideoCUPS learns a taxonomy that significantly correlates with human-defined taxonomies, ideally, unsupervised approaches would learn a flexible, hierarchical taxonomy capable of expressing and discovering novel semantic categories. Creating more flexible approaches and benchmarks that treat unsupervised VPS as an open-vocabulary task would provide a path to overcoming this limitation. Additionally, unsupervised taxonomies require matching to a ground-truth taxonomy for validation. While most likely not suitable for evaluation, exploring ground-truth-free alignment between taxonomies could provide a powerful way to adapt unsupervised segmentation models to new taxonomies and to analyze the structure of the learned taxonomy [129, 135, 136].

**Scaling to multiple datasets.** Existing scene-centric unsupervised panoptic methods, such as CUPS, rely on stereo video during training. In contrast, VideoCUPS's pseudo-


Figure 7. **KITTI-STEP—Qualitative unsupervised VPS examples.** We compare our proposed method VideoCUPS to the proposed baselines DepthG [91] + VideoCutLER [110], U2Seg [81] + SORT, CUPS [41] + SORT [9], and CUPS† [41] + SORT [9] on KITTI-STEP [114] val.

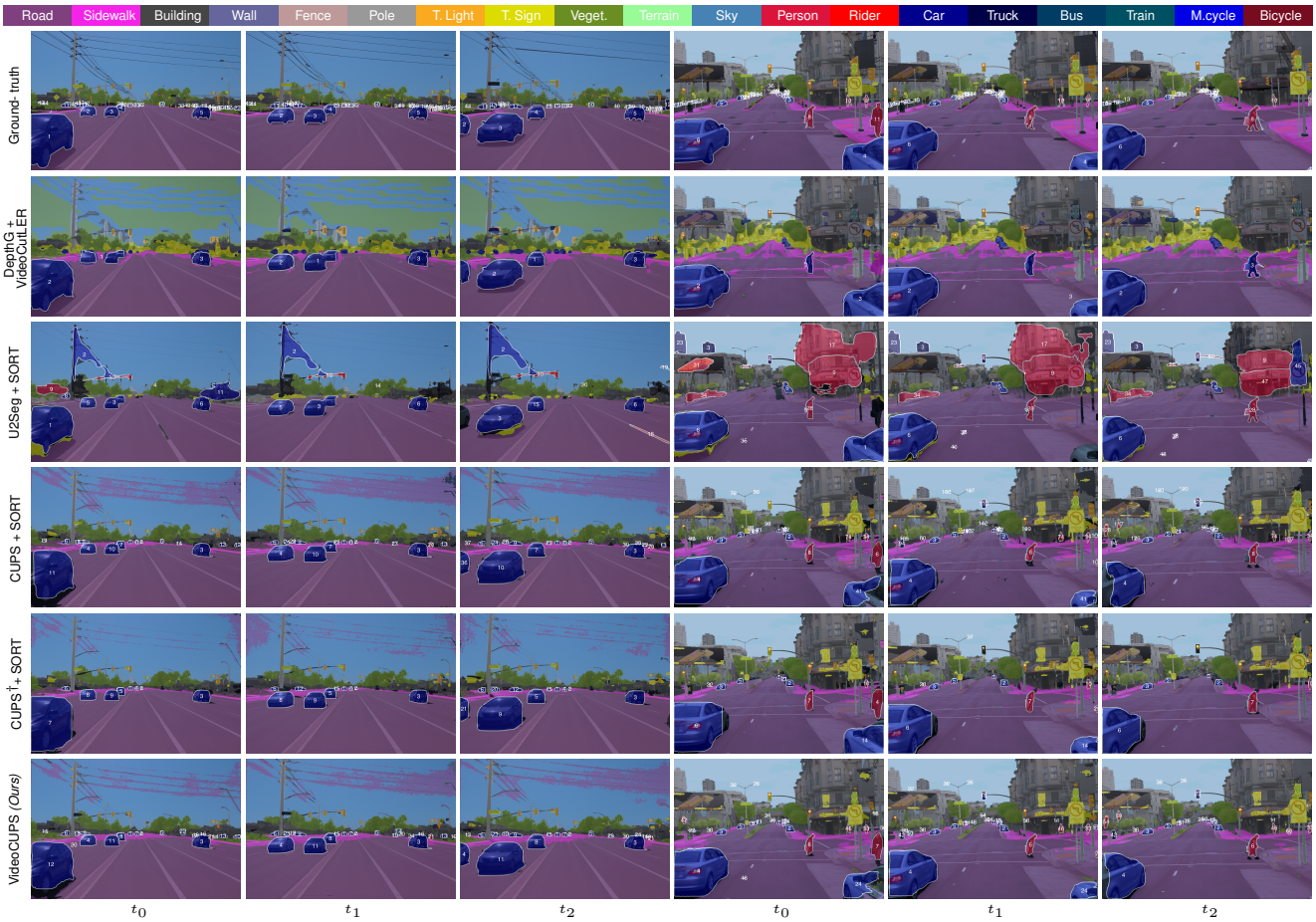


Figure 8. **Waymo—Qualitative unsupervised VPS examples.** We compare our proposed method VideoCUPS to the proposed baselines DepthG [91] + VideoCutLER [110], U2Seg [81] + SORT, CUPS [41] + SORT [9], and CUPS† [41] + SORT [9] on Waymo [77, 98] val.

labeling uses monocular videos. This provides an initial step toward scaling unsupervised panoptic video understanding to larger video datasets. Still, achieving true scalability to causal and monocular videos requires progress in two domains. *First*, while stereo depth estimation is robust and generalizes well, unsupervised monocular depth estimation is still limited. Current models, including DynamoDepth [99], are typically trained on a single dataset and typically do not generalize well to different cameras and other datasets/domains. Robust and generalizable unsupervised monocular depth estimation, including a static and dynamic scene decomposition, would enable more high-quality pseudo-labels, enabling scaling VideoCUPS. *Second*, unsupervised semantic segmentation approaches must produce consistent pseudo semantics across diverse datasets and a large set of pseudo-categories. Current DINO-based unsupervised semantic segmentation approaches, including DepthG [91], typically train a segmentation head for a specific dataset. The resulting pseudo-categories do not necessarily align with pseudo-categories obtained when training on another dataset. Additionally, diffusion-based unsupervised semantic segmentation approaches, such as DiffCut [24], use language supervision and often provide pseudo-categories only consistent within a single image, requiring per-image matching for validation. Obtaining an approach that can express a globally consistent, hierarchical, and large-scale taxonomy of pseudo-categories is needed for scaling unsupervised VPS and unsupervised scene understanding in general.

## References


[127] William A. Falcon and The PyTorch Lightning team. PyTorch Lightning. https://github.com/Lightning-AI/pytorch-lightning, 2019. ii

[128] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. i

[129] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The Platonic representation hypothesis. In *ICML*, pages 20617–20642, 2024. vi

[130] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic

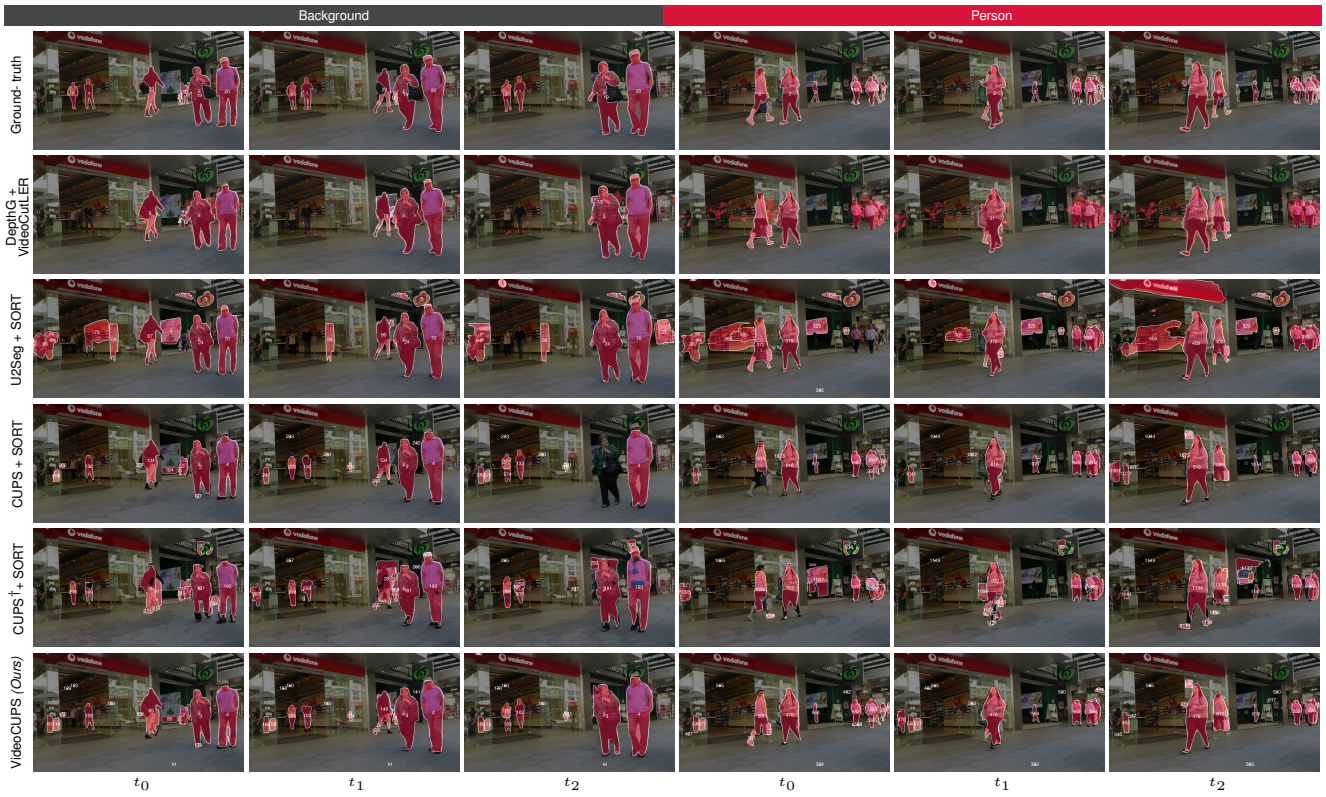


Figure 9. **MOTS—Qualitative unsupervised VPS examples.** We compare our proposed method VideoCUPS to the proposed baselines DepthG [91] + VideoCutLER [110], U2Seg [81] + SORT, CUPS [41] + SORT [9], and CUPS† [41] + SORT [9] on MOTS [107] val.

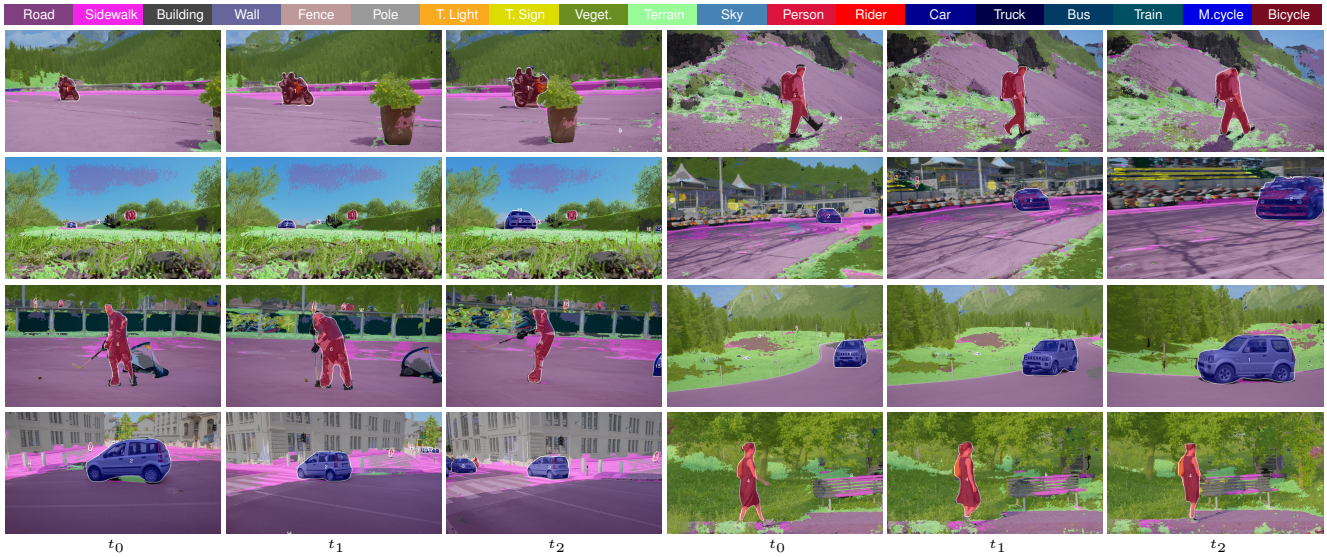


Figure 10. **Qualitative unsupervised VPS examples on DAVIS** [132]. We provide qualitative samples for VideoCUPS inference on DAVIS videos using the Cityscapes-VPS class assignments for visualization purposes. VideoCUPS generalizes to the unseen dataset and even to unseen semantic concepts.

segmentation in the wild: A benchmark. In *CVPR*, pages 21033–21043, 2022. iii

[131] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. ii

[132] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016 v, ix

[133] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Bradski Ethan, and Gary Bradski. Kornia: An open source differentiable computer vision library for PyTorch. In *WACV*, pages 8024–8035, 2020. ii

[134] Mennatullah Siam, Alex Kendall, and Martin Jagersand. Video class agnostic segmentation benchmark for autonomous driving. In *CVPR*, pages 2825–2834, 2021. iv

[135] Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the universal geometry of embeddings. In *NeurIPS*, 2025. vi

[136] Dominik Schnaus, Nikita Araslanov, and Daniel Cremers. It's a (blind) match! Towards vision-language correspondence without parallel data. In *CVPR*, pages 24983–24992, 2025. vi

[137] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, efficient, accurate RAFT for optical flow. In *ECCV*, pages 36–54, 2024. iv

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

[139] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *NeurIPS*, pages 21875–21911, 2024. iv