Scene-Centric Unsupervised Panoptic Segmentation

Supplementary Material

Oliver Hahn[*1] Christoph Reich[*1,2,4,5] Nikita Araslanov[2,4]
Daniel Cremers[2,4,5] Christian Rupprecht[3] Stefan Roth[1,5,6]
[1]TU Darmstadt [2]TU Munich [3]University of Oxford [4]MCML [5]ELIZA [6]hessian.AI [*]equal contribution
https://visinf.github.io/cups

In this appendix, we first highlight the conceptual features of our unsupervised panoptic segmentation method CUPS. We elaborate on the training and validation approach as well as on implementation details to facilitate reproducibility. Next, we conduct further analyses of different design choices and the training stages. We then provide additional quantitative and qualitative results. Finally, we discuss the limitations of current unsupervised panoptic approaches as well as our CUPS approach.

A. CUPS *vs.* U2Seg: A Conceptual Comparison

Table 8 conceptually compares CUPS to U2Seg [55]. While both frameworks address the problem of unsupervised panoptic segmentation, CUPS features novel distinctions:
(1) Scene-centric training. Object-centric images typically depict a center-aligned foreground object on a fairly homogeneous background. The photographic bias inherent to this type of imagery also implies the need for manual curation in the data collection process. By contrast, scene-centric data encapsulates the complexity of real-world environments where multiple objects coexist and interact. Furthermore, collecting scene-centric imagery is substantially cheaper, since it obviates the need for artificially isolating objects from their context. Training on scene-centric data is crucial to producing models that are capable of understanding real-world complexity and serving the needs of challenging applications, such as autonomous driving, robotic navigation, augmented reality, and assistive technologies for visually impaired individuals. Although we are not the first to leverage motion for retrieving instance cues, accomplishing this in a self-supervised fashion is a novel aspect in the context of unsupervised panoptic segmentation.
(2) High-resolution pseudo labels. High-resolution training is crucial for capturing fine details in scene-centric data, which lower-resolution settings cannot address. Our depth-guided semantic inference (*cf*. Sec. 3.1) provides a semantic pseudo-labeling component with twice the resolution of previous methods. This enhancement allows CUPS to learn semantic cues to a higher degree of detail, which can be observed in our qualitative results (*cf*. Fig. 7).
(3) Thing-stuff separation. Our integration of motion cues enables a precise distinction between semantic pseudo "thing" and pseudo "stuff" classes. This is because motion helps us identify "thing" classes as objects that move relative to the camera. In contrast, U2Seg cannot really distinguish between "stuff" and "thing" classes; this ambiguity is only resolved at test time via oracle matching of the pseudo labels with ground-truth semantic categories and object instances. The capacity of CUPS to discriminate between "stuff" and "thing" categories is an advancement toward solving unsupervised panoptic segmentation in a more principled way.

Table 8. **A conceptual comparison of CUPS and U2Seg.**

	U2Seg [55]	**CUPS** *(Ours)*
Unsupervised panoptic segmentation	✓	✓
Scene-centric training	✗	✓
High-resolution pseudo labels	✗	✓
Thing-stuff separation	∼	✓

B. Reproducibility

To facilitate reproducibility, we elaborate on the technical and implementation details. Note that our code is available at https://github.com/visinf/cups.

B.1. Implementation details

CUPS is implemented using PyTorch [89], PyTorch Lightning [87], and Kornia [90]. We partly build upon public codebases from previous work [30, 55, 62, 65, 66].

Stage 1. CUPS pseudo-label generation uses 27 pseudo classes and a thing-stuff threshold ψ^{ts} of 0.08. This setting enables comparison against existing unsupervised panoptic and semantic segmentation approaches without relying on significant overclustering (*cf*. [55]). Instance pseudo labeling uses motion and depth estimates from a pre-trained SMURF model [66]. For our ensembling-based SF2SE3 clustering, we build upon the original implementation by Sommer *et al*. [65]. Semantic pseudo labeling uses a pre-trained SMURF [66] to generate the depth to train the semantic segmentation network following Sick *et al*. [62]. For depth-guided semantic inference, we first resize the input image so that its smaller side is 320 pixels, matching the standard resolution in unsupervised semantic segmenta-

tion. We then perform a second inference pass using sliding windows on an image scale of 640 pixels with a stride of half the window size. Depth-guided semantic inference uses the SMURF depth estimate to weight the two semantic segmentation predictions. The size of the sliding window is half the image width and height. Finally, we perform post-processing by further aligning the prediction to the image using a fully connected conditional random field (CRF) [43, 92].

For a fair comparison and to demonstrate the impact of our pseudo labeling as well as the proposed training scheme, we use the same panoptic network as U2Seg. In particular, we follow Niu *et al*. [55] by employing the Panoptic Cascade Mask R-CNN [8, 39] with a ResNet-50 [32] backbone pre-trained using self-supervised DINO [12] for two epochs on ImageNet [91].

Stage 2. CUPS pseudo-label bootstrapping proceeds by training for 4 000 steps with AdamW [48], using a learning rate of 10^{-4}, and a weight decay of 10^{-5}. The drop-loss overlap threshold τ^{IoU} is set to 0.4. After 1 000 steps, we start utilizing our self-enhanced copy-paste augmentation, randomly pasting between 1 and 8 objects into each image.

Stage 3. CUPS self-training runs for 1 500 steps using AdamW with a learning rate of 10^{-5} and no weight decay. The EMA decay for updating the momentum network is set to 0.9999. We only update the detection heads, the mask head, and the semantic head, freezing all normalization layers. For self-labeling augmentation, we use three different scales of the original image (0.75, 1.0, and 1.25), as well as horizontal flips at each scale, resulting in six views. We follow Chen *et al*. [14] to set up the photometric augmentation and employ our self-enhanced copy-paste augmentation also during self-training.

For both pseudo-label training and self-training, we utilize four NVIDIA A100 GPUs (40 GB) with a batch size of 16 per GPU. We evaluate CUPS on the native resolution of each dataset, except for unsupervised semantic segmentation (*cf*. Tab. 3) where we follow the common evaluation protocol [19, 30, 35, 62].

B.2. Computing panoptic quality

As we train without any supervision, the semantic pseudo-class IDs are not aligned with the ground-truth semantic class IDs. Therefore, to compute the panoptic quality (PQ) [40], we need to align the pseudo-class IDs with the ground truth, distinguishing between "thing" and "stuff" semantic categories at the same time.

While U2Seg [55] also utilizes the panoptic quality and proposes an elaborate matching approach, significant limitations remain. Niu *et al*. [55] establish a semantic matching using three steps. First, predicted segments are matched with all ground-truth segments, ignoring the "thing" and "stuff" separation. Segments with an overlap of less than a pre-defined threshold (hyperparameter) are discarded. Second, using the set of matched segments and both the semantic pseudo-class IDs and the ground-truth class IDs, a cost matrix is constructed on a per-segment basis. Third, for each semantic pseudo class, the most frequent ground-truth class ID based on the cost matrix is matched. This matching approach entails two significant limitations. First, the overlap threshold is a crucial hyperparameter and can significantly impact the final PQ value. This is mainly due to the fact that the segment-wise cost matrix finds relatively few overlapping objects, and thresholding is required to consider only accurate predictions for matching. Second, the matching approach does not consider the "thing" and "stuff" separation, leading to matches between both "thing" and "stuff" categories. This is highly undesired as "thing" segments entail object-level masks, whereas "stuff" segments only capture the semantic level. Finally, code for evaluation on the Cityscapes dataset has not been published by the authors of [55].

Principles. We redefine the matching process in alignment with the following core principles: *Simplicity:* Introducing additional hyperparameters within the matching is undesirable, as it complicates evaluation. Semantic segmentation is a pixel-wise classification, hence we aim to perform matching of the pseudo classes to ground-truth classes on the pixel level as well. More specifically, every predicted pixel should be considered in the alignment between pseudo classes and ground-truth annotations. This resembles the simplest form of approaching the problem and is common in unsupervised semantic segmentation [19, 30, 35, 62]. *Clear thing and stuff separation:* The distinction between "thing" and "stuff" classes is a core aspect of panoptic segmentation. Consequently, it should be addressed by the method itself rather than the matching process. To ensure alignment, only pseudo classes labeled as "stuff" are matched with "stuff" ground-truth classes, and the same applies to "thing" classes.

Approach. To this end, we propose a simple but effective approach for matching. Taking inspiration from the established semantic matching for the task of unsupervised semantic segmentation [19, 30, 35, 62], we perform matching purely utilizing semantics. In particular, we obtain the semantic segmentation prediction $\bar{\mathbf{P}} \in \{1, \dots, \xi_{\text{p}}\}^{\text{H} \times \text{W}}$ from the unsupervised panoptic prediction, with ξ_{p} denoting the number of pseudo classes. We use the ground-truth semantic segmentation $\hat{\mathbf{P}} \in \{1, \dots, \xi_{\text{GT}}\}^{\text{H} \times \text{W}}$, with ξ_{GT} indicating the number of ground-truth semantic classes, to construct a cost matrix $\mathbf{A} \in \mathbb{N}^{\xi_{\text{p}} \times \xi_{\text{GT}}}$. This cost matrix counts the number of overlapping pixels of each pseudo-class ID with all ground-truth class IDs. The full cost matrix is obtained using all validation samples. To ensure no "thing" class ID is matched to a "stuff" class ID or vice versa, we extract a "thing" and a "stuff" cost matrix from the full cost

matrix $\mathbf{A}$. By using the "thing" and "stuff" splits of classes in the pseudo classes as well as the ground-truth classes, we construct a "thing" cost matrix $\mathbf{A}^{\text{Th}} \in \mathbb{N}^{\xi_{\text{p}}^{\text{Th}} \times \xi_{\text{GT}}^{\text{Th}}}$ and "stuff" cost matrix $\mathbf{A}^{\text{St}} \in \mathbb{N}^{\xi_{\text{p}}^{\text{St}} \times \xi_{\text{GT}}^{\text{St}}}$. Hungarian matching [44] is then applied to maximize overlap and establish a one-to-one matching between pseudo-class IDs and ground-truth class IDs by running matching on $\mathbf{A}^{\text{Th}}$ and $\mathbf{A}^{\text{St}}$, separately. As we can have more semantic pseudo-class IDs than ground-truth class IDs (*i.e.*, $\xi_{\text{p}}^{\text{Th}} > \xi_{\text{GT}}^{\text{Th}}$ and/or $\xi_{\text{p}}^{\text{St}} > \xi_{\text{GT}}^{\text{St}}$), we assign all remaining pseudo classes, not assigned by Hungarian matching, to the respective ground-truth class ID with the maximum overlap. This process leads to a permutation of the pseudo-class IDs, maximizing the overlap with the ground-truth class IDs while adhering to the "thing" and "stuff" separation. Finally, we utilize the permuted (*i.e.*, matched) semantics alongside the instance mask—the binary masks predicted for instances—to compute PQ. For evaluating on the task of unsupervised semantic segmentation, we skip the step of separating $\mathbf{A}$ into $\mathbf{A}^{\text{Th}}$ and $\mathbf{A}^{\text{St}}$ and perform a single matching on $\mathbf{A}$ as done by the related work in the field [19, 30, 35, 62].

To conclude, our class matching for unsupervised panoptic quality builds on established protocols, performs a straightforward and efficient matching, and adheres to the "thing" and "stuff" class split, while not introducing any hyperparameters. Interestingly, we observe that evaluating U2Seg with our matching leads to better panoptic quality than reported in the original paper (*cf*. Tab. 1). This suggests that we find a better correspondence between pseudo and ground-truth classes. We make the evaluation code for all settings publicly available to facilitate future research.

B.3. Datasets

We provide further details about the datasets used to train and evaluate CUPS.

Cityscapes [21] is an ego-centric driving scene dataset, which contains 5 000 high-resolution images with 2048×1024 pixels. It is split into 2 975 train, 500 val, and 1 525 test images with pixel-level annotations provided for grouping into 27, 19, or 7 categories. Each of the training images stems from a short video sequence. We leverage all 86 275 video frames of the training split for unsupervised training and evaluate on the validation split, in line with previous work.

The **KITTI** [26, 53] vision benchmark suite is a comprehensive driving-scene dataset with ground truth for a variety of tasks, such as semantic segmentation, optical flow estimation, depth estimation. Mohan *et al*. [53] introduced the KITTI panoptic segmentation dataset for urban scene understanding by providing panoptic annotations for a subset of 1 055 images. The images have a resolution of 1280×384 pixels and adhere to the 19-class grouping of the Cityscapes taxonomy. We use the 200 validation images for evaluation. Furthermore, we use all 42 150 rectified KITTI images excluding the validation split and calibration scenes for unsupervised training.

Table 9. **Comparison of motion networks for pseudo-label generation.** Investigating the contribution of the correspondence matching network, using PQ, SQ, and RQ (in %, ↑) for pseudo labels generated on Cityscapes val. We use our full configuration and only change the motion network.

Optical flow method	PQ	SQ	RQ
BrightFlow [88] (unsupervised)	17.8	46.4	22.4
SMURF [66] (unsupervised)	18.1	47.3	22.6
SEA-RAFT [93] (supervised)	19.2	51.8	23.4
RAFT [71] (supervised)	20.4	52.6	24.7

BDD [84] is a driving scene dataset, which also contains panoptic annotations with 19 class definitions identical to those in Cityscapes. The images have a resolution of 1280×720 pixels. The validation set contains 1 000 images.

MUSES [7] is a multi-modal dataset representing adverse conditions in driving scenes. The labels use the 19 class taxonomy of Cityscapes. For evaluation, we utilize the "daytime clear" validation split, containing 50 images with a resolution of 1920×1080.

Waymo [68] is a another driving scene dataset. We use the "front" camera, providing a resolution of 1920×1280 pixels and evaluate using the 1 930 images of the 2D panoptic segmentation validation split. Waymo classes are remapped to ensure compatibility of its label space with the Cityscapes classes, resulting in 16 classes.

MOTS [75] allows to assess scene-centric panoptic segmentation outside of driving scenarios. Evaluation is performed using the MOTChallenge sequences for multi-object tracking and segmentation of humans in indoor and outdoor scenes. The annotations include two classes "background" and "person", where "background" is considered as a "stuff" class and "person" is a "thing" class. We evaluate on 2 862 images of resolutions 640×480 or 1920×1080.

C. Additional Results

In the following, we analyze the results presented in the main paper in greater detail.

C.1. CUPS pseudo-labels results

Supervised *vs*. unsupervised optical flow. In conjunction with the pseudo-label generation analysis presented in Tab. 5, we investigate the influence of different approaches for optical flow and two-frame disparity estimation on our pseudo labels in Tab. 9. Identical to the analysis in the main paper, we generate pseudo labels on the validation split to ensure comparability with the CUPS panoptic segmentation results and CUPS analysis.

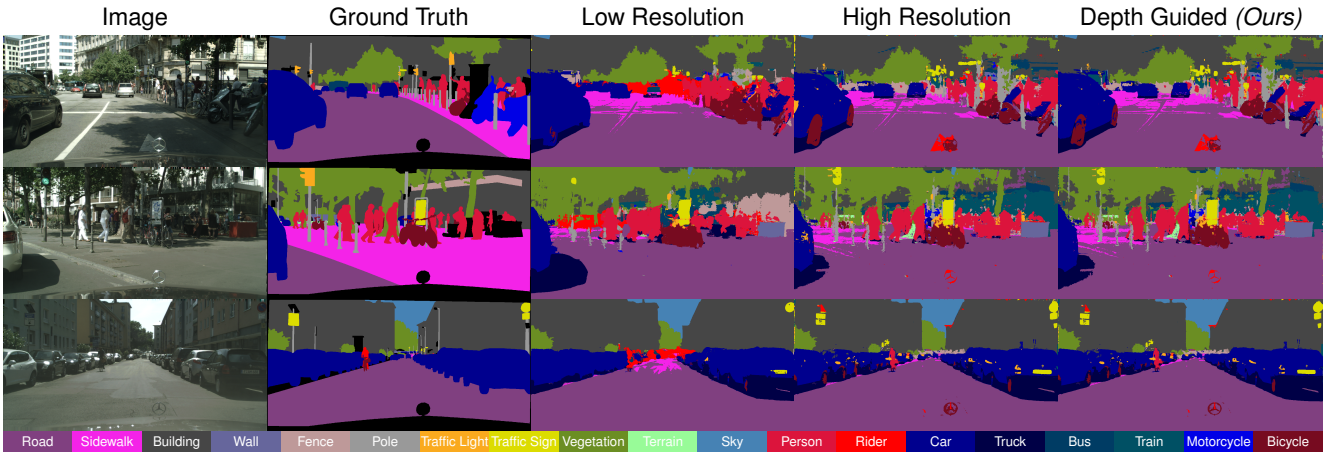


Figure 6. **Depth-guided semantic pseudo-label examples.** Qualitative semantic pseudo-label examples comparing low resolution $\mathbf{P}^{\text{low}}$, high resolution $\mathbf{P}^{\text{high}}$, and depth-guided semantic fusion $\mathbf{P}^{*}$.

Table 10. **Depth-guided semantic pseudo label analysis.** Semantic pseudo labels evaluated on Cityscapes val (for consistency both in 19 class setting). *(a)* comparing the resolutions and merging approaches. *(b)* decomposing depth-guided semantic segmentation accuracy for different depth ranges. All metrics in %.

(a) **Depth-guided semantic pseudo labeling components.**

Method	**PQ**	**SQ**	**RQ**
Low Resolution (P^{low})	15.9	47.0	19.5
High Resolution (P^{high})	17.9	46.8	22.4
Mean	16.7	42.7	20.9
Depth-guided (P^{*})	18.1	47.3	22.6

(b) **Analyzing different depth ranges.**

	mIoU^{19}			
Distance (in m)	**0 – 10**	**10 – 30**	**>30**	**all**
Low Resolution (P^{low})	30.7	28.7	23.3	29.5
High Resolution (P^{high})	28.6	29.6	27.3	30.9
Depth-guided (P^{*})	29.2	29.7	27.3	31.1

Tab. 9 shows the direct quantitative evaluation of pseudo labels generated using different motion estimation methods against the ground truth (*i.e.*, without the panoptic segmentation network). Alongside another unsupervised approach, BrightFlow [88], we include results obtained with supervised methods, RAFT-large [71] (a supervised analog of SMURF [66]) and SEA-RAFT-large [93]. We observe a rather consistent panoptic quality of the pseudo labels across different motion estimation networks. As expected, the more accurate supervised optical flow methods can improve PQ further. The slightly weaker panoptic quality with SEA-RAFT compared to RAFT might be due to SEA-RAFT being fine-tuned on multiple diverse datasets, whereas RAFT is fine-tuned specifically on KITTI. To conclude, CUPS is already effective with unsupervised flow and depth estimation methods, while exhibiting a notable margin for improvement in settings where some supervision of optical flow is available (and permissible).

Table 11. **Instance pseudo label comparison.** Using MaskCut instance masks (U2Seg [55]) in our CUPS pseudo-label generation. We compare using PQ, SQ, and RQ (in %, ↑) for pseudo labels generated on Cityscapes val.

Instance pseudo-label approach	**PQ**	**SQ**	**RQ**
MaskCut [78]	9.9	41.6	12.4
SF2SE3-ensembling *(Ours)*	**18.1**	**47.3**	**22.6**

Analysis of depth-guided semantic pseudo labeling. Following Sec. 3.1, we aim to analyze our proposed depth-guided semantic pseudo labeling in more detail. Table 10 shows that depth guidance fuses low- and high-resolution semantic predictions more effectively than an arithmetic mean. We use the identical experimental setting as in Tab. 5. We further analyze pseudo labels by splitting images into depth ranges. Low resolution is best for pixels closer than 10 m, both predictions perform similarly between 10 – 30 m, and high resolution is superior beyond 30 m. These effects stem from DINO features trained on fixed-resolution, object-centric images, causing reduced representational quality at extreme scales. In short, low-resolution predictions produce blurry outputs for distant fine details, while high-resolution (sliding-window) predictions are more accurate at large distances but introduce errors near the camera. Overall, our depth-guided fusion yields the best metric performance. We show qualitative examples in Fig. 6.

Instance pseudo labeling analysis. Supporting the qualitative results presented in Fig. 2, we further analyze the performance of our SF2SE3-ensembling approach against MaskCut [78]. In particular, Tab. 11 presents pseudo-label evaluation results, replacing our SF2SE3-ensembling with

Table 12. **Per-class unsupervised panoptic segmentation on Cityscapes.** Comparing CUPS to existing unsupervised panoptic methods, using PQ at the class level, as well as the mean PQ (in %, ↑).

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	Mean (PQ)
Supervised [39]	96.5	72.4	85.9	16.4	30.1	48.6	48.8	67.2	86.9	34.8	86.0	65.0	60.8	79.2	58.5	77.2	59.8	54.9	59.3	62.3
DepthG [62] + CutLER [78]	80.9	1.4	55.6	**3.0**	**0.2**	0.4	0.3	0.0	72.9	5.8	61.8	6.0	0.0	17.5	0.0	0.7	0.0	0.0	0.0	16.1
U2Seg [55]	82.5	0.0	42.4	2.0	0.0	0.0	0.0	0.0	76.6	1.5	62.9	8.3	**2.2**	22.3	10.2	**27.0**	**4.7**	**0.7**	6.7	18.4
CUPS *(Ours)*	**85.8**	**6.0**	**64.4**	0.0	**0.2**	**12.4**	**6.2**	**32.1**	**83.7**	**17.1**	**78.2**	**39.1**	0.0	**62.9**	**16.3**	1.2	0.0	0.0	**30.6**	**27.8**

Table 13. **CUPS self-training analysis**. Decomposing the self-training by analyzing the augmentation quality using PQ, SQ, and RQ (in %, ↑) on Cityscapes val.

Training configuration	PQ	SQ	RQ	Runtime (ms)
CUPS w/o self-training	26.6	**57.5**	33.5	65.9
CUPS w/o self-training + TTA	27.4	57.2	34.9	413.4
CUPS *(Ours)*	**27.8**	57.4	**35.2**	65.2

Table 14. **DepthG [62] + VideoCutLER [94] baseline.** We compare CUPS to a baseline using VideoCutLER on the Cityscapes val dataset (all metrics in %, ↑).

Method	PQ	SQ	RQ
DepthG [62] + CutLER [78]	16.1	45.4	21.1
DepthG [62] + VideoCutLER [94]	16.6	42.6	20.5
CUPS *(Ours)*	**27.8**	**57.4**	**35.2**

MaskCut. All other pseudo-label generation components are kept the same. MaskCut fails to generate high-quality instance masks on scene-centric images, as PQ and RQ almost halved compared to our SF2SE3-ensembling.

C.2. Unsupervised panoptic segmentation results

Class-level PQ. Table 12 expands Tab. 1 by detailing class-wise PQ. CUPS demonstrates substantial improvements on most categories, particularly excelling on "Car" (62.9 %), "Person" (39.1 %), "Traffic Sign" (32.1 %), and "Sky" (78.2 %). Although CUPS has difficulties with a few classes, *e.g.*, "Wall", "Fence", and "Rider", our baseline DepthG [62] + CutLER [78] and U2Seg [55] also struggle with segmenting these classes. The only exception is "Bus", on which CUPS exhibits lower PQ than U2Seg. In the case of "Rider", CUPS does not learn this as a separate class, which is probably due to the motion cue used for instance pseudo labeling, which cannot easily separate a "Rider" from their means of transportation. Accordingly, CUPS usually predicts person instead of rider or the entire unit of "Bicycle" and "Rider" is predicted as "Bicycle" (*cf*. Fig. 7a, second example). Nevertheless, CUPS significantly improves the panoptic quality for the majority of classes and narrows the gap to the supervised upper bound.

Panoptic self-training *vs*. test-time augmentation. Following up on the ablation in Tab. 7a, we provide a finer-grained analysis of the self-training process in Tab. 13 by comparing against using the self-labeling augmentations as test-time augmentation (TTA) at inference time directly after Stage 2 instead of the self-training. Recall that the self-labeling augmentations involve resizing the input image to three different scales and applying horizontal flipping, followed by aggregating the predictions. Self-labeling augmentations, combined with confidence thresholding and self-enhanced copy-paste augmentations, provide self-labels for self-training (Stage 3). Note that we report TTA without thresholding in Tab. 13. While the results in Tab. 13 show that TTA improves the panoptic quality, it is not a practical approach due to the significantly increased inference time. By contrast, panoptic self-training retains the original runtime of the network and even surpasses TTA in panoptic quality.

DepthG+VideoCutLER baseline. Since CUPS leverages two consecutive frames to generate instance pseudo labels, it inherently exploits temporal consistency. Consequently, we combine VideoCutLER [94], an unsupervised method for video instance segmentation, with DepthG as an additional baseline. We performed the experiment using five consecutive frames as the video input to VideoCutLER. The semantic and instance predictions of DepthG and VideoCutLER are combined identically to the DepthG+CutLER baseline. As shown in Tab. 14, DepthG+VideoCutLER is slightly worse for SQ and RQ, yet better in PQ. We attribute this to the improved temporal consistency. Our CUPS approach strongly outperforms this video baseline as well.

Overclustering analysis. Overclustering refers to setting the number of pseudo labels significantly higher than the number of ground-truth categories. Extending the analysis presented in Tab. 7b, we analyze the impact of overclustering along two dimensions. First, we test CUPS with an increased number of pseudo classes. Second, we run CUPS in the default setting, but evaluate it on the group-level class hierarchies defined by Cityscapes. Here, the 19-class taxonomy is mapped down to 7 broader groups of classes.

Table 15. **Unsupervised panoptic segmentation for CUPS on Cityscapes, KITTI, BDD, MUSES, Waymo, and MOTS.** Comparing CUPS to existing unsupervised panoptic methods, using PQ, SQ, and RQ (in %, ↑) for different numbers of pseudo classes. By default, CUPS uses 27 pseudo classes to facilitate the comparison against both unsupervised panoptic and unsupervised semantic segmentation approaches. We also test 40 (150 % of the default) and 54 pseudo classes (200 % of the default), showcasing the impact of overclustering.

Method	Pseudo classes	Cityscapes			KITTI			BDD			MUSES			Waymo			MOTS		
		PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ
Supervised [39]	–	62.3	81.8	75.1	31.9	71.7	40.4	33.0	76.3	42.0	38.1	62.4	49.6	31.5	70.1	40.9	73.8	86.4	84.6
DepthG [62] + CutLER [78]	27	16.1	45.4	21.1	11.0	34.5	13.8	14.4	41.9	19.2	10.1	30.1	13.1	13.4	37.3	17.0	49.6	78.4	60.6
U2Seg [55]	800 + 27	18.4	55.8	22.7	20.6	52.9	25.2	15.8	57.2	19.2	20.3	45.8	26.5	19.8	50.8	23.4	50.7	79.2	64.3
CUPS *(Ours)*	27 *(default)*	27.8	57.4	35.2	25.5	58.1	32.5	19.9	60.3	25.9	24.4	48.5	33.0	26.4	60.3	33.0	67.8	86.4	76.9
CUPS *(Ours)*	40	30.3	64.3	37.5	28.1	**63.1**	35.3	**21.9**	57.3	**28.1**	**28.2**	**52.9**	**35.4**	27.2	62.4	**33.6**	74.0	88.4	82.8
CUPS *(Ours)*	54	**30.6**	**65.1**	**37.8**	**28.5**	60.6	**36.0**	21.8	**62.5**	27.6	22.8	45.4	29.3	**27.3**	**65.3**	32.5	**78.7**	**89.3**	**87.4**

Table 16. **Hierarchical unsupervised panoptic segmentation on Cityscapes.** Comparing CUPS to existing unsupervised panoptic methods, using PQ (all in %, ↑) on different class hierarchies. All datasets are analyzed on 19 and 7 ground truth classes. The number of ground-truth classes is indicated by the superscript of the metric.

Method	Pseudo classes	Cityscapes		KITTI		BDD		MUSES		Waymo	
		PQ^{19}	PQ^{7}	PQ^{19}	PQ^{7}	PQ^{19}	PQ^{7}	PQ^{19}	PQ^{7}	PQ^{16}	PQ^{7}
Supervised [39]	–	62.3	79.8	31.9	57.9	33.0	54.6	38.1	69.4	31.5	62.3
DepthG [62] + CutLER [78]	27	16.1	44.1	10.9	27.6	14.4	38.5	10.1	22.1	13.4	37.7
U2Seg [55]	800 + 27	18.4	43.5	20.6	44.4	15.8	37.3	20.3	41.4	19.8	39.6
CUPS *(Ours)*	27	**27.8**	**63.9**	**25.4**	**57.4**	**19.9**	**49.3**	**24.4**	**53.5**	**26.4**	**54.7**

When training CUPS with a larger number of pseudo classes—specifically, 40 (150 % of the default number of pseudo classes)—we observe a significant improvement in the panoptic segmentation metrics (*cf*. Tab. 15). Further increasing the number of pseudo classes to 54 (200 % of the default number of pseudo classes) yields additional improvements but also exhibits a saturation trend. However, significantly increasing the number of pseudo-classes can impede generalization, as visible on MUSES when using 54 pseudo classes. In general, we use 27 pseudo classes for fair comparison, as it is the lowest number of pseudo classes that allows for a comparison to both unsupervised panoptic and unsupervised semantic segmentation.

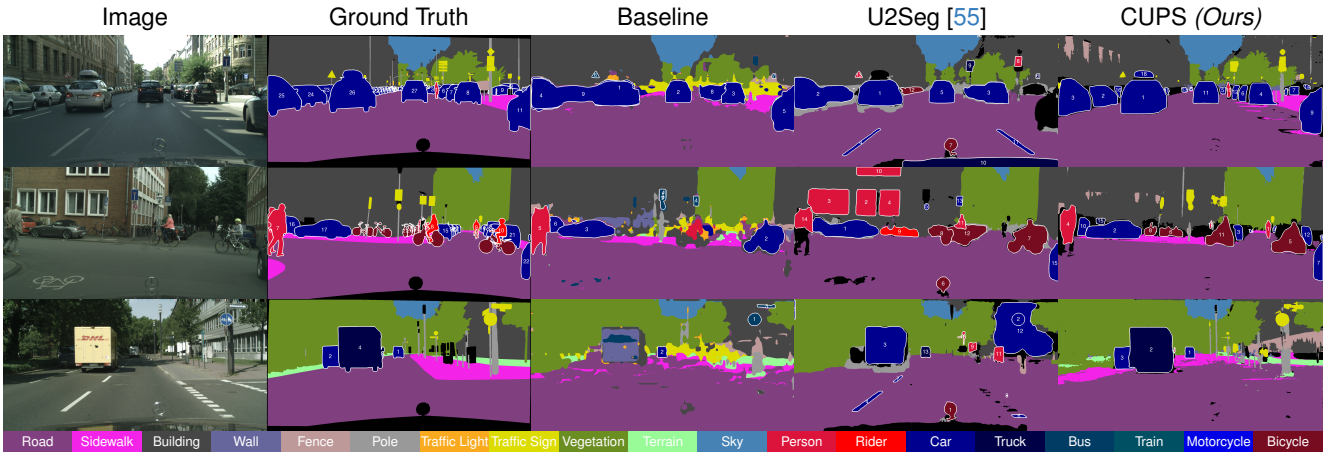
In Tab. 16, we evaluate CUPS on different class hierarchies. While the main paper demonstrates substantial gains in the standard 19-class evaluation, we show that the gains extrapolate to the setting with a coarser grouping of 7 Cityscapes classes: "Flat" (*e.g.*, "Road", "Sidewalk"), "Human" (*e.g.*, "Person", "Rider"), "Vehicle" (*e.g.*, "Car", "Truck"), "Construction" (*e.g.*, "Building", "Wall"), "Object" (*e.g.*, "Pole", "Traffic Sign"), "Nature" (*e.g.*, "Vegetation", "Terrain"), and "Sky". Although the accuracy improvement on the coarser label set is expected, this experiment empirically demonstrates that our analysis and conclusions hold for different granularities of the semantic taxonomy. As another remark, we follow up on our observation from Tab. 2 in the main text, where the supervised model (trained on Cityscapes) suffers a noticeable drop in segmentation performance outside the training domain. In the coarser setting here, this observation applies to a more striking extent: CUPS nearly approaches the supervised bound and achieves competitive panoptic quality with the supervised model (*e.g.*, only 0.3 % worse on KITTI).

Table 17. **Panoptic segmentation architecture analysis.** We evaluate CUPS after stage-1 training on the Cityscapes val datasets (all metrics in %, ↑).

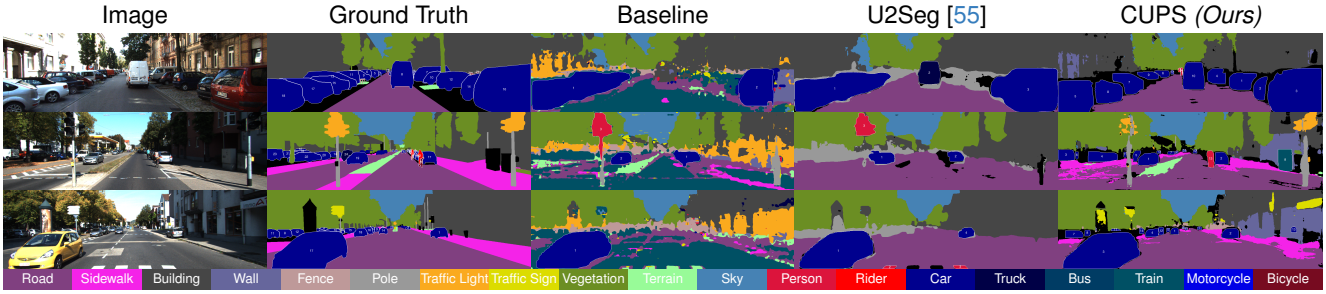
Segmentation model	PQ	SQ	RQ
Mask2Former [18]	25.1	**57.7**	31.7
Panoptic Cascade Mask R-CNN [8, 39]	**26.6**	57.5	**33.5**

Analysis of panoptic segmentation architecture. Our method does not make particular assumptions regarding the downstream panoptic segmentation model. In principle, CUPS can be applied to various panoptic segmentation architectures without significant changes; hyperparameter tuning may be required for optimal accuracy. As a preliminary experiment, we perform stage-1 training (*i.e.*, only pseudo-label training) of CUPS using the Mask2Former [18] architecture and observe comparable panoptic segmentation accuracy relative to the Panoptic Cascade Mask R-CNN baseline. Specifically, Mask2Former achieves slightly inferior RQ but marginally superior SQ, resulting in an overall lower PQ. We attribute this weaker recognition performance to architectural differences: Mask2Former jointly predicts semantic and instance labels per mask, whereas Panoptic Mask R-CNN separates these tasks into two branches, facilitating a more effective application of the drop loss. In particular, Mask2Former

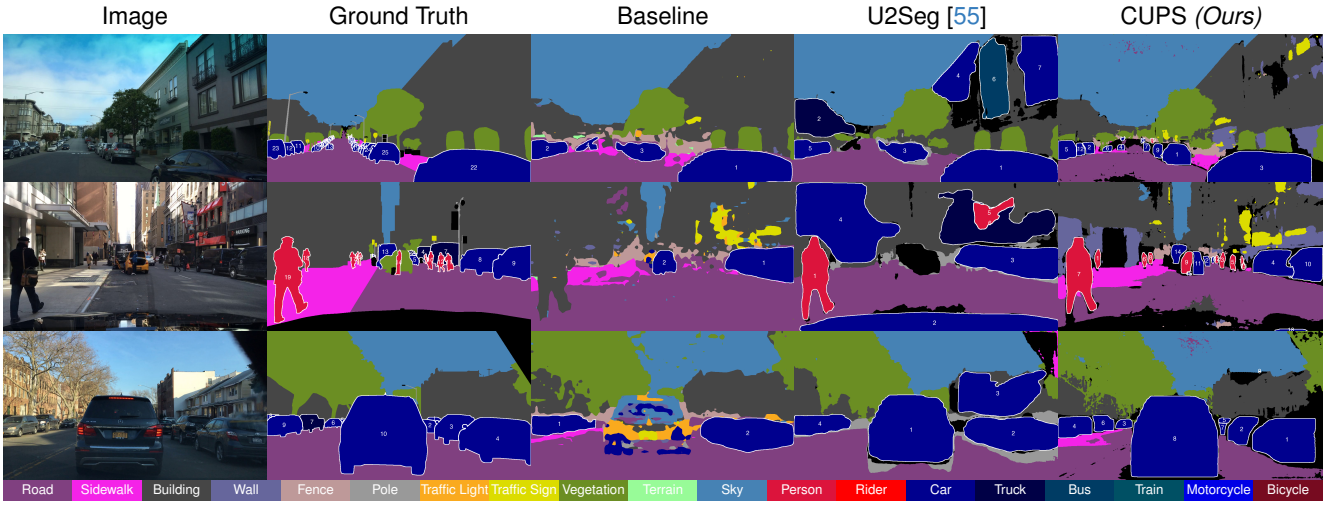
Figure 7. **Qualitative unsupervised panoptic segmentation examples** across all datasets after Hungarian matching. We compare CUPS *(Ours)* to the DepthG+CutLER baseline and U2Seg. CUPS produces more consistent and accurate panoptic segmentations.



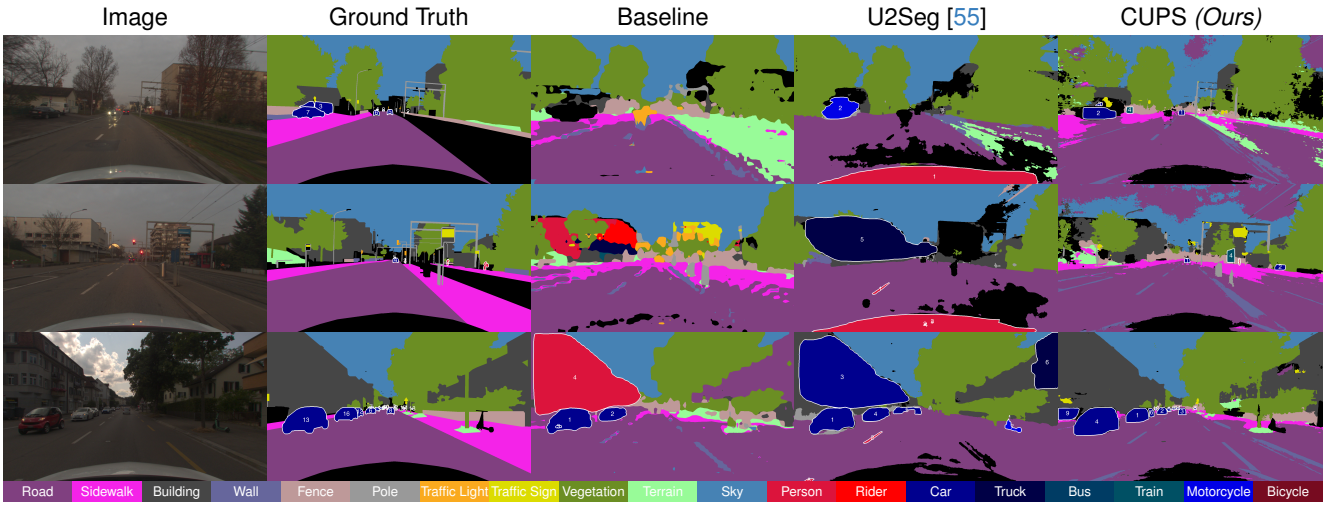
(a) **Cityscapes** — Qualitative unsupervised panoptic segmentation examples.



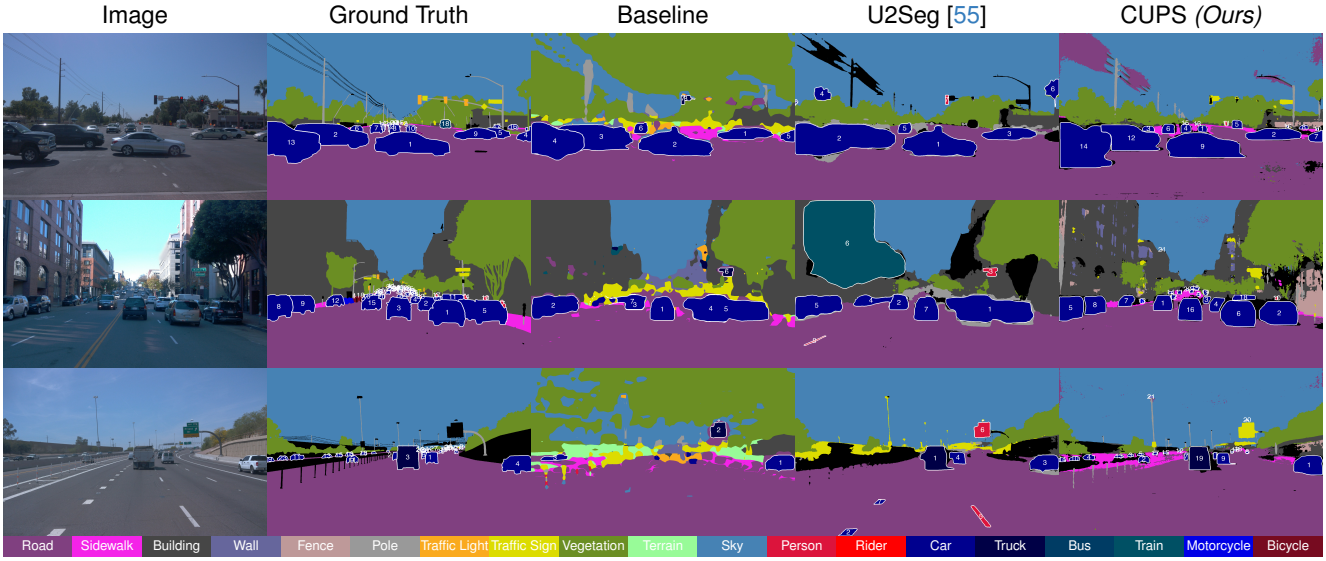
(b) **KITTI** — Qualitative unsupervised panoptic segmentation examples.



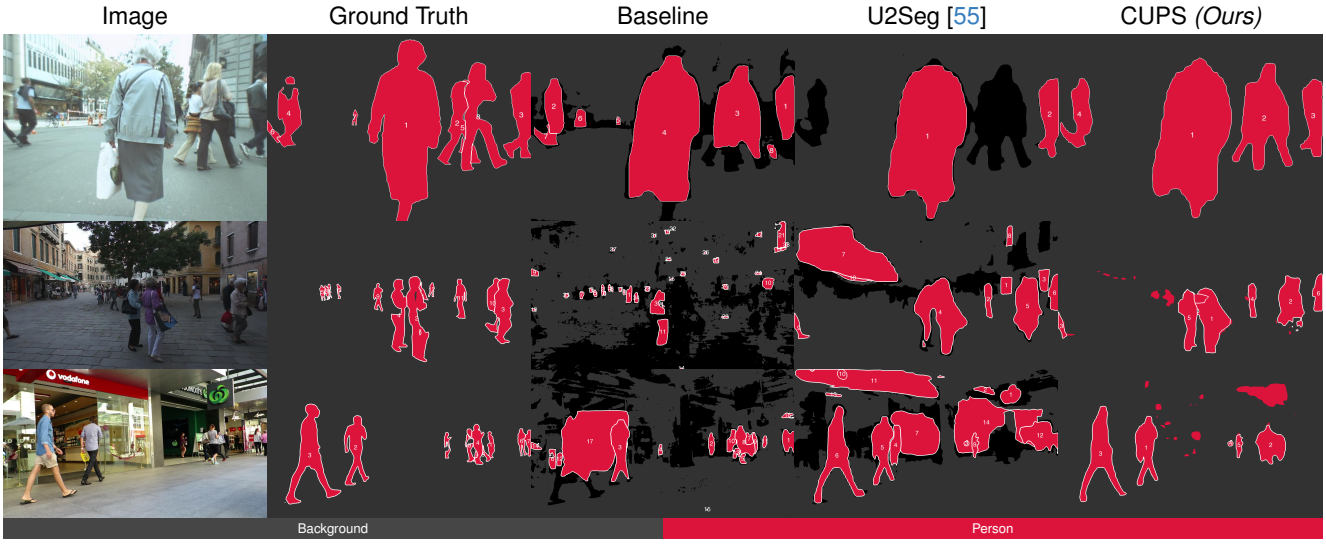
(c) **BDD** — Qualitative unsupervised panoptic segmentation examples.



(d) **MUSES** — Qualitative unsupervised panoptic segmentation examples.



(e) **Waymo** — Qualitative unsupervised panoptic segmentation examples.



(f) **MOTS** — Qualitative unsupervised panoptic segmentation examples.

Figure 8. **Qualitative OOD examples for CUPS on ImageNet val.** Applying the pseudo class to ground-truth matching of CUPS from Cityscapes for visualization purposes.

applies the drop loss to both "thing" and "stuff" predictions, while Panoptic Cascade Mask R-CNN only drops "thing" masks. Our findings indicate that prior work [79] has only partially addressed the application of the drop loss to Mask2Former, thus limiting the effectiveness of the drop loss. While initial results appear promising, further investigation is necessary.

D. Qualitative Results

We show a qualitative comparison of CUPS to the DepthG+CutLER baseline and U2Seg [55] across all datasets in Fig. 7.

On Cityscapes (*cf*. Fig. 7a), we observe a significant qualitative difference to U2Seg. We attribute this improvement to the quality of our pseudo labels, which enable predicting small instances in the background. Despite some errors, such as the "Fence" being incorrectly predicted in small regions of the building in the upper image, CUPS identifies substantially more classes and provides more precise instance segmentation compared to both the baseline and U2Seg. On KITTI (*cf*. Fig. 7b), we observe a similar trend. CUPS detects and segments more objects, offering a finer-grained panoptic segmentation compared to U2Seg, which tends to merge overlapping objects. For instance, in the upper example, the parked cars are incorrectly merged into a single mask by U2Seg, while CUPS successfully separates them. On the BDD dataset (*cf*. Fig. 7c), the impact of the domain shift is evident across all methods. CUPS exhibits minor artifacts, such as predictions related to parts of the ego vehicle or dirt on the windshield. Additionally, signs on buildings are occasionally misclassified as traffic signs. In contrast, U2Seg often produces large, erroneous masks that span across the image, resembling the MaskCut artifacts in Fig. 2. Similarly, for MUSES and Waymo (*cf*. Figs. 7d and 7e), all methods are somewhat affected by the domain shift and challenging viewing conditions. However, CUPS consistently detects instances compared to both other approaches. For the upper Waymo example, one can observe an occasional artifact for CUPS. For example, it incorrectly classifies the shadows forming underneath the vehicles in sunny weather conditions. This is a result of the instance cue being derived from unsupervised flow, which can introduce artifacts due to the apparent motion. MOTS (*cf*. Fig. 7f) is challenging for all approaches. Nonetheless, CUPS produces accurate predictions with fewer artifacts compared to both the baseline and U2Seg, showcasing its robustness even in complex scenarios.

Overall, CUPS predicts less noisy and more accurate semantics, aligning well with the image while predicting significantly more and better instance masks. This observation is in line with our quantitative experiments (*cf*. Sec. 4).

Additionally, we run CUPS and U2Seg on a demo (validation) video sequence from Cityscapes (*cf*. `https://visinf.github.io/cups`). For this analysis, we process each individual frame independently using the respective method and concatenate the outputs into a video, as both methods are designed for per-frame processing. On this sequence, CUPS is qualitatively superior to U2Seg.

Results on object-centric images. To further evaluate the generalization capabilities of our approach, we tested CUPS on randomly selected out-of-domain images, sampled from ImageNet [91]. Qualitative results, shown in Fig. 8, demonstrate that CUPS effectively generalizes to novel domains, viewpoints, and object categories. We find that objects such as tractors, forklifts, and airplanes are classified as cars, which is reasonable given the classes available in Cityscapes. Additionally, objects and surroundings in diverse scenarios are accurately segmented. For instance, despite never encountering a racing car on a mountain road during training, CUPS provides contextually appropriate and coherent segmentation, further highlighting the robustness of our method.

E. Limitations and Future Work

CUPS utilizes stereo videos to extract depth cues for pseudo labeling of complex scenes. Although stereo videos are widely available and are inexpensive to record, overcoming the need for the stereo setup could further broaden the application spectrum. Future work could explore replacing the stereo input with a state-of-the-art self-supervised monocular depth estimation method, such as ProDepth [95].

The evaluation of CUPS has been also largely constrained to driving datasets. This is due to the wide availability of panoptic annotation specifically for this domain. Nevertheless, we believe that CUPS has the potential for applications beyond traffic scenarios, as it relies on domain-agnostic cues, such as depth and motion as well as general-purpose visual representations.

U2Seg and CUPS approach the task of unsupervised panoptic segmentation from two distinct perspectives: object-centric and scene-centric training data. Combining the strengths of both methods could open a promising avenue for future research, offering a more comprehensive solution to the challenges of unsupervised panoptic scene understanding.

An additional direction for future work could scale such an approach by exploring more advanced panoptic segmentation networks, such as Mask2Former [18], and increasing the amount of training data.

References

[87] William A. Falcon, The PyTorch Lightning team. PyTorch Lightning. *GitHub*, (2019). i

[88] Rémi Marsal, Florian Chabot, Angélique Loesch, Hichem Sahbi. BrightFlow: Brightness-change-aware unsupervised learning of optical flow. In *WACV*, pages 2060–2069, 2023. iii, iv

[89] 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, Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. i

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

[91] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(13):211–252, 2015. ii, ix

[92] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, pages 10052–10062, 2021. ii

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

[94] Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, Trevor Darrell. VideoCutLER: Surprisingly simple unsupervised video instance segmentation. In *CVPR*, pages 22755–22764, 2024. v

[95] Sungmin Woo, Wonjoon Lee, Woo Jin Kim, Dogyoon Lee, Sangyoun Lee. ProDepth: Boosting self-supervised multi-frame monocular depth with probabilistic fusion. In *ECCV*, pages 201–217, 2024. x