## Appendix

This Appendix provides additional details and qualitative results organized as follows. In Appendix A, we thoroughly describe the datasets used in our paper. Additional details about concurrent methods and comparison are discussed in Appendix B. A detailed description of the evaluation protocols is presented in Appendix C. Additional results on semi-supervised video segmentation are reported in Appendix D. In Appendix E, we briefly discuss the efficiency and computational overhead of CrOC. Finally, we provide examples of the clusters found on the combined views of complex scene images with the proposed online clustering algorithm in Appendix F.

## A. Datasets

**COCO.** The COCO (Microsoft Common Objects in Context) dataset [29] consists of scene-centric images spanning 91 stuff categories and 80 objects/things categories. The `train2017`, `val2017` and `test2017` splits incorporate approximately 118k, 5k and 41k images, respectively. Additionally, a set of ∼123k unlabeled images, `unlabeled2017`, can be used in conjunction with the `train2017` split to obtain the so-called COCO+ training set.

**COCO-Things.** The COCO-Things dataset follows the implementation of [51]. This dataset is based on COCO images and the panoptic labels of [24]. More precisely, the instance-level labels are merged, and so are the 80 "things" categories, yielding the following 12 super-categories: `electronic`, `kitchen`, `appliance`, `sports`, `vehicle`, `animal`, `food`, `furniture`, `person`, `accessory`, `indoor`, and `outdoor`. As the underlying images are the same as in the COCO dataset, so are the training/validation/test splits.

**COCO-Stuff.** The COCO-Stuff dataset follows the implementation of [51]. The stuff annotations are those of [3]. As for COCO-Things, the 91 "stuff" categories are merged into 15 super-categories: `water`, `structural`, `ceiling`, `sky`, `building`, `furniture-stuff`, `solid`, `wall`, `raw-material`, `plant`, `textile`, `floor`, `food-stuff`, `ground` and `window`. This dataset follows the same training/validation/test splits as in the COCO dataset.

**PVOC12.** The PASCAL VOC12 (PVOC12) dataset [13] is a scene-centric dataset. The `trainaug` split relies on the extra annotations of [17] such that 10582 images with pixel-level labels can be used for the training phase as opposed to the 1464 segmentation masks initially available. The validation set encompasses 1449 finely annotated images. The dataset spans 20 object classes (+1 background class): `person`, `bird`, `cat`, `cow`, `dog`, `horse`, `sheep`, `aeroplane`, `bicycle`, `boat`, `bus`, `car`, `motorbike`, `train`, `bottle`, `chair`, `dining table`, `potted plant`, `sofa`, `tv/monitor` and `background`.

**ADE20K.** The ADE20K dataset [50] is a scene-centric dataset encompassing more than 20K scene-centric images and pixel-level annotations. The labels span 150 semantic categories, including "stuff" categories, *e.g.* `sky`, `road`, or `grass`, and "thing" categories, *e.g.* `person`, `car`, *etc*.

## B. Implementation details

### B.1. Comparison with competing methods

To compare CrOC on an equal footing with concurrent methods, we evaluate all baselines using our evaluation pipeline, except for the evaluation of ResNet50 on the semi-supervised video segmentation task which are taken as is from [49]. With our implementation, the results were worse than the ones reported in [49] or [21]; hence we report their results. Furthermore, for BYOL [15][3], ORL [47], DenseCL [41], SoCo [43], ReSim [45], PixPro [48], VICRegL [2] and CP$^2$ [39], we use publicly available model checkpoints. The only two exceptions are MAE [18] and DINO [6] methods. Indeed, no public model checkpoint exists for ViT-S/16 pre-trained with MAE. Since our implementation builds upon DINO, it is important to have CrOC and DINO models trained in a similar setting for comparison purposes.

**MAE.** The ViT-S/16 is pre-trained under MAE framework on the COCO dataset with the following parameters:

- `mask_ratio`: 0.75
- `weight_decay`: 0.05
- `base_lr`: 0.00015
- `min_lr`: 0.0
- `warmup_epochs`: 40
- `batch_size`: 256
- `epochs`: 300

We use the following decoder architecture:

- `decoder_embed_dim`: 512
- `decoder_depth`: 8
- `decoder_num_heads`: 16

**DINO.** The ViT-S/16 is pre-trained under DINO framework on the COCO dataset with the following parameters[4]:

- `out_dim`: 65536
- `norm_last_layer`: false
- `warmup_teacher_temp`: 0.04
- `teacher_temp`: 0.07
- `warmup_teacher_temp_epochs`: 30
- `use_fp16`: true
- `weight_decay`: 0.04
- `weight_decay_end`: 0.4

[3]The checkpoint for BYOL is provided and trained by the authors of ORL [47].

[4]CrOC uses the same setting.

- clip_grad: 0
- batch_size: 256
- epochs: 300
- freeze_last_layer: 1
- lr: 0.0005
- warmup_epochs: 10
- min_lr: 1e-05
- global_crops_scale: [0.25, 1.0]
- local_crops_number: 0
- optimizer: adamw
- momentum_teacher: 0.996
- use_bn_in_head: false
- drop_path_rate: 0.1

## C. Evaluation protocols

For all evaluation protocols and models, the evaluation operates on the frozen features of the backbone. The projection heads, if any, are simply discarded. The output features from layer4 of ResNet50 are used in all downstream tasks. The resulting features have dimension $d = 2048$, whereas the spatial tokens of a ViT-S/16 have dimension $d = 384$ only. We concatenate the spatial tokens from the last $n_b$ transformer blocks, similar to [6], to compensate for that difference.

**Transfer learning via linear segmentation.** Our implementation is based on that of [38, 51]. The input images are re-scaled to $448 \times 448$ pixels and fed to the frozen model. Following existing works [51], in the case of ResNet50, dilated convolutions are used in the last bottleneck layer such that the resolution of the features is identical for all models. Prior to their processing by the linear layer, the features are up-sampled with bilinear interpolation such that the predictions and the ground-truths masks have the same resolution. Unlike previous works [2, 38, 51], we use Adam [23] as an optimizer instead of SGD. Indeed, we observe that this led to significant improvements for all baselines, indicating that the reported results were obtained in a sub-optimal regime and hence did not fully reflect the quality of the learned features. We report results on the PVOC12 validation set after training the linear layer on the trainaug split for 45 epochs. For the COCO-Things and COCO-Stuff, the linear layer is first trained for 10 epochs on the training set and subsequently evaluated on the validation set. Regardless of the evaluation dataset and model, we find that a learning rate lr=1e-3 works well and that the selected number of epochs is sufficient to reach convergence. Note that contrary to [51], which randomly samples 10% of the COCO-Things/-Stuff training images, we use the full set of available images to avoid introducing additional randomness in the results.

For the evaluation with ADE20K, we rely on MMSegmentation [9] and the *40k iterations schedule*. We set the batch size to 16, and we report for each method the best result after trying learning rates in $\{1e{-}03, 8e{-}04, 3e{-}04, 1e{-}04, 8e{-}05\}$.

**Transfer learning via unsupervised segmentation.** Our implementation is based on that of [38, 51]. The input images are re-scaled to $448 \times 448$ pixels and fed to the frozen model. Following existing works [51], in the case of ResNet50, dilated convolutions are used in the last bottleneck layer such that the resolution of the features is identical for all models. Similarly to [51], the ground-truth segmentation masks and features are down-/up-sampled to have the same resolution ($100 \times 100$). Consequently, we ran K-Means on the spatial features of all images with as many clusters as there are classes in the dataset. A label is greedily assigned to each cluster with Hungarian matching [25]. We report the mean Intersection over Union (mIoU) score averaged over five seeds. Importantly, [51] observed that better results could be obtained by using a larger number of clusters $K$ than the number of classes in the dataset and hereby having clusters of object-parts instead of objects. Indeed, if this approach provides information on the consistency of the features within object-part clusters, it does not tell anything about the inter-object-parts relationship. For instance, the mIoU scores will reflect the ability of features corresponding to "car wheels" to be clustered together and similarly for "car body" features, but it won't be impacted by the distance of the two clusters from one another, which is undesirable. We report results on the PVOC12, COCO-Things, and COCO-Stuff validation sets.

**Semi-supervised video object segmentation.** The semi-supervised video object segmentation evaluation follows the implementation of [6, 49]. We report the mean contour-based accuracy $\mathcal{F}_m$, mean region similarity $\mathcal{J}_m$ and their average $(\mathcal{J}\&\mathcal{F})_m$ on the 30 videos from the validation set of the DAVIS'17 [34]. The following parameters are used:

- n_last_frames: 7
- size_mask_neighborhood: 12
- topk: 5

## D. Semi-supervised video segmentation results

Good results are obtained on the semi-supervised video segmentation (Table A1), indicating the ability of CrOC to produce features consistent through time and space.

## E. Computational overhead

An important property of CrOC is that it generates pseudo-labels/cluster assignments online. Consequently, this step must be efficient. In Table A2, we verify that the operations inherent to the clustering step amount to less than 10% of the total time of the CrOC pipeline.

Table A1. **Semi-supervised video object segmentation task**. The frozen spatial features are evaluated on the video segmentation task by nearest neighbor propagation DAVIS'17 challenge. The mean region similarity $\mathcal{J}_m$, mean contour-based accuracy $\mathcal{F}_m$, and their average $(\mathcal{J}\&\mathcal{F})_m$ are reported. † indicates results taken from [49].

| Method | Model | Dataset | $(\mathcal{J}\&\mathcal{F})_m$ | $\mathcal{J}_m$ | $\mathcal{F}_m$ |
|---|---|---|---|---|---|
| *Global features* | | | | | |
| DINO [6] | ViT-S/16 | COCO | 57.1 | 55.3 | 58.9 |
| *Local features* | | | | | |
| DenseCL† [41] | ResNet50 | ImageNet | 50.7 | 52.6 | 48.9 |
| ReSim† [45] | ResNet50 | ImageNet | 49.3 | 51.2 | 47.3 |
| DetCo† [46] | ResNet50 | ImageNet | 56.7 | **57.0** | 56.4 |
| ODIN [21] | ResNet50 | ImageNet | 54.1 | 54.3 | 53.9 |
| MAE [18] | ViT-S/16 | COCO | 48.9 | 47.3 | 50.6 |
| $CP^2$ [39] | ViT-S/16 | ImageNet | 53.7 | 51.3 | 56.1 |
| *Ours* | | | | | |
| CrOC | ViT-S/16 | COCO | 57.4 | 55.7 | 59.1 |
| CrOC | ViT-S/16 | COCO+ | **58.4** | 56.5 | **60.2** |
| CrOC | ViT-S/16 | ImageNet | 44.7 | 43.5 | 45.9 |

Table A2. **The runtime of the main operations in CrOC** for a batch size of 256 samples distributed over 2 Tesla V100. CrOC-specific operations are highlighted.

| operation | absolute time [ms] | relative time [%] |
|---|---|---|
| $f_t(\cdot) + \bar{h}_t(\cdot)$ | 177.8 | 21.1 |
| $f_s(\cdot) + \bar{h}_s(\cdot)$ | 183.9 | 21.9 |
| $\mathbf{Q}^* = \mathcal{C}(\cdot)$ | 67.0 | 8 |
| $h_t(\cdot) + h_s(\cdot)$ | 4.8 | 0.5 |
| `backprop.` + `EMA` | 408.0 | 48.5 |
| *total* | 841.5 | 100 |

## F. Qualitative results

The cluster assignments found by CrOC's dedicated online clustering algorithm $\mathcal{C}$ over the combined views are depicted in Fig. A1. The model used to generate the illustrated assignments is pre-trained on the COCO+ for 300 epochs with CrOC and the following meta-parameters: $\lambda_{\text{pos}} = 4$, $K_{\text{start}} = 12$ and `values` tokens. During training, we use the same augmentations as in DINO [6]; consequently, we visualize the generated masks based on augmented views in the same manner, such that the results depicted in Fig. A1 truly reflect the consistency enforced by CrOC.

Figure A1. **Illustration of the clusters found online in the space of the combined views**. Rows correspond to combined views and columns to heads of the ViT. Bicubic interpolation is used to up-sample the assignments $\mathbf{Q}^*$ to the same resolution as the images.