– Supplementary material –
Decoupled Semantic Prototypes enable learning from diverse annotation types for semi-weakly segmentation in expert-driven domains

Simon Reiß[1] Constantin Seibold[1] Alexander Freytag[2] Erik Rodner[3] Rainer Stiefelhagen[1]
[1]Karlsruhe Institute of Technology [2]Carl Zeiss AG [3]University of Applied Sciences Berlin
{simon.reiss, constantin.seibold, rainer.stiefelhagen}@kit.edu,
alexander.freytag@zeiss.com, erik.rodner@htw-berlin.de

Abstract

In this supplementary material, we provide additional details and results to complement and strengthen the findings of the main paper and to make reproducibility as easy as possible. A detailed derivation of how to efficiently implement DSP's loss term can be found in Section 1. Section 2 gives an explanation for the choice of the segmentation architecture as used for the experiments in the main paper and contains details about the configurations. In Section 3, we provide additional information on dataset sizes and construction of weak labels. Section 4 contains additional details with respect to baseline methods and their ablations. Quantitative results as numerical values are shown in Section 5, thereby complementing the graphical result representations given in the main paper. In addition, Section 6 contains additional qualitative results, with a specific focus on failure cases and a visualization of DSP's embedding space during training. Finally, we give in Section 7 answers to several questions about DSP.

1. Details on efficiently implementing DSP's loss term

As shown in the main paper, DSP's loss term L_{DSP} is built on the idea of decoupled contrastive learning by Yeh *et al*. [22]. This is especially attractive for segmentation tasks, i.e., with large amounts of pixels, as it removes the necessity to compute different negatives for each positive case. In the following, we will give a more detailed description of how the final loss term as shown in Equation (9) of the main paper can be implemented efficiently (as hinted on in the last part of section 3.5 of the main paper).

Let us start with the loss term L_{DSP} as in Equation (9) of the main paper:

$$L_{DSP} = \sum_{l \in \{m,b,p,im\}} \lambda_l \sum_{c=1}^{C} \sum_{f_i \in \Omega_c^l} L(f_i, c) \quad . \tag{S.1}$$

Let us look at the last sum in Equation (S.1). For a specific class c and an annotation type l, the last sum can be expressed as follows given the definition of $L(f_i, c)$ from Equation (6) of the main paper:

$$\sum_{f_i \in \Omega_c^l} L(f_i, c) = \sum_{f_i \in \Omega_c^l} -\log \frac{\exp\left(s_c(f_i, P_c)/\tau\right)}{Z_{i,c}} \quad . \tag{S.2}$$

Recall that $Z_{i,c}$, as defined in Equation (8) of the main paper is the denominator of our decoupled contrastive term:

$$Z_{i,c} = \sum_{j=1}^{B \cdot H \cdot W} \sum_{k=1, c \notin \mathcal{A}_j \vee k \neq c}^{C} \exp\left(s_k(f_j, P_k)/\tau\right) \quad . \tag{S.3}$$

As a reminder from section 3.4: the requirement $c \notin \mathcal{A}_j$ OR $k \neq c$ as given under the second sum is motivated by the design that all associations between pixel-embeddings and prototypes shall be counted as negatives either when i) $k \neq c$ (*i.e.* a negative association can safely be assumed) or ii) for the case of $k = c$ but then only if $c \notin \mathcal{A}_j$ (*i.e.* the class c is not possible under the available annotation for pixel j).

Due to the these two requirements, $Z_{i,c}$ and hence the denominator in Eq. (S.2) becomes independent of f_i. Thus, we write Z_c for short. Simultaneously, we apply the logarithmic division law and simplify the right part of Eq. (S.2)

to obtain:

$$\sum_{f_i \in \Omega_c^l} L(f_i, c) = \sum_{f_i \in \Omega_c^l} -(\log \exp\left(s_c(f_i, P_c)/\tau\right) - \log Z_c) \tag{S.4}$$

$$= \sum_{f_i \in \Omega_c^l} -\left(s_c(f_i, P_c)/\tau - \log Z_c\right) \quad . \tag{S.5}$$

At this point, with the independence of Z_c from f_i, we could simply scale the denominator Z_c with the number of positives in the set Ω_c^l to bring it in front of the summation:

$$\sum_{f_i \in \Omega_c^l} L(f_i, c) = |\Omega_c^l| \cdot \log Z_c - \sum_{f_i \in \Omega_c^l} s_c(f_i, P_c)/\tau \quad . \tag{S.6}$$

However, we found that scaling the negatives Z_c by a value > 1 creates large loss magnitudes which were experimentally hard to weight. Hence, we instead re-scale the loss as given in Eq. (S.6) by dividing by $|\Omega_c^l|$:

$$L(c) \doteq \log Z_c - \frac{1}{|\Omega_c^l|} \cdot \sum_{f_i \in \Omega_c^l} s_c(f_i, P_c)/\tau \quad . \tag{S.7}$$

Putting everything together, we obtain the final implementation $L_{DSP}^{\star}$ for L_{DSP} as used in our code as follows:

$$L_{DSP}^{\star} = \sum_{l \in \{m,b,p,im\}} \lambda_l \sum_{c=1}^{C} L(c) \cdot \delta(|\Omega_c^l| \neq 0) \quad , \tag{S.8}$$

where the delta function $\delta(\cdot)$ prevents from division by zero in the case of $|\Omega_c^l| = 0$ (*i.e.* when class c does not occur in the batch). This loss is used for weakly- and strongly augmented inputs.

2. A note on architecture choices

In order to obtain comparable results, all methods in our evaluation share the same segmentation architecture. The results as presented in the main paper have all been obtained with a Unet [16]. To be more precise, we leverage a Unet with successive feature-map channel sizes of $\{64, 128, 256, 512, 1024\}$ in the encoder and the corresponding reverse channel sizes in the decoder. This gives a versatile and yet efficient network with $\sim$ 22 million trainable parameters.

So – why Unets and no recent architectures that give state-of-the-art results on domains like Cityscapes [4]? In preliminary experiments, we indeed explored recent transformer architectures. Specifically, we analyzed some baseline methods on Segformers by Xie *et al*. [21] and on Swin-Unets by Cao *et al*. [2]. However, we observed unstable trainings and extremely poor segmentation results, both in cases when trained from scratch as well as when initialized with ADE-20K [24] pre-trained weights. Qualitative results of failed segmentations with these architectures are shown in Figure 1. As can be seen, the segmentation results which we were able to obtain by training these architectures are clearly not acceptable. We assume that one main reason for insufficient applicability of Segformers and Swin-Unets in the task at hand is the lack of pre-trained weights for the domain of electron microscopy imagery. At the same time, we expect that weights obtained from the natural domain do not allow these architectures to transfer sufficiently well to application domains which are starkly different from natural images. Furthermore, we observed training instability of these architectures with respect to hyperparameters and schedulers of optimization parameters, *e.g*. learning rate warm starts and learning rate decay, which Unets are known to be robust against. Finally, especially Segformers might be negatively impacted from the padding of images for same-sized images in the batch.

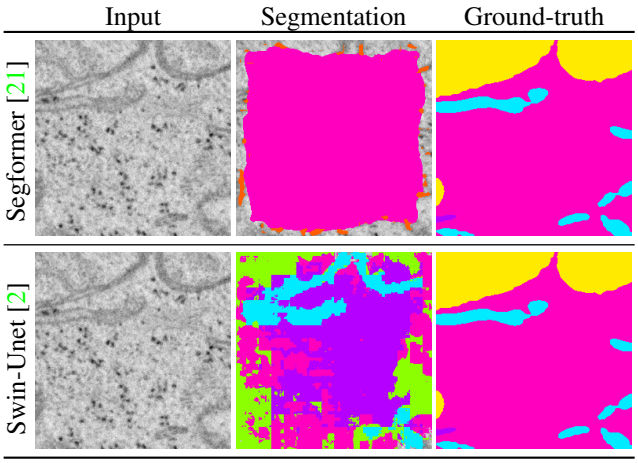

Figure 1. Results for segmentation transformers trained from scratch with full supervision on *HELA-2*.

Given the previous considerations, and given the requirements for a statistically sound evaluation which requires to robustly train > 1500 models for our experiments, we decided for Unets. This choice is further supported in many non object-centric, expert-driven domains, especially in the medical area [7, 9, 10]. At the same time, we expect that improvements like pre-training on in-domain data [13] will enable segmentation transformers also for this domain. For the future, we are especially interested to see how transformers perform when being adapted to and trained with *Decoupled Semantic Prototype*.

3. Additional information about the datasets

As described in the main paper, we conducted experiments on a total of four cell organelle datasets from

Class name	*HELA-2*	*HELA-3*	*MACROPHAGE-2*	*JURKAT-1*
Extracellular Space	✓	✓	✓	✓
Plasma Membrane	✓	✓	✓	
Mitochondria	✓	✓		✓
Vesicle	✓	✓	✓	✓
MVB	✓	✓	✓	✓
Lysosome	✓		✓	
Endoplasmic Reticulum	✓	✓	✓	✓
Nucleus	✓	✓	✓	✓
Nucelear Envelope	✓	✓		
Microtubule	✓	✓		✓
Cytosol	✓	✓	✓	✓

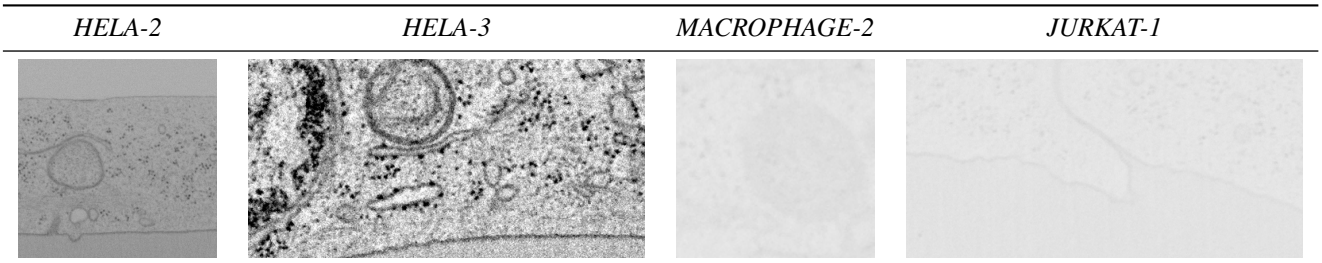

Figure 2. Top: Organelle classes which are present in at least three sub-volumes in the different datasets of Heinrich *et al*. [7] for train/validation/test cross-validation are indicated with checkmarks. Bottom: Example images from the different cell organelle electron microscopy datasets used in the main paper.

the OPENORGANELLE data as outlined in Heinrich *et al*. [7]. In low annotation scenarios, rigorous cross-validation is necessary to get robust insight into method performances. Therefore, we performed cross-sub-volume cross-validation, *i.e*. we split the set of annotated sub-volumes of the datasets into train/validation/test sets multiple times. As mentioned in the main paper, this was done 10 times for *HELA-2* and 5 times for the remaining datasets.

For creating these splits, we obviously can only include classes that occur in at least three annotated sub-volumes to distribute them among train, validation, and test. Directly applying this constraint to the four cell organelle datasets would leave us with little annotated data. Therefore, as mentioned in the main paper, we merged the original 37 classes into 17 merged classes. For merging classes, we followed the the organelle class hierarchy provided in the code of the original publication[1]. In the top part of Fig. 2, we list the subset of classes from these 17 merged classes that satisfy the occurence requirement in three sub-volumes on the different datasets.

To get an intuition about the difficulty, properties, and differences of these cell organelle datasets, we show in the lower part of Fig. 2 example images from all FIB-SEM datasets used in the experiments. As can clearly be seen, the different datasets show a large variance in appearance as well as aspect ratios (different aspect ratios of labeled regions also occur within datasets).

3.1. Dataset sizes

As we outline in the main paper, we performed cross-sub-volume cross-validation either 10 or 5 times. Here, we report the average number of images in the training-, validation-, and testing sets for the four datasets:

- *HELA-2*: 2321 training images, 924 validation images, and 930 testing images
- *HELA-3*: 1634 training images, 731 validation images, and 791 testing images
- *MACROPHAGE-2*: 1482 training images, 685 validation images, and 740 testing images
- *JURKAT-1*: 1525 training images, 745 validation images, and 742 testing images

[1] https://github.com/saalfeldlab/CNNectome/blob/7c5250edf2ba8ce43127c457b755ea30721f638f/CNNectome/utils/hierarchy.py

3.2. How to obtain weak annotations from pixel-wise masks

The OPENORGANELLE dataset by Heinrich *et al*. [7] comes with pixel-wise annotations. From these, we also derived weak annotations to conduct semi-weakly supervised experiments with diverse annotation types. For creating image-level labels, we simply extracted the unique classes that occur in each pixel-wise mask. To extract bounding boxes, we computed the connected components of each mask, extracted bounding boxes, and assigned the class from the corresponding mask. For creating point annotations from masks, we finally would have several choices. We draw inspiration from the way humans generally point at objects, *i.e*. clicking on the medial axis [6] or in the center of regions [3]. Hence, we computed the medoids of the connected components of the mask and obtained one point annotation from each medoid location, associated with the class label of the corresponding mask.

4. Additional details for baseline methods and their ablation

Batch construction During training, we ensured for the creation of mini-batches that on average all annotation types used in a training scenario are equally present. This "stratified sampling" was applied for all methods. Thereby, losses based on specific annotation types are computed on a roughly regular schedule rather than after an irregular number of iterations. This is especially useful when the training scenario has severely unbalanced portions of pixel-wise and weak annotations (*e.g*. at ACR $= 64$). To obtain this, we oversampled images from less frequent annotation types, similar to the commonly done oversampling of the set of labeled images for semi-supervised segmentation [15, 17].

Pseudo-label [12] We implemented an online version of pseudo-labeling, *i.e*. computing the labeles for un- or weakly labeled images on the fly while training the segmentation network. Although this is in contrast to precomputing pseudo-labels and then continuing training with fixed pseudo-labels, we decide for the online version as all other compared methods also work in an online setting.

FixMatch [17] The original FixMatch approach was designed for image classification and computes pseudo-labels with a given threshold on the predictions. Thereby, it disregards pseudo-labeled images that have a lower prediction score than this threshold. We investigated the effect of such a threshold when being applied in the segmentation scenario and report results in the left part of Table 1. Surprisingly simply not applying a threshold lead to the best results. Furthermore, FixMatch leverages strong augmentations, including CutOut [5] which randomly sets image regions of size 32×32 in the input image to black. We analyzed the effect of how often to apply CutOut (*i.e*. for how often to successively apply CutOut with a probability of 0.5). Results of this ablation study are given in Table 1. We see that the best result is obtained with applying Cutout up to nine times. We also use these strong augmentations for DSP and Con2R [15].

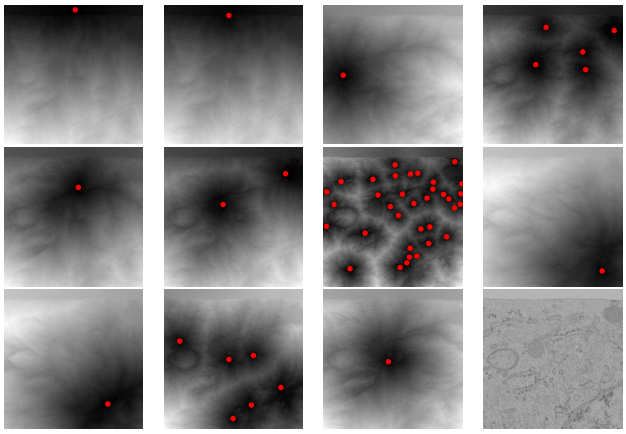

Figure 3. Class-wise geodesic distance maps based on point annotations (marked in red and enlarged for visibility). In the lower right, the input image from *HELA-2* is displayed.

threshold	DICE	# CutOut	DICE
0.0	51.7 ± 3.6	0	48.5 ± 3.7
0.1	51.3 ± 4.2	1	49.0 ± 4.1
0.2	51.2 ± 3.9	2	50.0 ± 3.1
0.3	51.4 ± 3.5	3	50.3 ± 3.6
0.4	51.4 ± 4.2	4	50.4 ± 4.0
0.5	51.0 ± 4.0	5	50.4 ± 4.0
0.6	51.6 ± 3.9	6	50.4 ± 4.0
0.7	51.0 ± 3.7	7	50.8 ± 3.5
0.8	51.0 ± 3.6	8	50.8 ± 4.2
0.9	50.8 ± 3.6	9	51.7 ± 3.6
0.95	50.5 ± 3.4	10	51.1 ± 3.4

Table 1. Ablation study for the baseline method FixMatch by Sohn *et al*. [17] for a standard semi-supervised learning scenario. Validation accuracy measured as DICE score on *HELA-2*, $ACR = 8$. Left: Varying the pseudo-label threshold in FixMatch. Right: Varying the maximum number of applying CutOut as strong augmentation.

Classification branch For the classification branch architecture, which is trained on top of the Unet architecture [16], we followed the original publication from Mlynarski *et al*. [14] as closely as possible. To gauge with the different image size used in the original implementation, we merely needed to add four more convolutions with ReLU activation after the mean pooling operation and single convolution to end up at the desired size of 11×11. From there, we apply the exact same classification branch architecture consisting of linear layers, ReLU activations, and residual connection.

We also investigated whether the method of Bae *et al*. [1]

Method	$\mathcal{I}$	$\mathcal{B}$	$\mathcal{P}$	$ACR = 1$	$ACR = 2$	$ACR = 4$	$ACR = 8$	$ACR = 16$	$ACR = 32$	$ACR = 64$
UNet	–	–	–	50.1 ± 4.6	50.3 ± 5.6	48.2 ± 4.9	43.6 ± 7.0	34.2 ± 6.6	24.6 ± 3.7	20.2 ± 6.5
CLS Branch [14]	✓	–	–	50.4 ± 5.5	50.3 ± 5.1	50.4 ± 4.6	47.3 ± 5.8	43.8 ± 6.4	34.6 ± 7.3	26.1 ± 8.2
Box Proj. [18]	–	✓	–	48.4 ± 4.9	49.7 ± 5.2	49.9 ± 4.6	47.8 ± 5.9	43.0 ± 5.7	33.5 ± 6.5	26.7 ± 8.5
Euclidean branch	–	–	✓	51.3 ± 4.6	50.9 ± 5.2	50.4 ± 5.3	48.7 ± 6.1	41.4 ± 6.9	31.0 ± 5.9	19.9 ± 7.5
Geodesic branch	–	–	✓	50.4 ± 4.2	49.6 ± 5.4	50.4 ± 4.0	48.6 ± 6.0	42.2 ± 6.0	33.5 ± 5.8	23.6 ± 6.1
Pseudo-label [12]	✓	–	–	50.1 ± 4.6	50.5 ± 5.5	50.2 ± 4.9	49.3 ± 5.4	46.1 ± 6.2	35.8 ± 6.9	27.9 ± 6.8
	–	✓	–	50.1 ± 4.6	50.7 ± 5.1	50.6 ± 5.1	48.1 ± 4.7	45.2 ± 6.2	34.3 ± 6.2	27.9 ± 6.0
	–	–	✓	50.1 ± 4.6	50.9 ± 5.0	50.1 ± 5.0	48.5 ± 5.1	45.8 ± 5.1	35.1 ± 6.7	26.9 ± 6.3
Con2R [15]	✓	–	–	51.6 ± 4.3	52.4 ± 4.9	51.4 ± 5.2	48.9 ± 7.1	43.4 ± 5.7	32.6 ± 5.3	22.1 ± 8.8
	–	✓	–	51.6 ± 4.3	53.0 ± 5.0	51.6 ± 5.2	49.3 ± 5.6	43.8 ± 7.0	33.6 ± 5.9	22.1 ± 10.1
	–	–	✓	51.6 ± 4.3	52.4 ± 4.6	51.5 ± 5.4	49.2 ± 6.4	43.3 ± 7.5	32.8 ± 5.4	22.1 ± 8.8
FixMatch [17]	–	–	–	52.9 ± 3.9	53.5 ± 4.5	53.6 ± 4.1	53.0 ± 5.1	48.0 ± 6.7	37.8 ± 7.9	22.4 ± 11.7
	✓	–	–	52.9 ± 3.9	53.8 ± 4.5	53.7 ± 4.7	52.4 ± 4.6	49.1 ± 6.9	42.6 ± 8.2	33.0 ± 10.5
	–	✓	–	52.9 ± 3.9	53.3 ± 4.0	52.8 ± 4.7	52.7 ± 4.4	51.2 ± 5.7	43.9 ± 10.5	39.4 ± 8.9
	–	–	✓	52.9 ± 3.9	53.7 ± 4.5	53.5 ± 4.6	52.1 ± 4.7	49.5 ± 6.1	42.5 ± 9.1	32.6 ± 10.8
DSP (Ours)	✓	–	–	53.4 ± 3.6	53.8 ± 5.0	53.5 ± 4.5	52.0 ± 5.6	49.8 ± 6.3	42.4 ± 8.2	32.5 ± 10.4
	–	✓	–	53.4 ± 3.6	53.7 ± 4.8	53.0 ± 3.6	52.9 ± 5.0	50.9 ± 5.7	47.7 ± 7.0	44.2 ± 10.1
	–	–	✓	53.4 ± 3.6	53.7 ± 4.4	53.5 ± 4.9	53.4 ± 4.8	52.3 ± 4.6	47.7 ± 8.2	37.7 ± 12.5

Table 2. Segmentation accuracy of different methods at increasing pixel-wise annotation compression rates, measured as mean and standard deviation in DICE. Results are obtained from 10 splits. Random baseline: 5.1 ± 0.3 DICE. Class-prior baseline: 6.1 ± 0.6 DICE.

can give additional benefits, *i.e.* if restricting the segmentation prediction by the classifier's output during inference can improve the result. However, we did not find better results in ablation studies. We assume that the initial observation by Bae *et al.* specifically holds for exceptionally good classifiers as available in object-centric image domains, but does not necessarily generalize to cell organelle images.

Euclidean/Geodesic point branch This baseline leverages an auxiliary output-head which regresses distance maps that are generated via point labels. For each class, a distance map is computed based on point annotations, *i.e.* the smallest distance from every pixel to all point annotations of this class is computed. We implement these distance maps for the Euclidean distance and the Geodesic distance. The choice of Geodesic distance is inspired by point- or click annotations. These are frequently used for medical interactive segmentation [20] and weakly supervised medical segmentation [23], where the Geodesic distance can be exploited. Regressing distance maps is also commonly explored in medical and cell datasets [8, 11]. We visualize an example of such a point-based geodesic map in Figure 3. During training, such Geodesic maps serve as targets to efficiently exploit point annotations and to supply the model with more structural information than singular points alone could offer.

Box loss We integrated the bounding box-based loss of Tian *et al.* [18] for the scenario with pixel-wise annotations and boxes and directly applied it on top of the segmentation output-head for the boxes that are supplied in this scenario.

5. Quantitative results in numerical form

In the main paper, we provided all results as graphical representations by plotting segmentation accuracy against increasing ACRs. The graphical representation aimed at making the progression of accuracy easier to view and better to interpret. For completeness, we provide here underlying numerical results (rounded to one decimal) for all presented experiments.

5.1. Numerical results for all methods

The *HELA-2* results for annotation type pairs are displayed in Tab. 2 with checkmarks indicating the weak annotation type used alongside masks. For models trained with all supervision types mixed, results are shown in Tab. 3 for *HELA-2*, *HELA-3*, *MACROPHAGE-2*, and *JURKAT-1*.

5.2. The benefit of pseudo-label filtering

We further investigated the effect of pseudo-label filtering in the scenario of semi-weakly supervised segmentation. I.e., does the filtering of pseudo-labels based on available weak annotation give benefits for training? For this,

Method	$\mathcal{U}$	$\mathcal{I}$	$\mathcal{B}$	$\mathcal{P}$	$ACR = 1$	$ACR = 2$	$ACR = 4$	$ACR = 8$	$ACR = 16$	$ACR = 32$	$ACR = 64$
HELA-2											
UNet	–	–	–	–	50.1 ± 4.6	50.3 ± 5.6	48.2 ± 4.9	43.6 ± 7.0	34.2 ± 6.6	24.6 ± 3.7	20.2 ± 6.5
FixMatch [17]	✓	✓	✓	✓	52.9 ± 3.9	$\mathbf{53.8 \pm 4.4}$	53.0 ± 4.2	52.9 ± 4.7	50.2 ± 6.3	46.0 ± 7.4	36.7 ± 11.6
DSP (Ours)	✓	✓	✓	✓	$\mathbf{53.4 \pm 3.6}$	53.7 ± 4.5	$\mathbf{54.0 \pm 4.0}$	$\mathbf{54.2 \pm 4.6}$	$\mathbf{52.1 \pm 5.6}$	$\mathbf{51.6 \pm 6.0}$	$\mathbf{49.5 \pm 6.1}$
HELA-3											
UNet	–	–	–	–	47.3 ± 7.2	45.6 ± 7.4	44.8 ± 8.4	41.6 ± 8.5	42.0 ± 6.5	34.5 ± 6.4	28.8 ± 9.4
FixMatch [17]	✓	✓	✓	✓	54.2 ± 3.4	$\mathbf{54.5 \pm 3.7}$	$\mathbf{55.9 \pm 2.8}$	$\mathbf{55.7 \pm 2.7}$	54.6 ± 3.0	48.7 ± 4.1	49.4 ± 5.4
DSP (Ours)	✓	✓	✓	✓	$\mathbf{54.5 \pm 3.9}$	53.9 ± 5.0	54.6 ± 3.0	55.5 ± 2.1	$\mathbf{56.2 \pm 1.7}$	$\mathbf{53.3 \pm 5.0}$	$\mathbf{52.4 \pm 4.2}$
MACROPHAGE-2											
UNet	–	–	–	–	30.7 ± 5.5	29.2 ± 7.7	22.1 ± 4.2	19.0 ± 6.3	10.5 ± 7.5	5.7 ± 2.7	9.3 ± 6.9
FixMatch [17]	✓	✓	✓	✓	44.3 ± 2.1	42.5 ± 7.2	40.1 ± 6.7	23.7 ± 8.4	12.2 ± 6.8	11.7 ± 8.4	13.3 ± 18.8
DSP (Ours)	✓	✓	✓	✓	$\mathbf{45.0 \pm 2.6}$	$\mathbf{43.6 \pm 8.5}$	$\mathbf{43.0 \pm 8.9}$	$\mathbf{31.3 \pm 10.4}$	$\mathbf{21.1 \pm 8.9}$	$\mathbf{14.5 \pm 11.4}$	$\mathbf{15.3 \pm 14.8}$
JURKAT-1											
UNet	–	–	–	–	28.4 ± 8.2	26.4 ± 6.3	21.1 ± 5.1	14.2 ± 7.1	5.0 ± 2.0	8.2 ± 3.4	5.1 ± 1.9
FixMatch [17]	✓	✓	✓	✓	32.7 ± 9.6	$\mathbf{32.6 \pm 8.6}$	$\mathbf{32.3 \pm 9.5}$	15.7 ± 7.1	5.7 ± 3.5	6.3 ± 3.9	$\mathbf{6.5 \pm 2.7}$
DSP (Ours)	✓	✓	✓	✓	$\mathbf{32.9 \pm 9.1}$	32.4 ± 8.3	29.5 ± 5.3	$\mathbf{18.0 \pm 7.6}$	$\mathbf{7.8 \pm 4.1}$	$\mathbf{10.5 \pm 5.9}$	6.3 ± 3.6

Table 3. Segmentation accuracy of different methods at increasing pixel-wise annotation compression rates with weak labels of all types distributed uniformly, measured as mean and standard deviation in DICE. Results are obtained from 5 splits (10 for *HELA-2*).

we trained the best semi-supervised learning baseline FixMatch [17] with and without pseudo-label filtering. Results can be found in Tab. 2, first row in FixMatch results for without pseudo-label filtering, and remaining three rows in FixMatch results with pseudo-label filtering. We observe that in the low ACR scenarios, pseudo-label filtering does not give additional benefits as only few weakly labeled examples are integrated. However, especially when training with few pixel-wise masks, *i.e.* ACRs $\in \{16, 32, 64\}$ we see a tremendous positive effect. Hence, we conclude that FixMatch coupled with pseudo-label filtering is a simple and strong semi-weakly supervised baseline.

6. Supplementary qualitative results

6.1. Segmentation results

In the top three rows of Fig. 4, we provide additional qualitative results from DSP on an image from the *HELA-2* dataset. As can be seen, DSP is able to uncover small organelle structures and start capturing their semantics *even* with very few pixel-wise annotated masks in training and a mix of annotation types.

In the middle four rows of Figure 4, we show the qualitative segmentation results of DSP when being trained with different combinations of annotation types (all scenarios from the main paper). We find two observations worth to note. First, we see point annotations allow to quickly learn to find central regions which lie on the organelles. Second, we also see that training with bounding boxes leads to models which slightly oversegment the organelles. These observations combined can serve as explanation why the combination of these annotation types (in the mixed scenario) works better than each type individaully.

Finally, we provide a failure case where DSP leads to an insufficient segmentation in the last row of Fig. 4. As can be seen, DSP is not able to clearly delineate the nuclear envelope (dark blue) and fails in correctly assigning the class microtubule (purple). We expect these errors to originate from a too strong learned shape bias for the microtubules.

6.2. Evolution of Decoupled Semantic Prototypes

Finally, we were interested in a deeper understanding about the semantic prototype vectors of DSP. To this end, we plotted t-SNE projections [19] of class-wise prototypes P_c alongside several few pixel-embeddings f_i after the 10th, 50th and 100th epoch. These are shown in Fig. 5. It can be seen that over the time of training, clusters are formed and the prototypes delimit which semantics occupy what regions of the embedding space.

7. FAQ – Good questions and honest answers

During the reviewing process and in internal discussions, several valid and insightful questions have been raised. We list some of them together with honest answers here, as we

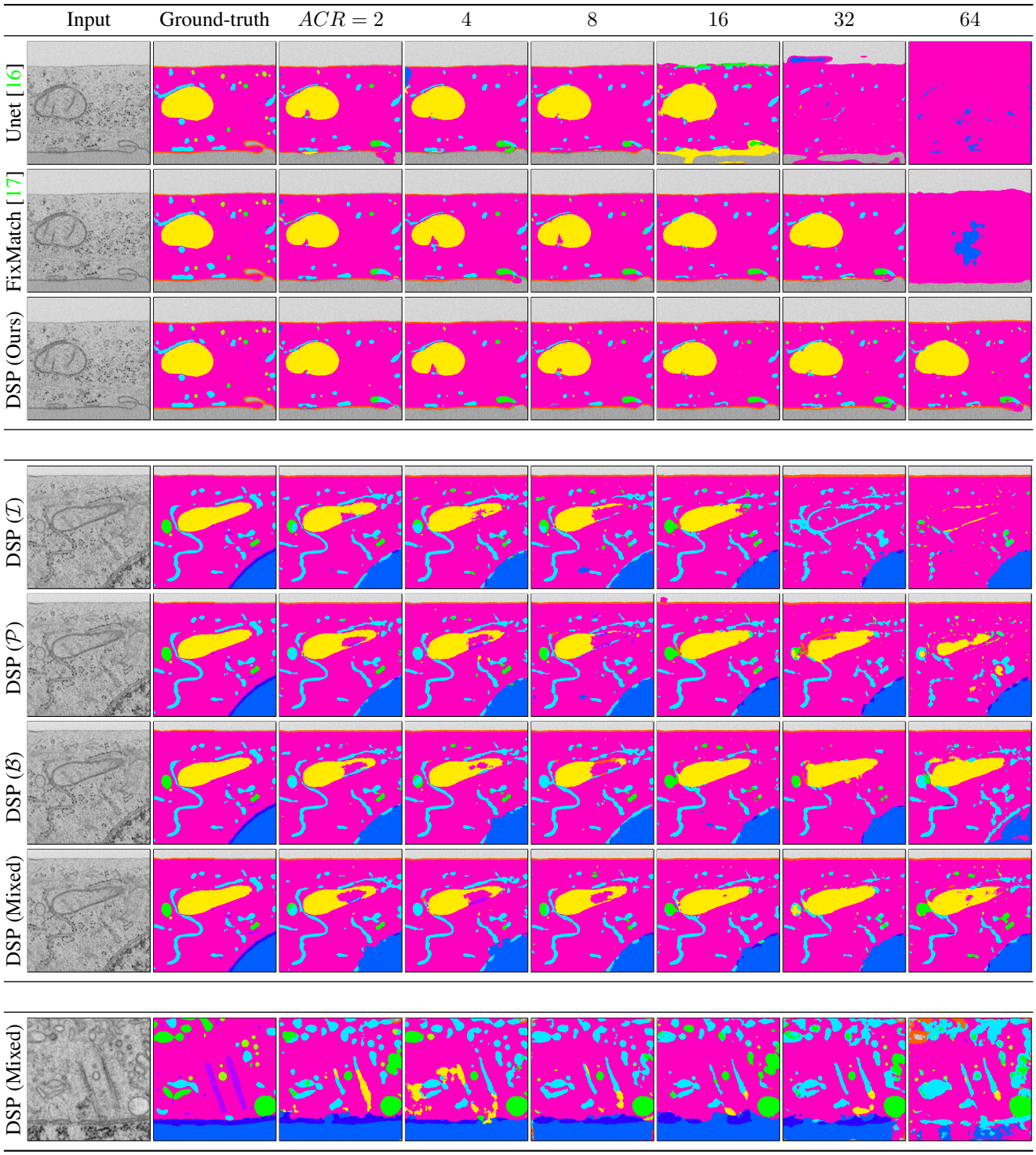

Figure 4. From left to right: test image, ground-truth segmentation of the organelle structures, and predicted segmentations from models trained with diverse annotation types and decreasing number of annotated pixel-wise masks on *HELA-2*. Top three rows: Models trained in the mixed annotation type scenario as outlined in the main paper. Middle four rows: Our method *Decoupled Semantic Prototypes* trained with different combinations of annotation types to show their effects. Bottom row: Failure case of DSP in the mixed scenario

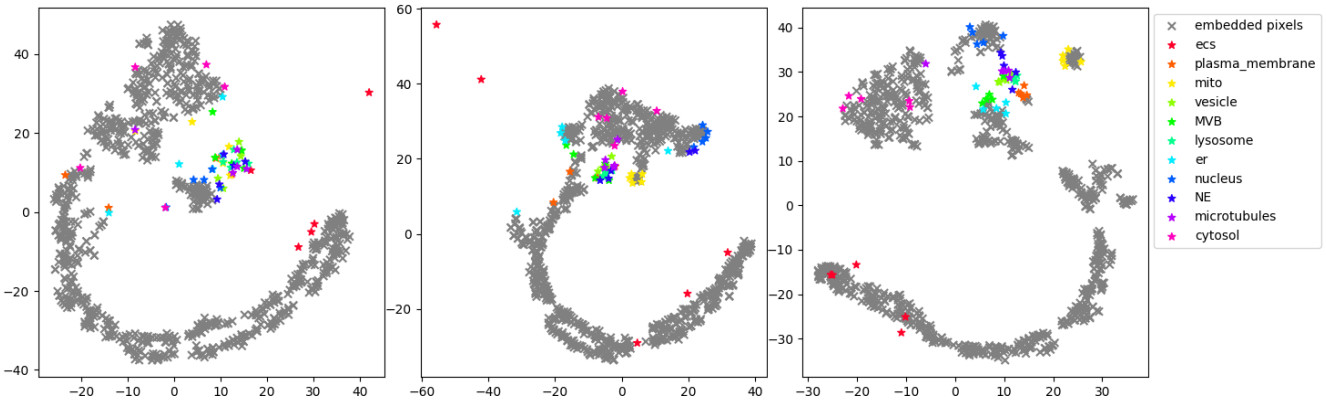

Figure 5. Three plots of the learned class-wise prototypes (colored stars) and randomly selected pixel-embeddings (gray crosses) throughout the training process (left to right: after 10, 50 and 100 epochs of training). Prototypes and embeddings are projected via t-SNE.

believe that they are relevant to a broader audience.

Q: Does L_{DSP} use unlabeled images? L_{DSP} is mainly based on annotations, specifically it unifies arbitrary annotation-types. Still, from unlabeled images, prototype associations that do not belong to the positive class can be derived. You can see this by closely observing the final loss L_{DSP} in Eq. (9): although the sum goes only over annotation types $l \in \{m, b, p, im\}$, also unlabeled images are contributing via negative associations in $L(f_i, c)$ Specifically, are included in included in computing $Z_{i,c}$ (Eq. (8): criteria $k \neq c$ is fulfilled). Furthermore, unlabeled images are also used in the filtered pseudo-labels via L_{PLF}.

Q: What is the difference of base model with a linear projection + cross-entropy compared to your DSP architecture? Our proposed prototypes are a matrix of shape $D \times C \times |P_C|$. From that point of view, a plain projection head of size $D \times C \times |P_C|$ can indeed be used to produce $1 \times C|P_C|$ scores. However, while standard projection heads rely on dot product, we exploit the cosine distance in Eq. (1) which uses normalized vectors to bound the similarity scores (as typically done in contrastive learning [22]). Furthermore, a common projection head design would directly predict C class scores, rather than predicting $C|P_C|$ scores followed by averaging the $|P_C|$ values per class as done in DSP.

With respect to results: on HELA-2, all splits, ACR=8, a plain Unet achieves $43.6 \pm 7.0\%$ (this is the special case of projection head and $|P_C| = 1$). The implementation of a projection head with $|P_C|$ scores per class, class-wise averaging, and trained with L_{CE} results in $45.1 \pm 6.6\%$. The same projection head with cosine similarity instead of dot product but still only trained with L_{CE} leads to $48.2 \pm 5.7\%$. Finally, DSP is able to integrate all annotation-types via L_{DSP} and reaches $54.2 \pm 4.6\%$.

Q: Isn't the computation for training quite expensive? In fact, all approaches in our experiments had access to the same resources available to us and are therefore comparable. Furthermore, since hardware is scalable via a higher budget, we investigated the limited resource 'expert-availability' which can't be scaled easily.

Q: Would post processing give better performance? Absolutely, yes. We clearly acknowledge that all approaches would benefit from smart post-processing rules. Here, we focused on a plain comparison of underlying training algorithms.

Q: Since ACR does not reflect varying efforts for weak annotations, is ACR a good measure for expert centric scenarios? We think it is! The idea of expert-centrism is that a segmentation training is not restricted to masks but can handle *any* preference of annotation combinations (as DSP does).

Q: Why don't you additionally compare with purely weakly supervised methods? A comparison to weakly supervised methods would give nice lower bounds but would not be a fair comparison as they don't consider mask annotations.

Q: Why don't you additionally compare on standard computer vision datasets? We agree: more datasets are always better (and we're curious too!). Yet, training the over 1600 networks on the four datasets was already a considerable effort given our available compute hardware.

References

[1] Wonho Bae, Junhyug Noh, Milad Jalali Asadabadi, and Danica J Sutherland. One weird trick to improve your semi-weakly supervised semantic segmentation model. *arXiv preprint arXiv:2205.01233*, 2022. 4

[2] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 2

[3] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1300–1309, 2022. 4

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[5] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4

[6] Chaz Firestone and Brian J Scholl. "please tap the shape, anywhere you like" shape skeletons in human vision revealed by an exceedingly simple measure. *Psychological science*, 25(2):377–386, 2014. 4

[7] Larissa Heinrich, Davis Bennett, David Ackerman, Woohyun Park, John Bogovic, Nils Eckstein, Alyson Petruncio, Jody Clements, Song Pang, C Shan Xu, et al. Whole-cell organelle segmentation in volume electron microscopy. *Nature*, 599(7883):141–146, 2021. 2, 3, 4

[8] Larissa Heinrich, Jan Funke, Constantin Pape, Juan Nunez-Iglesias, and Stephan Saalfeld. Synaptic cleft segmentation in non-isotropic volume electron microscopy of the complete drosophila brain. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 317–325. Springer, 2018. 5

[9] Nicholas Heller, Fabian Isensee, Klaus H Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. *Medical Image Analysis*, 67:101821, 2021. 2

[10] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018. 2

[11] Philipp Kainz, Martin Urschler, Samuel Schulter, Paul Wohlhart, and Vincent Lepetit. You should use regression to detect cells. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 276–283. Springer, 2015. 5

[12] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 4, 5

[13] Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of influence for transfer learning across diverse appearance domains and task types. *arXiv preprint arXiv:2103.13318*, 2021. 2

[14] Pawel Mlynarski, Hervé Delingette, Antonio Criminisi, and Nicholas Ayache. Deep learning with mixed supervision for brain tumor segmentation. *Journal of Medical Imaging*, 6(3):034002, 2019. 4, 5

[15] Simon Reiß, Constantin Seibold, Alexander Freytag, Erik Rodner, and Rainer Stiefelhagen. Graph-constrained contrastive regularization for semi-weakly volumetric segmentation. In *European Conference on Computer Vision*, pages 401–419. Springer, 2022. 4, 5

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 4, 7

[17] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 4, 5, 6, 7

[18] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. Boxinst: High-performance instance segmentation with box annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5443–5452, 2021. 5

[19] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 6

[20] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1559–1572, 2018. 5

[21] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2

[22] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*, 2021. 1, 8

[23] Shuwei Zhai, Guotai Wang, Xiangde Luo, Qiang Yue, Kang Li, and Shaoting Zhang. Pa-seg: Learning from point annotations for 3d medical image segmentation using contextual regularization and cross knowledge distillation. *arXiv preprint arXiv:2208.05669*, 2022. 5

[24] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2