DEARLi: Decoupled Enhancement of Recognition and Localization for Semi-supervised Panoptic Segmentation

Supplementary Material

A. Additional ablations and results

Increasing backbone size. Tables 9 and 10 present the performance of DEAR and DEARLi with ConvNeXt-B (CN-Base) and ConvNeXt-L (CN-Large) backbones on ADE20K and COCO-Panoptic. These experiments correspond to Figure 6 from the main manuscript, but are tabulated for easier comparison and reference for future work. Both DEAR and DEARLi benefit from increased backbone capacity. We observe this benefit across all data partitions on both datasets. Note that a frozen CN-Large backbone still enables DEAR and DEARLi to be trained on a single A100-40GB GPU.

Method	Backbone Size	1/128 (158)	1/64 (316)	1/32 (632)	1/16 (1263)	1/8 (2526)
DEAR	❄ CN-B-L2B	27.7	32.3	34.8	38.3	40.6
DEAR	❄ CN-L-L2B	28.2	34.3	36.1	40.0	42.5
DEARLi	❄ CN-B-L2B	29.9	34.6	36.3	39.0	41.6
DEARLi	❄ CN-L-L2B	31.6	35.6	39.6	41.3	43.9

Table 9. Panoptic performance (PQ) on ADE20K when increasing the backbone size. CN-B and CN-L refer to ConvNeXt-Base and ConvNeXt-Large, respectively. L2B denotes the LAION-2B pre-training dataset. ❄ represents a frozen backbone.

Method	Backbone Size	1/512 (232)	1/256 (463)	1/128 (925)	1/64 (1849)	1/32 (3697)
DEAR	❄ CN-B-L2B	34.7	38.6	40.8	43.0	44.8
DEAR	❄ CN-L-L2B	35.3	40.2	43.1	45.6	47.4
DEARLi	❄ CN-B-L2B	38.8	41.3	43.1	44.5	46.4
DEARLi	❄ CN-L-L2B	39.9	43.2	45.3	47.3	48.4

Table 10. Panoptic performance (PQ) on COCO-Panoptic when increasing the backbone size.

On training stability. Table 11 presents PQ of our method on ADE20K across three different seeds. The first row corresponds to the results of the run presented in Tab. 1 of the main manuscript. The next two rows present additional runs with the comparable performance. Overall, the experiments exhibit small variance, which demonstrates the training stability of our method.

Panoptic vs. semantic labels. Table 12 examines the impact of training labels on the semi-supervised semantic segmentation performance of our method. The first row repeats performance of DEAR from the Tab. 3 of the main manuscript. The second row presents experiments where the same model is retrained with semantic segmentation labels, which requires additional adaptation to the pseudo-labels generation procedure for the semantic segmentation objective. We observe that the models achieve comparable performance, which justifies our comparison with previous methods in Tab. 3 from the main manuscript. This demonstrates that the influence of the training labels is minimal. Indeed, the semantic segmentation labels significantly outperform the panoptic labels in the 1/32 experiment, while underperforming at 1/128 and performing roughly the same in the remaining three experiments.

Method	1/128	1/64	1/32	1/16	1/8
DEARLi	**29.9**	**34.6**	**36.3**	**39.2**	**41.6**
	30.5	34.8	36.5	39.1	42.0
	30.7	34.8	36.1	39.5	41.6
	$30.4_{\pm 0.3}$	$34.7_{\pm 0.1}$	$36.3_{\pm 0.2}$	$39.3_{\pm 0.2}$	$41.7_{\pm 0.2}$

Table 11. Panoptic performance (PQ) of our final method, DEARLi, across three different seeds for standard data partitions of ADE20K. Bolded results correspond to the runs reported in Tab. 1 of the main manuscript. The last row represents $\text{mean}_{\pm \text{std}}$.

Method	Backbone	1/128 (158)	1/64 (316)	1/32 (632)	1/16 (1263)	1/8 (2526)
DEAR	❄ CN-B-L2B	36.5	40.5	42.8	45.8	47.5
DEAR-*semseg*		35.8	40.9	44.8	45.2	47.4

Table 12. Semantic performance (mIoU) on different ADE20K data partitions. DEAR is trained with respect to the standard panoptic labels as in the main manuscript. In contrast, DEAR-*semseg* is trained with respect to the semantic segmentation labels. Notably, the reported performances are closely comparable.

Semantic segmentaiton ablations. Table 13 presents ablations of DEARLi for semantic segmentation. These models are identical to those in Tab. 1 of the main manuscript, but evaluated with standard M2F semantic segmentation inference. The trends observed are similar to those for panoptic segmentation. DEAR significantly outperforms the M2F-Lang baseline across all data partitions, while DEARLi further enhances performance in 4 out of 5 scenarios. Furthermore, we observe (*cf*. rows 1 and 2) that our M2F-Lang baseline in 3 out of 5 scenarios outperforms state-of-the-art semantic segmentation method SemiVL [22], which proves that our panoptic baseline is indeed strong.

Method		SSL	1/128	1/64	1/32	1/16	1/8
SemiVL (*cf*. Tab. 3)	❄ CN-B-L2B	✓	30.2	33.3	36.3	37.7	40.7
M2F-Lang	❄ CN-B-L2B	✓	20.9	30.1	37.4	40.7	45.4
↳ + $\mathbf{P}_{\text{ENS}}$ (**DEAR**)		✓	36.5	40.5	42.8	**45.8**	47.5
↳ + DeWa (**DEARLi**)		✓	**38.9**	**42.0**	**44.3**	45.0	**48.1**

Table 13. Semantic segmentation ablations (mIoU) on ADE20K.

Method	SSL	Backbone	1/128 (158)			1/64 (316)			1/32 (632)			1/16 (1263)			1/8 (2526)		
			mPQ	mSQ	mRQ	**mPQ**	mSQ	mRQ	**mPQ**	mSQ	mRQ	**mPQ**	mSQ	mRQ	**mPQ**	mSQ	mRQ
M2F	–	❄ CN-B-IN1k	7.4	32.7	9.3	13.3	52.2	16.7	17.8	62.7	21.7	22.1	65.1	27.0	25.5	69.3	30.7
	–	🔥 CN-B-IN1k	9.3	38.6	11.7	15.0	55.2	18.6	20.5	62.9	25.2	25.0	68.9	30.5	29.9	73.2	35.9
	✓	🔥 CN-B-IN1k	16.4	45.7	20.4	22.8	60.4	27.9	27.8	70.2	34.0	30.1	71.4	36.6	33.6	75.1	40.3
M2F	–	❄ CN-B-IN21k	9.7	37.7	12.7	14.4	53.2	17.8	20.8	64.2	25.5	25.8	67.6	31.6	30.1	72.2	36.1
	–	🔥 CN-B-IN21k	11.2	41.9	14.1	16.7	56.6	20.6	23.3	66.4	28.4	27.9	69.8	33.7	32.9	74.4	39.3
	✓	🔥 CN-B-IN21k	18.7	46.5	23.1	26.3	62.1	32.0	30.9	70.0	37.1	34.3	74.0	41.2	37.6	77.1	45.1
M2F	–	❄ CN-B-L2B	10.0	38.7	12.4	16.4	56.2	20.1	23.4	65.4	28.5	28.7	69.5	34.9	34.2	73.7	40.9
	–	🔥 CN-B-L2B	12.9	42.8	16.1	19.3	59.1	23.9	26.0	69.1	31.7	31.1	72.0	37.7	36.3	75.6	43.7
	✓	🔥 CN-B-L2B	19.4	46.2	24.0	28.4	63.7	34.6	33.2	71.3	40.2	36.8	75.9	44.3	40.2	78.8	48.2
M2F-Lang	–	❄ CN-B-L2B	11.7	43.7	14.7	19.1	61.4	23.9	26.3	70.2	32.1	31.6	72.8	38.3	36.6	76.7	44.0
↳ + $\mathbf{P}_{\text{ENS}}$ (eval only)	–		18.3	63.7	23.3	23.9	69.5	30.0	29.0	73.3	35.4	33.6	75.6	40.8	37.8	77.6	45.7
M2F-Lang	✓		18.3	55.0	22.6	26.4	68.0	32.3	32.5	76.1	39.4	35.5	74.5	42.8	39.2	77.2	47.2
↳ + $\mathbf{P}_{\text{ENS}}$ (eval only)	✓		21.6	62.6	26.7	30.2	73.9	37.0	33.9	76.2	41.1	36.7	77.6	44.2	40.2	77.2	48.4
↳ + $\mathbf{P}_{\text{ENS}}$ (**DEAR**)	✓		27.7	70.3	34.4	32.3	74.0	39.6	34.8	75.4	42.1	38.3	79.3	46.2	40.6	80.7	48.7
↳ + DeWa (**DEARLi**)	✓		**29.9**	74.0	36.6	**34.6**	75.4	41.2	**36.3**	77.1	43.5	**39.2**	80.3	47.1	**41.6**	80.6	49.5
M2F-Lang	✓	❄ CN-B-L2B	18.3	55.0	22.6	26.4	68.0	32.3	32.5	76.1	39.4	35.5	74.5	42.8	39.2	77.2	47.2
↳ + $\mathbf{P}_{\text{ENS}}$ (teacher only)	✓		27.3	68.7	34.0	31.5	72.9	38.6	34.4	74.5	41.5	38.0	78.2	45.8	40.3	79.2	48.4
M2F-Lang	✓		18.3	55.0	22.6	26.4	68.0	32.3	32.5	76.1	39.4	35.5	74.5	42.8	39.2	77.2	47.2
↳ + DeWa	✓		20.5	55.6	25.1	29.1	71.1	35.0	33.9	75.6	40.6	36.9	75.3	44.3	40.8	78.3	48.6

Table 14. Extension of Tab. 1 from the main manuscript. The new experiments are shown in gray. Pre-training on ImageNet-21k performs in between pre-training on ImageNet-1k and LAION-2B [57]. Ensembling with zero-shot CLIP helps more when applied during training within the teacher (pseudo-labels generation). Decoder warm-up (DeWa) contributes with and without zero-shot CLIP ensembling.

Additional panoptic ablations. Table 14 extends Table 1 from the main manuscript. The second section shows experiments with ImageNet-21k [54] backbone initialization. We observe improvements over the ImageNet-1k initialization (*cf*. first section) in all setups across all data regimes. Nevertheless, pre-training on LAION-2B [57] still prevails (*cf*. third section). This confirms the benefits of contrastive language-image pretraining with respect to the traditional pre-training on categorical image-wide labels. The last section presents additional ablations of our contributions. The second row evaluates DEAR without ensembling with zero-shot CLIP features during inference, using ensembling only in the teacher to enhance pseudo-labels. The experiments reveal that inference-time ensembling brings slight improvement of 0.3-0.8 PQ points, depending on the data partition. The last row shows the contribution of the decoder warm-up (DeWa) to the Mean Teacher baseline. We observe that DeWa consistently improves the performance by 1.4-2.7 PQ points. However, it still performs significantly worse than DEARLi. These results highlight the complementary nature of the proposed contributions.

B. SemiVL with a ConvNeXt Backbone

We first successfully reproduced SemiVL [22] experiments with ViT. Then, we reconfigure SemiVL with a ConvNeXt-B backbone by introducing several straightforward modifications to the published source code. First, new language embeddings are generated using the appropriate text encoder from OpenCLIP [10]. Second, the upsampling component of SemiVL is adjusted to align with the feature dimensions of ConvNeXt at each stage. An extra skip connection and upsampling block are added to accommodate the hierarchical structure of convolutional backbones. All other implementation details remain consistent with the ViT-B/16 setup.

SemiVL [22] only fine-tunes the attention weights of the ViT backbone. Since ConvNeXt lacks a directly analogous mechanism, we freeze the backbone to ensure a direct comparison with our approach. We also conduct experiments with a fully fine-tuned ConvNeXt backbone, using the same hyperparameters as described in the original paper. In Tables 3 and 5 (main manuscript), we report experiments with a frozen backbone, while Tables 15 and 16 include additional experiment with a fine-tuned backbone.

Method	Net	1/128 (158)	1/64 (316)	1/32 (632)	1/16 (1263)	1/8 (2526)
SemiVL [22] [ECCV'24]	🔥 ViT-B/16	28.1	33.7	35.1	37.2	39.4
SemiVL† [22] [ECCV'24]	❄ CN-B-L2B	30.2	33.3	36.3	37.7	40.7
SemiVL† [22] [ECCV'24]	🔥 CN-B-L2B	29.5	33.8	36.1	38.3	40.0
DEAR	❄ CN-B-L2B	36.5	40.5	42.8	**45.8**	47.5
DEARLi	❄ CN-B-L2B	**38.9**	**42.0**	**44.3**	45.0	**48.1**

Table 15. Comparison with the state of the art in semi-supervised semantic segmentation on **ADE20K**. † indicates our experiments with public source code. Underline denotes CLIP-WiT [52] initialization. 🔥 denotes partial fine-tuning (attention weights only).

Method	Net	1/512 (232)	1/256 (463)	1/128 (925)	1/64 (1849)	1/32 (3697)
SemiVL [22] [ECCV'24]	🔥 ViT-B/16	50.1	52.8	53.6	55.4	56.5
SemiVL† [22] [ECCV'24]	❄ CN-B-L2B	47.6	49.1	50.1	52.6	52.9
SemiVL† [22] [ECCV'24]	🔥 CN-B-L2B	47.2	47.3	50.0	51.4	52.6
DEAR	❄ CN-B-L2B	52.9	53.9	56.2	58.7	59.3
DEARLi	❄ CN-B-L2B	**54.6**	**55.1**	**57.0**	**59.1**	**60.2**

Table 16. Comparison with the state of the art in semi-supervised semantic segmentation on **COCO-Objects**. † indicates our experiments with public source code. Underline denotes CLIP-WiT [52] initialization. 🔥 denotes partial fine-tuning.

These results suggest that SemiVL [22] gains no benefit from fine-tuning the ConvNeXt backbone. Note that SemiVL [22] reports results from the best checkpoint on the validation set. In our SemiVL reproduction, we also report results from the best epoch. As outlined in the main manuscript, all results for DEAR and DEARLi are obtained by evaluating the checkpoint from the final training iteration.

C. Limitations

The hyperparameter α for geometric ensembling is manually set to 0.6, which may be suboptimal for different data partitions. From Tab. 8 in the main manuscript, we observe that higher values of α increase performance when more labeled data is available. This suggests that making α adaptive based on the quantity of labeled data could be beneficial. Additionally, it may be advantageous to dynamically adjust α during training (e.g., starting with a lower value early on and increasing it in later stages). These challenges present opportunities for future research.

D. Further Implementation Details

This section presents implementation details of image perturbations that are presented in Section 4 of the main manuscript. For color jittering, we use `torchvision` implementation with the following parameters: `brightness:(0.2, 1.8)`, `saturation:(0.2, 1.8)`,`contrast:(0.2, 1.8)`,and `hue:(-0.2, 0.2)`. Gaussian blur (`sigma:(0.1, 2.0)`) and CutMix [58, 78] are applied with a probability of 0.5, and grayscaling with a probability 0.2.

The number of mask queries in Mask2Former [9] is fixed at 200 across all experiments. Models trained without unlabeled data (supervised) begin with 10k iterations for the smallest data regime (*i.e.*, 1/128 for ADE20K and 1/512 for COCO), with an additional 10k iterations added for each subsequent regime. The batch size in these experiments is 8. For the decoder warm-up stage, we generate class-agnostic pseudo-labels using ViT-Huge SAM [32].

E. Panoptic Segmentation Examples

Figures 7 and 8 show panoptic predictions of our models on ADE20K and COCO-Panoptic validation images, respectively. The models are trained on the most challenging data partitions with the least amount of labeled images. The first two columns display the input image and the corresponding ground truth. The next three columns present overlaid panoptic predictions of the Mean Teacher baseline (M2F+SSL), DEAR and DEARLi. Comparing DEAR with the baseline reveals that the recognition enhancement often rectifies classification errors (e.g., *building* to *wall* in the 2nd row of Fig. 7, or *suitcase* to *handbag* in the 6th row of Fig. 8). Additionally, a comparison between DEAR and DEARLi highlights how the localization enhancement refines segmentation boundaries (e.g., *chair* legs in the third row of Fig. 7). The last row of Fig. 7 shows an interesting failure mode where our models isolate parts of the *house* such as *door* and *windows* into separate segments. While this decision is not entirely wrong, it deviates from the dataset's labeling policy which assigns these classes to indoor scenes only. This illustrates the limitations of CLIP zero-shot classification, as class text embeddings diverge from the labeling policy.

F. Comparison with the State of the Art

Figures 9 and 10 compare semantic segmentation predictions of our method DEAR with the state-of-the-art method SemiVL [22] on ADE20K and COCO-Objects, respectively. We consider models trained in most challenging data partitions with the least amount of labeled data. The columns show the input image, ground truth, and predictions of SemiVL [22] and DEAR. We observe that SemiVL occasionally misclassifies correctly segmented objects (e.g. the third and fifth row in Fig. 9). Furthermore, it sometimes splits a single object into two different classes (e.g., the fourth row in Fig. 9 and the fifth row in Fig. 10). In contrast, our model produces fewer misclassifications and more cohesive segmentations. We argue this is due to the mask transformer framework, which enables zero-shot CLIP classification at the mask level and delivers more consistent predictions than in the SemiVL's patch-level classification approach.

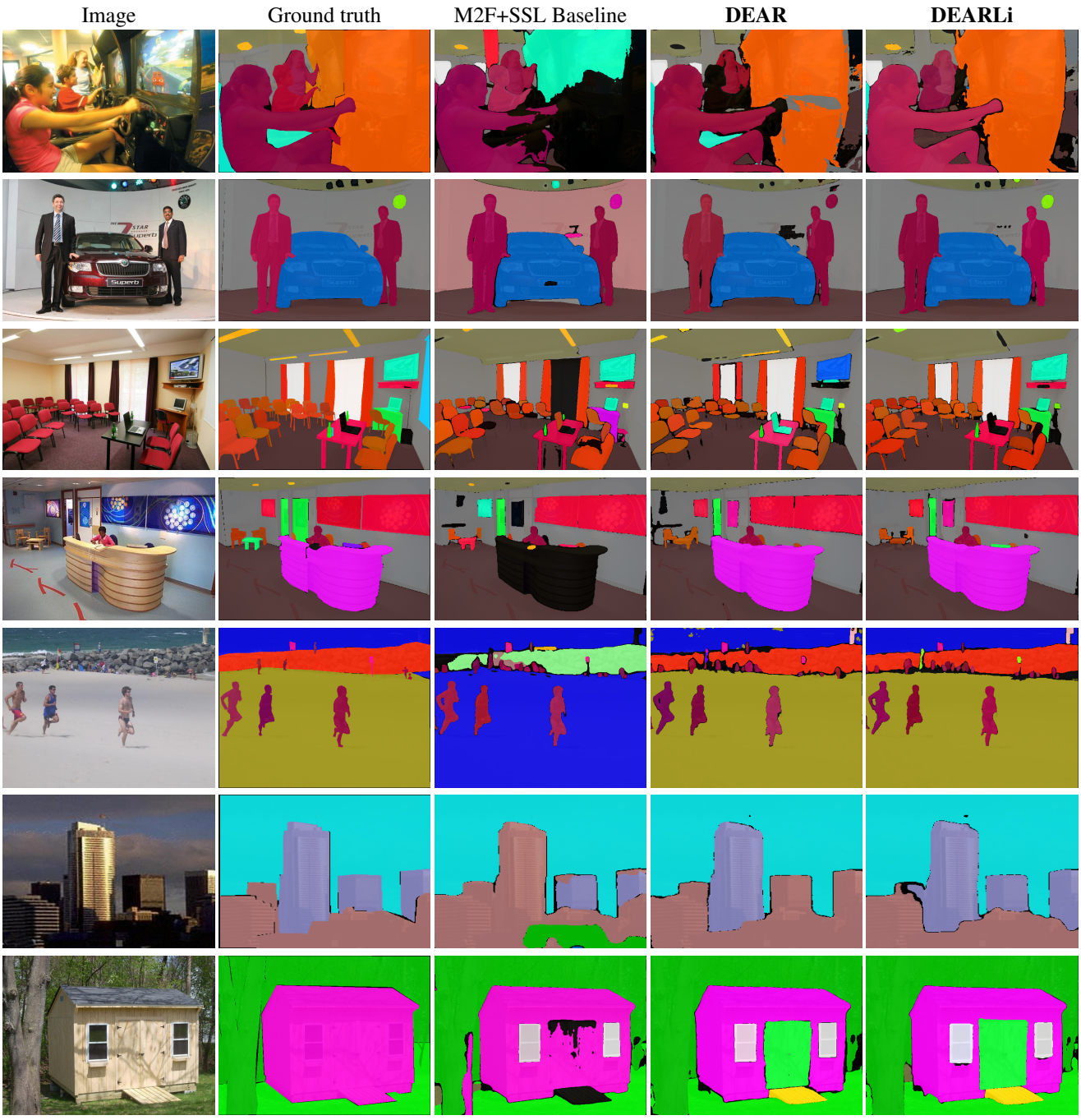

Figure 7. Panoptic predictions of the baseline M2F+SSL CN-B-L2B (*cf*. Tab. 1 in the main manuscript), DEAR, and DEARLi on few examples from the ADE20K validation set. All models are trained on the ADE20K 1/128 data partition (*i.e.*, only 158 labeled images).

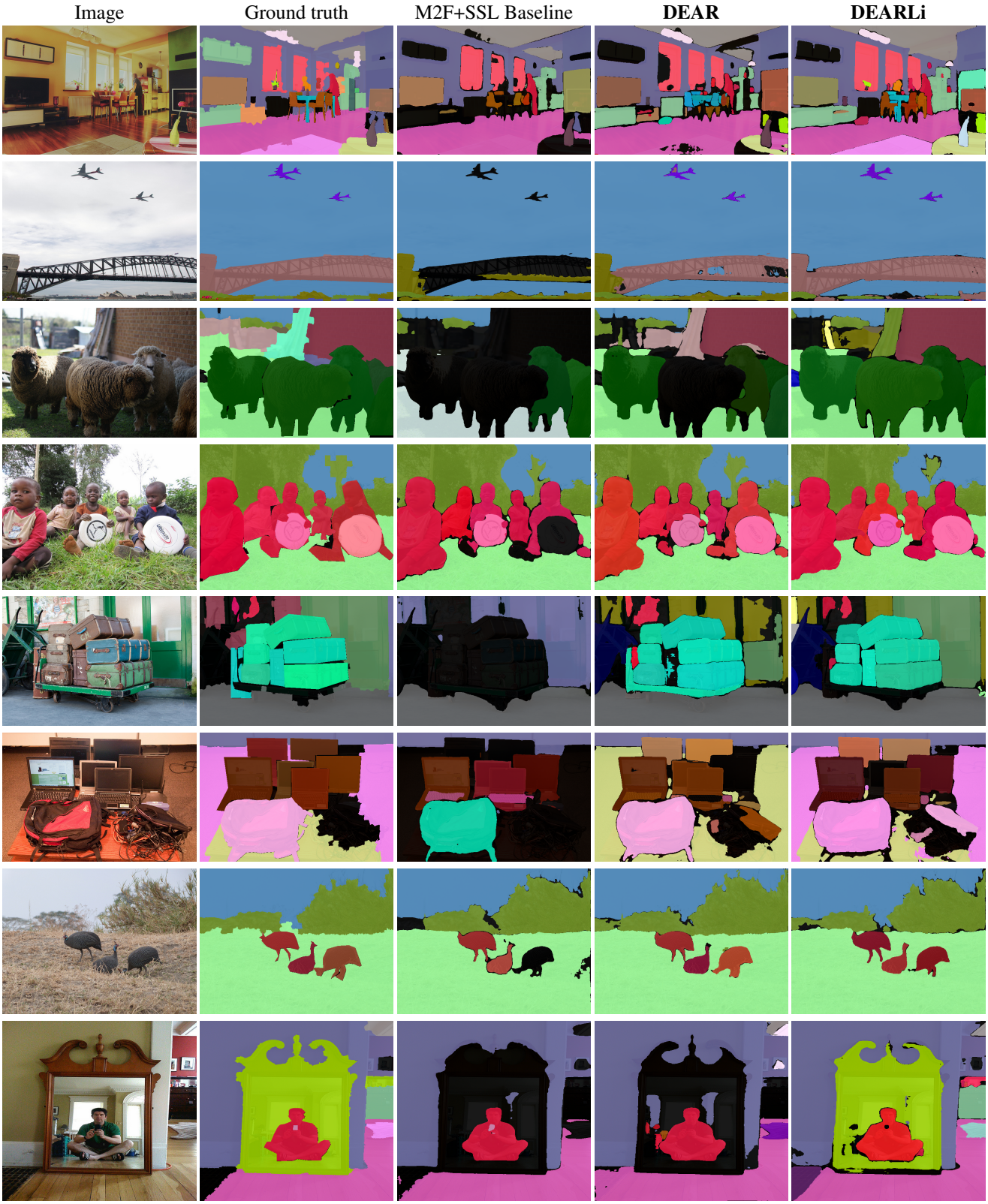

Figure 8. Panoptic predictions of the baseline M2F+SSL CN-B-L2B (*cf*. Tab. 1 in the main manuscript), DEAR, and DEARLi on few examples from the COCO-Panoptic val. All models are trained on the COCO-Panoptic 1/512 data partition (*i.e.*, only 232 labeled images).

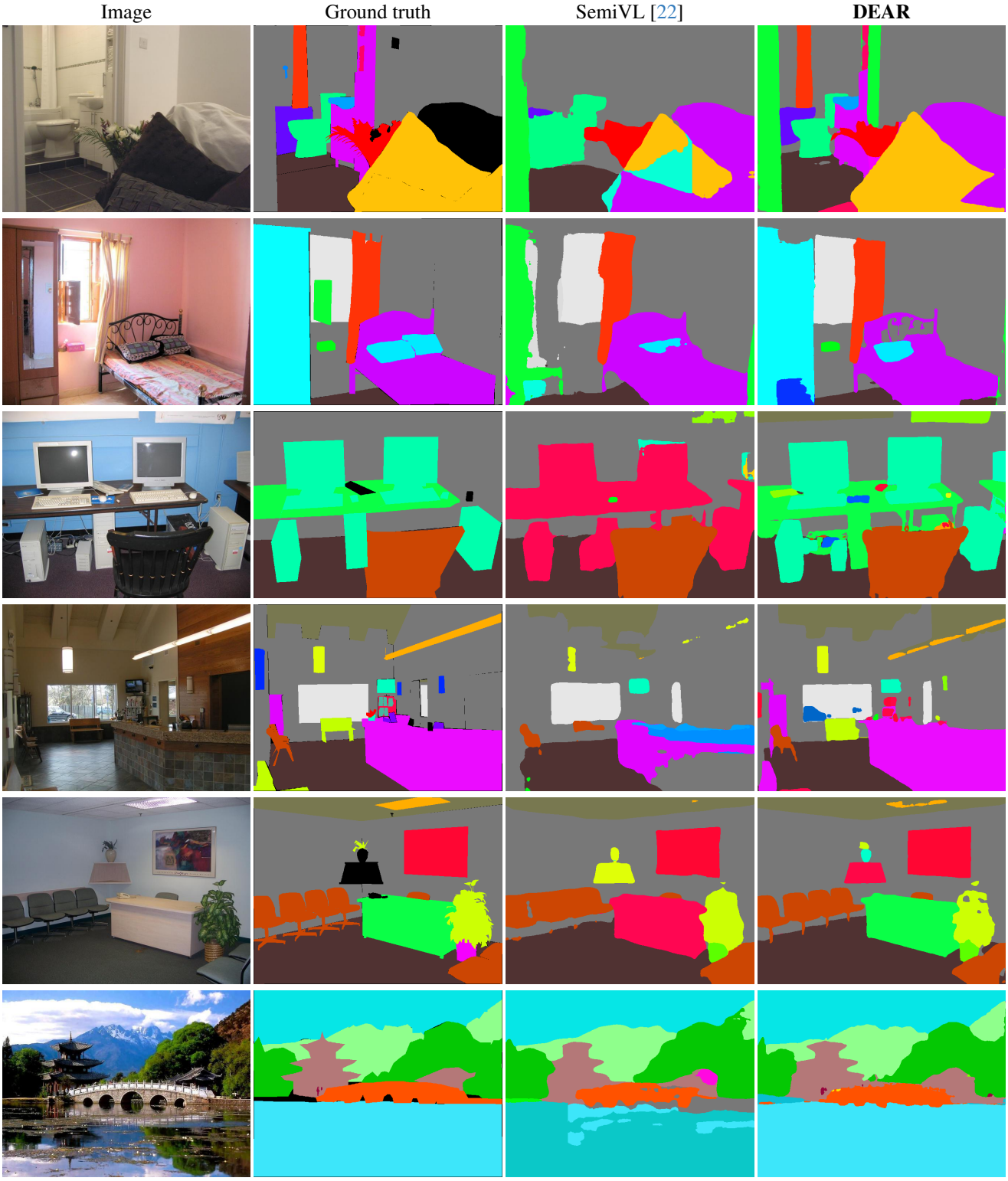

Figure 9. Qualitative comparison of DEAR with the state-of-the-art method SemiVL [22] for semantic segmentation. Models are trained on the ADE20K 1/128 partition (*i.e.*, 158 labeled images), with predictions visualized on examples from the ADE20K validation set. Predictions for SemiVL [22] are generated using the publicly released checkpoint.

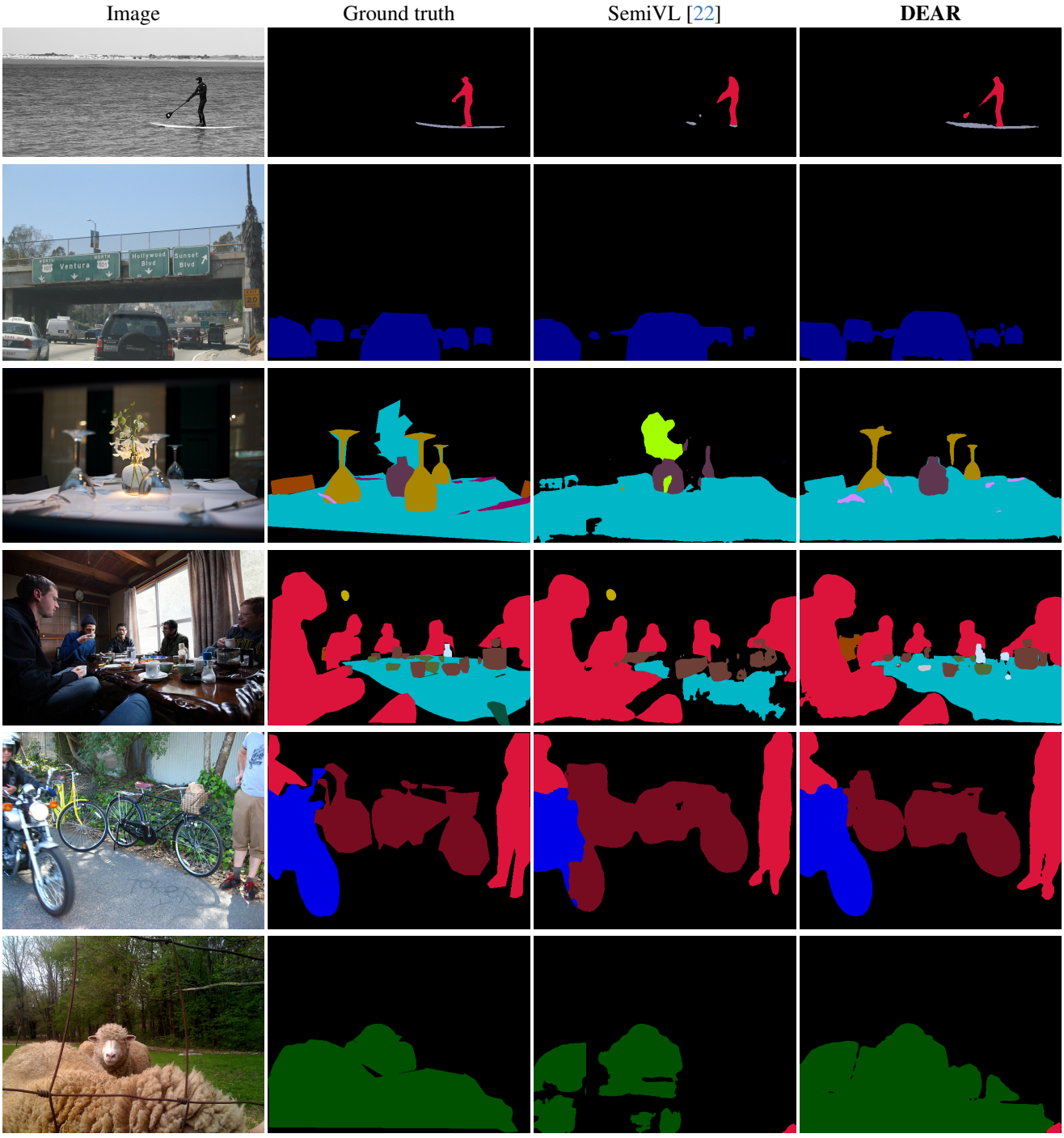

Figure 10. Qualitative comparison of DEAR with the state-of-the-art method SemiVL [22] for semantic segmentation. Models are trained on the COCO-Objects 1/512 partition (*i.e.*, 232 labeled images), with predictions visualized on examples from the COCO-Objects validation subset. In COCO-Objects, the *background* class is represented in black and included in the evaluation. Predictions for SemiVL [22] are generated using the publicly released checkpoint.