A. Implementation details

During post-training, we fine-tune only the last three transformer blocks of the pretrained ViT encoder $f(\cdot)$ while keeping the remaining layers frozen. Our MLP projector $h(\cdot)$ consists of two linear layers with a non-linear activation function GELU [35]. Hidden feature dimension is set to $7D$ and fixed output dimension 6144. ℓ_2-normalization is applied at the output of the MLP. We set the temperature parameter τ of the softmax operator to 0.07 (see Tab. 7). Our ablation study demonstrates that the method is robust to the choice of temperature, with values ranging from 0.03 to 0.09 yielding similar performance on the PascalVOC dataset. During post-training our support set consists of 1 positive and 7 "distractor" examples (8 total examples). During training, we use the AdamW [49] optimizer with a learning rate of 2.25×10^{-7} and a weight decay of 0.05. We train for 5 epochs. We employ cosine learning rate schedule.

Temperature	0.09	0.07	0.03
PascalVOC	80.8	81.0	81.0

Table 7. Ablation of temperature τ during post-training.

Generation time of pseudo semantic segmentation labels. COCO pseudo-label generation required 17 hours on 1 V100 GPU, with DiffCut inference as the bottleneck due to its current lack of batch processing optimization and suboptimal GPU utilization. However, this one-time offline process can be used to post-train multiple encoders.

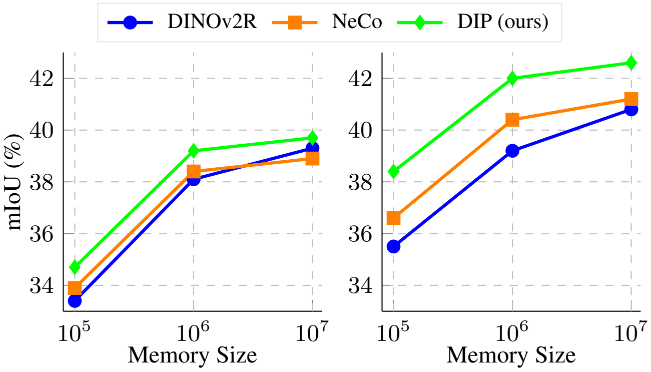
B. Additional quantitative results

In-context scene understanding: impact of memory size. In Fig. 6, we analyze the effect of memory size on in-context semantic segmentation using the ADE20K dataset (full) for DINOv2R, NeCo, and our DIP. Results show that DIP outperforms both DINOv2R and NeCo across all memory sizes.

Linear segmentation. Tab. 8 presents linear segmentation results on COCO and ADE20K benchmarks. For fair comparison, we re-evaluated both DINOv2R and NeCo using our implementation, ensuring consistent evaluation protocols across all methods. Our approach consistently improves over the strong DINOv2R baseline and shows improvements over NeCo. Notably, with the ViT-B/14 backbone on COCO, our method achieves 86.7 mIoU, surpassing DINOv2R by 2.0 points.

Method	Backbone	COCO	ADE20K
DINOv2R	ViT-S/14	82.1	33.5
NeCo	ViT-S/14	81.1 (−1.0)	33.1 (−0.4)
DIP (ours)	ViT-S/14	**82.6** (**+0.5**)	**33.7** (**+0.2**)
DINOv2R	ViT-B/14	85.5	38.6
NeCo	ViT-B/14	85.2 (−0.3)	**39.5** (**+0.9**)
DIP (ours)	ViT-B/14	**86.7** (**+2.0**)	**39.5** (**+0.9**)

Table 8. **Linear segmentation** results on COCO and ADE20K datasets. All methods (DINOv2R, NeCo, and DIP) are evaluated using our implementation. DIP consistently improves over our base model DINOv2R and outperforms NeCo across both datasets.



(a) ViT-S/14

(a) ViT-B/14

Figure 6. **In-context scene understanding: impact of memory size.** Semantic segmentation performance on ADE20K (full dataset) using dense nearest neighbor retrieval, evaluated across varying memory sizes.

Method	**Backbone**	**PascalVOC**			
		1	$\frac{1}{8}$	$\frac{1}{64}$	$\frac{1}{128}$
DINOv2R	ViT-B	79.0	75.3	67.6	60.3
DIP (ours)	ViT-B	82.1	**79.6**	**75.1**	**70.1**
DINOv2R	ViT-L	76.9	72.8	61.4	54.4
DIP (ours)	ViT-L	**81.1**	**78.7**	**70.0**	**64.6**

Table 9. **Larger backbones evaluation** We show performance of DIP and DINOv2R on a larger backbone ViT-L. Dense nearest neighbor retrieval performance (mIoU) for semantic segmentation on PascalVOC across varying proportions of training data.

Larger backbones. While ViT-L (DINOv2) underperforms ViT-B in in-context segmentation [3, 57], our method still improves results with ViT-L, as shown in Tab. 9. This demonstrates DIP's scalability across backbone sizes.

Nearest Neighbors (NN) vs. Two Crops We compare two strategies for creating positive examples: (1) retrieving nearest neighbors using DINOv2R image-wise features (NN) and (2) using two random crops from the same im-

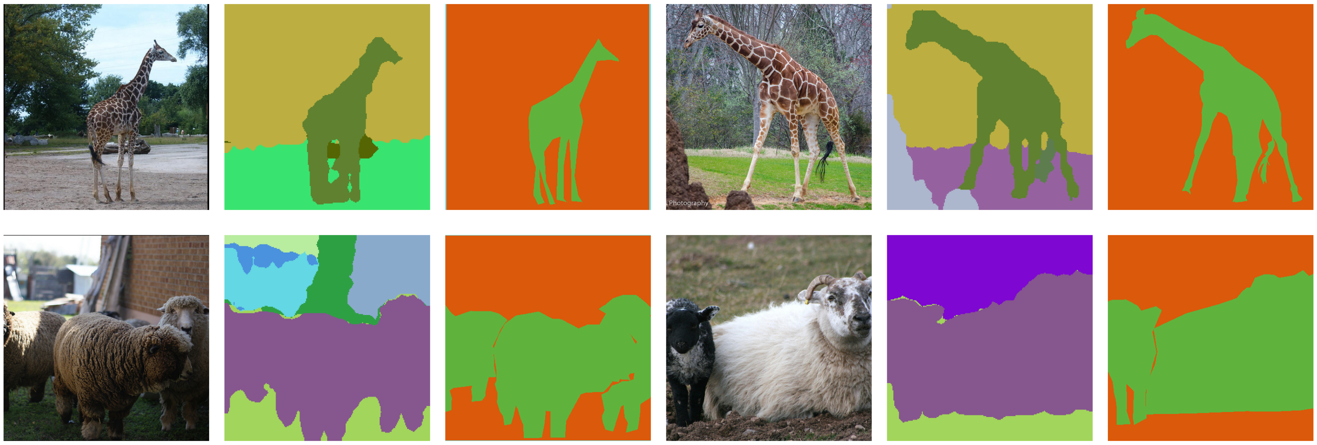
	PascalVOC				**ADE20K**			
	1	$\frac{1}{8}$	$\frac{1}{64}$	$\frac{1}{128}$	1	$\frac{1}{8}$	$\frac{1}{64}$	$\frac{1}{128}$
Two Crops	79.5	75.1	67.7	61.2	39.4	33.2	24.7	22.4
NN	**81.0**	**77.7**	**71.4**	**65.9**	**39.7**	**33.7**	**25.6**	**23.2**

Table 10. **Additional ablation on the construction of positive examples**. Dense nearest neighbor retrieval performance (mIoU) for semantic segmentation on PascalVOC and ADE20K across varying proportions of training data.

age. NN consistently outperforms Two Crops, with the performance gap widening when fewer training examples are available (see Tab. 10). This scalability advantage justifies our design choice.

C. Additional qualitative results

We present additional examples of automatically constructed in-context tasks in Fig. 7, showing the quality of our pseudo-labeling approach. We display query images paired with their corresponding positive support examples, along with both pseudo-labels and ground truth labels, included only for comparison.



(a) Query image (b) Pseudo labels (c) GT labels (d) Positive image (e) Pseudo labels (f) GT labels

Figure 7. **Examples of automatically constructed in-context scene understanding tasks.** Each row shows a query image and its corresponding positive support example. (a) and (b) display the query image and its pseudo segmentation labels, while (d) and (e) show the positive support image and its pseudo segmentation labels. (c) and (f) present the ground truth segmentation labels for the query and positive images, respectively, included only for comparison with the pseudo labels.