CODI: Improving Self-Supervised Vision Transformers with Cross Distillation

Siran Dai[1,2] **Qianqian Xu**[3,4*] **Peisong Wen**[5]
Yang Liu[5] **Xiaochun Cao**[6] **Qingming Huang**[5,3*]
[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] State Key Laboratory of AI Safety, Institute of Computing Technology, CAS
[4] Beijing Academy of Artificial Intelligence
[5] School of Computer Science and Tech., University of Chinese Academy of Sciences
[6] School of Cyber Science and Technology, Shenzhen Campus, Sun Yat-sen University
daisiran@iie.ac.cn xuqianqian@ict.ac.cn wenpeisong@ucas.ac.cn
liuyang232@mails.ucas.ac.cn caoxiaochun@mail.sysu.edu.cn qmhuang@ucas.ac.cn

Abstract

Self-supervised Vision Transformers learn strong image-level representations, but their patch-level representations can be unreliable for dense prediction. We study this mismatch and find that the class token concentrates global semantics on a small number of representative patches, while dense prediction errors often occur on low-attention regions. We provide a local gradient analysis showing that, along the [CLS] attention readout, image-level self-distillation gradients are weighted by [CLS]-to-patch attention mass. To provide masked patches with global semantics without changing the architecture, we propose Cross Distillation (CODI), which mixes a small amount of the teacher class-token distribution into each teacher patch-token target. CODI gives every masked patch an attention-independent semantic correction while preserving the standard self-distillation architecture. Experiments on dense and image-level benchmarks show that CODI improves both dense and image-level performance under aligned pretraining settings. Code is available at https://github.com/EldercatSAM/CODI.

1 Introduction

A general-purpose self-supervised Vision Transformer (ViT) should support both image-level recognition and dense prediction. These two goals rely on different representations: image-level recognition mainly uses the class token, whereas dense prediction depends directly on patch tokens. Therefore, ideal approaches should not only learn global semantics in the class token but also pass these semantics to patch tokens. Recent teacher-student self-distillation methods [3, 48, 27] have achieved great success by training a student with teacher-provided targets under both image-level discriminative and masked patch-level reconstruction losses. Despite this success, recent works [10, 33] observe that this semantic delivery is imperfect: during DINOv2-style pretraining, dense transfer performance can degrade in the later stage, indicating that patch-level representations do not always improve together with global semantic tokens.

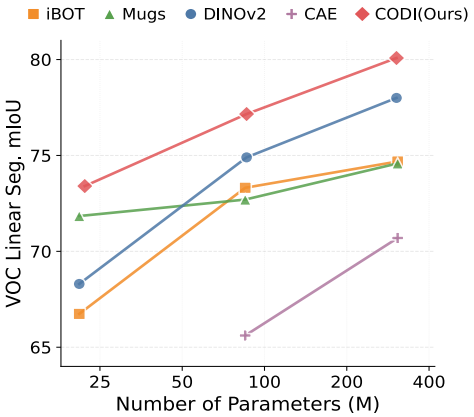


Figure 1: Comparison of CODI with baselines on Pascal VOC segmentation.

*Corresponding authors: Qianqian Xu and Qingming Huang.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

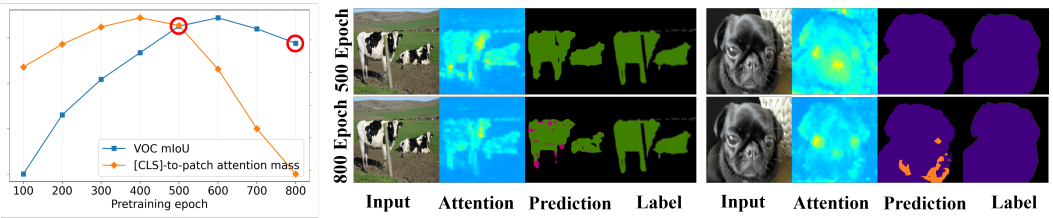


Figure 2: **Left:** During DINOv2 pretraining, downstream dense performance and [CLS]-to-patch attention mass follow similar trends. **Right:** The [CLS] token often focuses on representative patches, while prediction errors are more likely to occur on non-representative patches.

Since image-level self-distillation is optimized through the [CLS] token, we ask: *why does a stronger [CLS] representation not automatically lead to stronger patch-level representations?* To diagnose this mismatch, we examine how the class-token readout couples to patch tokens during DINOv2-style pretraining. As shown in Fig. 2, the [CLS]-to-patch attention mass decreases in the later stage of training and follows a trend similar to dense transfer performance. The attention pattern is also highly selective: the [CLS] token repeatedly focuses on a small number of visually discriminative regions, such as the eyes or mouths of animals. We call these high-[CLS]-attention regions representative patches, meaning that they serve as representative evidence for the image-level readout.

This selectivity is natural for image-level recognition, where a few discriminative patches can be sufficient for a confident prediction. Dense prediction, however, evaluates patch tokens at every spatial location. A patch that is not selected by the global readout can still be important for segmentation or depth estimation. Therefore, the relevant issue is not whether the model attends to an entire object, but whether semantic information is reliably available beyond the sparse evidence used by the [CLS] token. Consistent with this view, we observe that downstream errors are concentrated on low-attention, non-representative patches.

Why does self-distillation produce such an uneven allocation? We identify an attention-gated semantic path that helps explain this behavior. Along the [CLS] readout in ViT, image-level self-distillation sends semantic gradients to patch tokens in proportion to their [CLS]-to-patch attention mass. Through this path, high-attention representative patches receive stronger semantic updates, while low-attention patches receive weaker updates. This bias is well aligned with image-level recognition, but can leave many patch tokens less reliable for dense prediction.

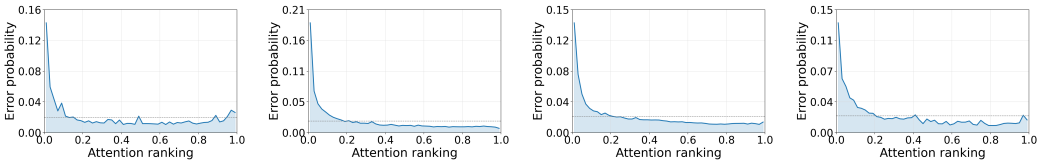
This analysis suggests that the teacher already contains useful global semantic information, but the standard training objective delivers this information to patch tokens unevenly through the [CLS] attention path. Based on this insight, we propose Cross Distillation (CODI), a target-side modification to masked patch distillation. For each masked patch, CODI mixes a small amount of the teacher class-token distribution into the teacher patch-token target. The student still learns from a primarily local patch target, but each masked patch also receives a weak global semantic correction that is not explicitly gated by its [CLS] attention mass. In this way, CODI improves the semantic reliability of low-attention patches while preserving the spatial specificity of patch supervision.

Empirically, we show that across seven evaluation settings spanning semantic segmentation, depth estimation, video object segmentation, and image classification, CODI *outperforms* its DINOv2 [27] counterpart when the pretraining setting is aligned. As shown in Fig. 1, CODI's performance scales with model size. These results reveal the potential of CODI as a more efficient training paradigm.

2 Related Work

2.1 Self-supervised Vision Transformers

Self-supervised Vision Transformers [12, 37, 3] mainly follow two paradigms: **1) Contrastive methods** use view discrimination or cross-view consistency [5, 8, 3]. **2) Prediction-based methods** predict masked content in pixel [18, 43, 6] or latent space [1, 48, 27]. Among these methods, iBOT [48] uses an online tokenizer to provide self-distillation targets for both class tokens and masked patch tokens, while DINOv2 [27] scales this framework using stronger training strategies and larger-scale data.



(a) Bird (b) Bus (c) Dog (d) Train

Figure 3: The model tends to make prediction errors on patches with low attention scores.

2.2 Self-supervised Dense Representation Learning

Despite progress in image-level SSL [4, 17, 15, 2, 46], dense prediction requires spatially meaningful local semantics [34, 44, 45, 36, 39, 30, 31, 26, 40]. Approaches include local feature alignment [38, 22], object-part grouping [50, 41], and masked prediction [18, 24, 21].

Longer self-supervised training can also degrade dense representations. DINOv3 [33] restores dense structure through Gram anchoring, while DSE [10] provides structural measures for model selection and regularization. CODI instead modifies patch targets during pretraining to address the representative-patch issue without an additional fine-tuning stage. Section 5.3 compares CODI with Gram anchoring under aligned settings.

3 The Representative Patch Phenomenon

3.1 Empirical Observation

To understand the root cause of the self-supervised dense degradation (SDD) phenomenon, we first empirically study the failure modes of existing baselines. Specifically, we train DINOv2 [27] on ImageNet-1K [19] from scratch for 800 epochs, and evaluate the final checkpoint on Pascal VOC [13]. We adopt the linear semantic segmentation setting, where the pretrained model is frozen and a linear head is trained to make pixel-level predictions based on patch representations.

Fig. 2 visualizes the `[CLS]`-to-patch attention maps and the predicted pixel labels. We observe two patterns: 1) Models tend to assign high attention to several key patterns, such as the eyes and mouth of a cat, rather than to the whole object. We refer to these key patterns as `Representative Patches`; in quantitative analyses, we define them as the top-$r\%$ image patches ranked by the final-layer `[CLS]`-to-patch attention mass averaged over heads, excluding register tokens. Similar discriminative regions frequently receive high attention across different instances. 2) Prediction errors usually occur on non-representative patches, such as the wheels of a bus. To determine whether this phenomenon arises from an observation error or a systematic bias, we bin image patches by normalized `[CLS]`-to-patch attention mass and compute the segmentation error rate within each bin. As shown in Fig. 3, we observe a monotonic trend between the error rate and the received global attention score. Since dense prediction tasks, such as semantic segmentation, require every patch, not only the representative patches, to be semantically meaningful, this observation motivates us to further study the cause of this phenomenon and address it.

3.2 Theoretical Justification

The observation above raises a natural question: if DINO/iBOT already includes image-level self-distillation, why do some object patches still become unreliable for dense prediction? Our explanation is that the strong global target is optimized through the `[CLS]` token. For a patch to benefit from this target through the `[CLS]` branch, it must first be selected by `[CLS]` attention. Therefore, the same softmax mechanism that helps the model identify representative evidence can also make the global target weak for low-attention patches.

Local `[CLS]` attention model. This subsection analyzes a one-step local `[CLS]` readout under the image-level class-token loss only. The masked patch loss is not part of this local formulation; it may provide additional patch-token gradients, while our claim concerns the semantic gradient transmitted through the `[CLS]` attention path. Let P denote image patch tokens and let T denote all tokens competing in the final `[CLS]` attention row, including register tokens when used. For token

$i \in T$, let k_i and v_i be its key and value. The local [CLS] readout is

$$s_i = \frac{a_{\text{cls}}^\top k_i}{\sqrt{d}}, \qquad \pi_i = \frac{\exp(s_i)}{\sum_{j\in T}\exp(s_j)}, \qquad z = \sum_{j\in T}\pi_j v_j. \tag{1}$$

For patch-level analysis we consider $i \in P$. We define the total [CLS]-to-patch mass as

$$\Pi_P = \sum_{i\in P}\pi_i,$$

and rank patches within an image by the normalized patch attention $\bar{\pi}_i = \pi_i/\Pi_P$. This abstraction treats upstream representations and other network paths as fixed, and studies the exact gradient allocation along this softmax-gated attention path.

We treat the teacher class-token target q_{cls}^t as fixed in this local calculation, and define

$$\mathcal{L}_{\text{cls}} = \text{CE}\left(q_{\text{cls}}^t, \text{softmax}(\phi_s(z)/\tau_s)\right), \qquad g := \nabla_z \mathcal{L}_{\text{cls}}. \tag{2}$$

The construction of q_{cls}^t is given in Sec. 4.1.

Theorem 1 ([CLS] attention gates semantic gradients along the value and score paths)**.** *For any patch token $i \in P$, the local readout in Eq. (1) sends a value-path gradient and a score update of the form*

$$\nabla_{v_i}\mathcal{L}_{\text{cls}} = \pi_i g, \qquad -\frac{\partial \mathcal{L}_{\text{cls}}}{\partial s_i} = \pi_i u_i, \qquad u_i := -\langle g, v_i - z\rangle, \tag{3}$$

where g denotes the global semantic gradient back-propagated from the teacher [CLS] target.

Remark 2. *The scalar u_i measures the local semantic utility of patch i: if $u_i > 0$, increasing s_i decreases the loss locally. The value-path gradient norm is exactly $\|\nabla_{v_i}\mathcal{L}_{\text{cls}}\| = \pi_i\|g\|$, and the score update also contains the multiplicative factor π_i. Thus, through this [CLS] readout path, low-attention patches are weakly coupled to the image-level semantic target.*

Theorem 1 identifies one mechanism by which representative patches can dominate the image-level semantic pathway. Classification benefits from sparse, discriminative evidence; dense prediction instead requires semantic features on many non-representative object regions. Consequently, longer self-distillation training can strengthen the [CLS] representation while leaving low-attention patches with weaker semantic guidance through this path, which matches our observation that segmentation errors concentrate on non-representative patches. Formal statements showing attention-mass amplification and sparse sufficiency of image-level readouts are provided in Appendix A.3 and Appendix A.4. To validate whether our theory aligns with practical observations, which are influenced by the projection head, multi-head attention, and registers, we conduct an experiment to measure the correlation between [CLS]-to-patch attention mass and the gradient norm received from the DINO loss. As shown in Fig. 4, these two quantities show a strong Spearman correlation of 0.925.

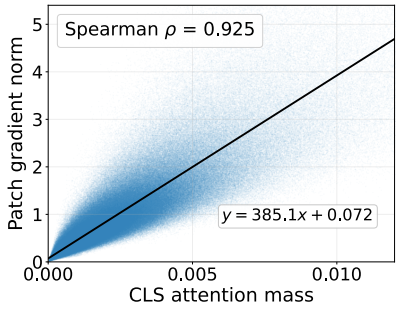


Figure 4: [CLS] attention mass vs. DINO gradient norm.

4 Methodology

In this section, we introduce Cross Distillation (CODI) as a target-side modification motivated by the analysis above. Specifically, within a typical self-distillation framework, we mix a small proportion of teacher class-token semantics into the patch targets. The analysis shows that CODI adds a target-level semantic correction that is not explicitly weighted by [CLS] attention mass. An overview of CODI is presented in Fig. 5.

4.1 Self-Distillation Framework

We build on a typical self-distillation framework [3, 48, 27]. A student network f_s is trained to match the target distributions produced by a momentum teacher network f_t, which is updated as

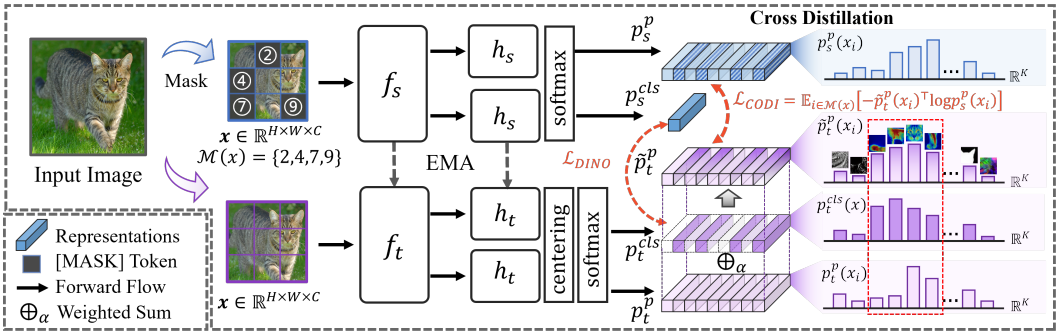


Figure 5: Overview of CODI. In the self-distillation framework, CODI constructs the masked-patch target by mixing the teacher patch-token distribution with a small teacher class-token component.

an exponential moving average of the student. Both networks use a ViT backbone [12]. Given an augmented view x, the teacher produces a class-token embedding $z_t^{\mathrm{cls}}(x)$ and patch-token embeddings $\{z_{t,i}^{\mathrm{p}}(x)\}_{i=1}^{N}$, where N is the number of patches. After a shared projection head, branch-specific centering, and sharpening, we obtain the teacher class-level and patch-level target distributions:

$$p_t^{\mathrm{cls}}(x) = \mathrm{softmax}\left(\frac{h_t(z_t^{\mathrm{cls}}(x)) - c^{\mathrm{cls}}}{\tau_t}\right), \quad p_t^{\mathrm{p}}(x_i) = \mathrm{softmax}\left(\frac{h_t(z_{t,i}^{\mathrm{p}}(x)) - c^{\mathrm{p}}}{\tau_t}\right), \tag{4}$$

where h_t is the shared teacher projection head that maps both class and patch tokens to K-dimensional prototype logits, c^{cls} and c^{p} are separate running centers for the class and patch teacher targets, and τ_t is the teacher temperature. Student predictions are obtained by applying softmax to student logits divided by the student temperature τ_s; they follow the same shared-head design with the student head h_s, but do not use teacher centering. The class-token distribution captures the global semantic assignment of the view, while the patch-token distribution provides local supervision for each patch. Our goal is to couple these two levels of supervision when constructing masked patch targets.

4.2 Cross Distillation via Target Mixing

In masked patch self-distillation, each masked student patch is usually supervised by the corresponding teacher patch target alone. This target is locally informative, but it may emphasize low-level appearance details and does not explicitly encourage all patch representations to preserve discriminative semantics. We therefore introduce *Cross Distillation* (CODI), a target-side mixing strategy that transfers global class-token semantics into masked patch supervision.

For each teacher-student patch-distillation pair, and for each masked patch position $i \in \mathcal{M}(x)$, we construct a mixed teacher target by combining the teacher patch distribution with the teacher class-token distribution from the same teacher view that provides the patch target:

$$\widetilde{p}_t^{\mathrm{p}}(x_i) = (1-\alpha)p_t^{\mathrm{p}}(x_i) + \alpha p_t^{\mathrm{cls}}(x), \tag{5}$$

where $\alpha \in [0, 1)$ controls the strength of global semantic guidance. When $\alpha = 0$, CODI reduces to the standard iBOT loss. Since class and patch tokens are projected by the same head, both targets are distributions over the same K prototype coordinates, although they are centered with different running centers before the softmax. The centers shift the logits before softmax but do not change the prototype coordinate system; therefore both distributions remain defined over the same prototype indices, and their convex combination remains a valid teacher target. The CODI loss is defined as:

$$\mathcal{L}_{\mathrm{CODI}} = -\frac{1}{|\mathcal{M}(x)|}\sum_{i\in\mathcal{M}(x)} \widetilde{p}_t^{\mathrm{p}}(x_i)^\top \log p_s^{\mathrm{p}}(x_i), \tag{6}$$

where $p_s^{\mathrm{p}}(x_i)$ is the student prediction for the masked patch. For image-level supervision, we keep the standard DINO objective, which aligns student and teacher class tokens across augmented views:

$$\mathcal{L}_{\mathrm{DINO}} = -\sum_{(u,v)\in\mathcal{V}} p_t^{\mathrm{cls}}(v)^\top \log p_s^{\mathrm{cls}}(u), \tag{7}$$

where $\mathcal{V}$ denotes the set of teacher-student view pairs. The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{DINO}} + \lambda_1 \cdot \mathcal{L}_{\text{CODI}} + \lambda_2 \cdot \mathcal{L}_{\text{koleo}}. \tag{8}$$

Here, $\mathcal{L}_{\text{koleo}}$ denotes the KoLeo loss adopted from DINOv2 [27]. Intuitively, CODI modifies the label used in masked image modeling. Instead of asking each masked patch to reconstruct a purely patch-level teacher target, we encourage the masked patch prediction to contain global semantic information from the teacher class token. This weak global guidance helps the model avoid overfitting to local texture or appearance details during reconstruction, while preserving more discriminative image-level semantics.

4.3 Theoretical Analysis

The result in Sec. 3.2 shows that the standard `[CLS]` path gates the global semantic signal to patch i by the attention factor π_i. We now show that CODI adds a direct target-level semantic path whose strength is controlled by α and does not depend on π_i. For a masked patch i, write

$$q_i^t := p_t^{\text{p}}(x_i), \qquad q_{\text{cls}}^t := p_t^{\text{cls}}(x), \qquad \xi_i := h_s(z_{s,i}^{\text{p}})/\tau_s, \qquad p_i^s = \text{softmax}(\xi_i), \tag{9}$$

where q_i^t and q_{cls}^t are the teacher targets produced by the same projection head with their respective centers, and ξ_i is the temperature-scaled student patch logit. The original iBOT patch loss and the CODI patch loss are

$$\mathcal{L}_{\text{patch}}^0(i) = -(q_i^t)^\top \log p_i^s, \qquad \mathcal{L}_{\text{patch}}^\alpha(i) = -(\widetilde{q}_i^t)^\top \log p_i^s, \qquad \widetilde{q}_i^t = (1-\alpha)q_i^t + \alpha q_{\text{cls}}^t. \tag{10}$$

Theorem 3 (Target mixing gives attention-independent guidance)**.** *Under negative logit-gradient flow for ξ_i,*

$$\dot{\xi}_i^0 = q_i^t - p_i^s, \qquad \dot{\xi}_i^\alpha = q_i^t - p_i^s + \alpha(q_{\text{cls}}^t - q_i^t). \tag{11}$$

Thus, each masked patch prediction receives an additional target-level logit-gradient term $\alpha(q_{\text{cls}}^t - q_i^t)$, which is not multiplied by its `[CLS]` attention mass π_i.

Remark 4. *The mixed target moves the local patch target toward the global teacher distribution along a line segment in the prototype simplex. Unlike the `[CLS]` attention gradient, this correction is not explicitly gated by π_i.*

5 Experiments

We first describe the pretraining settings in Sec. 5.1. Then, we evaluate CODI across seven protocols in Sec. 5.2 and compare with Gram anchoring in Sec. 5.3. Finally, in Sec. 5.4, we discuss the effect of cross distillation. Additional results on unsupervised segmentation, object detection, and smaller models are provided in Appendix D.

5.1 Pretraining Setting

Following the standard setup [3, 48], we pretrain our model on ImageNet-1K [29] using the AdamW optimizer [23] with a batch size of 1024. We use a ViT with a patch size of 14 as the encoder backbone and employ register tokens [11]. We pretrain CODI with ViT-S/14 for 800 epochs, ViT-B/14 for 400 epochs, and ViT-L/14 for 200 epochs. The initial learning rate is set to $2 \times 10^{-3} \times \sqrt{\texttt{batch_size}/1024}$ and decays with a cosine annealing scheduler. The `[CLS]`-to-patch distribution mixing ratio α is set to 0.1. Detailed parameter settings are provided in Appendix C.1.

5.2 Overall Performance

In this subsection, we evaluate CODI across seven protocols. We compare with representative SSL methods from different paradigms, including 1) **Contrastive Methods:** MoCo v2 [7] and MoCo v3 [8], 2) **Masked Modeling Methods:** MAE [18] and CAE [6], 3) **Clustering-based Methods:** SwAV [2], DINO [3], iBOT [48], Mugs [49], and DINOv2 [27], and 4) **Dense SSL Method:** Leopart [50].

Table 1: Quantitative results of linear semantic segmentation and classification. Gray rows denote oracles trained with substantially larger data/model; † denotes our reproduced result. The best and second-best results are highlighted with **bold** and underlined, respectively.

Methods	Arch.	Dataset	Epochs	Semantic Segmentation						ImageNet-1K Classification			
				COCO		ADE20K		VOC		k-NN		Linear	
				mIoU	Acc.	mIoU	Acc.	mIoU	Acc.	Top-1	Top-5	Top-1	Top-5
Distilled from Larger Models													
DINOv2 [27]	ViT-S/14	LVD-142M	–	57.4	77.7	43.5	78.9	79.4	95.3	79.0	91.5	80.4	95.1
DINOv2 [27]	ViT-B/14	LVD-142M	–	59.1	78.4	46.3	79.8	81.3	95.7	82.0	92.3	83.9	96.6
DINOv2 [27]	ViT-L/14	LVD-142M	–	58.2	77.5	46.5	79.1	80.4	95.4	83.8	94.3	85.9	97.4
Small Models													
MoCo v2 [7]	RN-50	IN1K	800	37.6	64.0	20.0	61.2	53.3	88.4	61.8	82.8	43.1	72.8
SwAV [2]	RN-50	IN1K	800	42.2	68.0	23.3	65.8	55.4	89.0	65.7	84.9	64.2	85.8
DINO [3]	RN-50	IN1K	800	40.7	66.6	22.4	64.5	53.1	88.4	67.5	86.0	65.9	87.2
MoCo v3 [8]	ViT-S/16	IN1K	300	43.4	70.6	24.4	70.5	61.7	91.4	32.2	50.5	54.8	78.8
DINO [3]	ViT-S/16	IN1K	800	39.5	67.1	23.1	67.4	50.0	88.4	74.4	90.0	76.0	92.7
iBOT [48]	ViT-S/16	IN1K	800	47.7	72.7	28.5	72.3	66.7	92.4	75.1	89.1	77.0	93.0
Mugs [49]	ViT-S/16	IN1K	800	48.7	72.6	**30.9**	72.8	71.8	93.3	75.5	89.0	77.7	93.5
Leopart [50]	ViT-S/16	IN1K+COCO	800+50	48.8	73.2	29.7	72.1	68.2	92.6	52.0	74.9	64.6	85.3
DINOv2† [27]	ViT-S/14	IN1K	800	48.1	72.7	29.6	72.6	68.3	92.7	76.9	91.4	77.9	93.7
CODI (ours)	ViT-S/14	IN1K	800	**50.3**	**74.0**	30.8	**73.4**	**73.1**	**93.6**	**77.1**	**91.7**	**78.3**	**94.1**
Base Models													
CAE [6]	ViT-B/16	IN1K	800	45.5	71.0	27.0	71.9	64.9	92.2	37.7	60.9	21.0	42.0
CAE [6]	ViT-B/16	IN1K	1600	46.2	71.6	27.5	72.2	65.6	92.3	33.7	53.0	20.9	41.8
MAE [18]	ViT-B/16	IN1K	800	37.9	65.5	20.6	66.2	52.3	88.8	18.4	32.7	39.2	62.3
MoCo v3 [8]	ViT-B/16	IN1K	300	48.2	73.0	29.6	72.9	69.6	92.9	71.3	88.4	60.8	83.8
DINO [3]	ViT-B/16	IN1K	800	45.7	71.5	28.8	72.0	63.6	91.6	76.1	89.8	77.8	93.7
DINO [3]	ViT-B/8	IN1K	800	47.0	72.3	30.5	72.7	65.3	92.3	77.3	90.5	79.7	94.8
iBOT [48]	ViT-B/16	IN1K	400	52.2	75.3	34.3	75.2	73.3	93.9	77.0	90.3	79.2	94.5
Mugs [49]	ViT-B/16	IN1K	400	51.4	74.1	34.6	74.5	72.7	93.5	78.0	**91.3**	80.1	94.5
DINOv2† [27]	ViT-B/14	IN1K	400	53.0	75.6	35.5	75.8	74.9	93.9	78.9	90.7	81.0	95.3
CODI (ours)	ViT-B/14	IN1K	400	**55.1**	**76.6**	**36.3**	**76.2**	**77.2**	**94.6**	**79.2**	90.8	**81.2**	**95.5**
Large Models													
CAE [6]	ViT-L/16	IN1K	1600	48.8	73.1	31.9	74.6	70.7	93.2	64.0	81.9	48.2	70.9
iBOT [48]	ViT-L/16	IN1K	250	53.5	76.2	36.0	76.0	74.7	94.4	78.0	90.8	80.6	95.3
Mugs [49]	ViT-L/16	IN1K	250	53.2	75.0	36.3	75.3	74.6	93.9	80.2	91.5	81.3	95.4
I-JEPA [1]	ViT-H/14	IN1K	300	45.7	69.3	28.1	70.2	69.6	92.3	71.4	87.9	75.4	91.0
DINOv2† [27]	ViT-L/14	IN1K	200	56.1	77.5	39.1	77.6	78.0	94.9	80.4	**92.3**	81.6	95.4
CODI (ours)	ViT-L/14	IN1K	200	**58.1**	**78.3**	**40.6**	**78.1**	**80.1**	**95.5**	**81.2**	91.7	**82.6**	**96.2**

5.2.1 Evaluation on Dense Prediction Tasks

Semantic Segmentation. To evaluate whether CODI learns well-structured representations, we use the standard off-the-shelf linear probing setting. The pretrained models are frozen, and a linear head is trained on top of the feature extractor to predict the label of each pixel. We sweep the learning rate of the linear head over a predefined set. Detailed settings are provided in Appendix C.2. We evaluate model performance on the widely used COCO [20], Pascal VOC [13], and ADE20K [47] datasets. The results are shown in Tab. 1. Among ImageNet-1K-pretrained baselines with comparable model scale, CODI achieves strong segmentation performance. Compared with reproduced DINOv2 under the aligned setting, CODI improves the average mIoU by $+2.7$, $+1.7$, and $+1.9$ for ViT-S/14, ViT-B/14, and ViT-L/14, respectively. Specifically, CODI achieves 80.1 mIoU with the ViT-Large model on Pascal VOC, approaching the original large-data DINOv2 [27] reference while using only ImageNet-1K pretraining in our setting.

Depth Estimation. Similar to the semantic segmentation setting, we freeze the encoder and train a linear head to predict depth. Following DINOv3 [33], we project depth values into discrete bins and sweep the learning rate for the projection head. We evaluate the models on the NYU-Depth v2 [32], KITTI [14], and SUNRGBD datasets. Specifically, SUNRGBD is evaluated under a zero-shot transfer setting, where models are trained on NYU-Depth v2 and tested only on SUNRGBD. The results are shown in Tab. 2. Although CODI does not obtain the best performance on all tasks, it still achieves the highest overall ranking, showing its effectiveness across settings. For ViT-S/14, CODI improves NYU δ_1 by $+1.4$ over reproduced DINOv2, and CODI matches or improves reproduced DINOv2 in 24 out of 27 metric comparisons across all settings.

Semi-supervised Video Object Segmentation. We also evaluate CODI on the DAVIS 2017 dataset [28]. Following the standard evaluation protocol, the pixel label of the first frame is given, and labels in the following frames are propagated using representation similarity. In this setting, we also sweep several key hyperparameters and report the best results. Detailed settings are introduced in

Table 2: Depth estimation performance on NYU-Depth V2, KITTI, and SUNRGBD zero-shot transfer. Rank is computed within each model size.

Methods	Arch.	Epochs	NYU-Depth V2			KITTI			NYU→SUNRGBD			Rank
			$\delta_1 \uparrow$	Rel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	Rel $\downarrow$	RMSE $\downarrow$	$\delta_1 \uparrow$	Rel $\downarrow$	RMSE $\downarrow$	$\downarrow$
Distilled from Larger Models												
DINOv2 [27]	ViT-S/14	–	84.8	0.134	0.437	91.0	0.096	3.119	77.4	0.269	0.473	–
DINOv2 [27]	ViT-B/14	–	87.8	0.120	0.408	92.0	0.092	2.931	80.7	0.255	0.443	–
DINOv2 [27]	ViT-L/14	–	89.5	0.111	0.390	92.3	0.090	2.891	81.4	0.248	0.439	–
Small Models												
MoCo v2 [7]	RN-50	800	62.9	0.230	0.719	82.4	0.130	4.319	55.5	0.366	0.749	7.0
SwAV [2]	RN-50	800	61.9	0.225	0.721	84.3	0.123	3.985	55.0	0.367	0.753	6.2
DINO [3]	RN-50	800	61.6	0.228	0.726	83.7	0.127	4.068	54.7	0.370	0.755	7.7
MoCo v3 [8]	ViT-S/16	300	69.8	0.192	0.615	77.2	0.170	5.007	50.6	0.411	0.815	7.7
DINO [3]	ViT-S/16	800	67.7	0.202	0.651	83.9	0.132	4.061	59.6	0.343	0.678	5.8
iBOT [48]	ViT-S/16	800	72.1	0.183	0.584	85.3	0.125	3.907	63.5	0.330	0.634	3.3
Mugs [49]	ViT-S/16	800	72.3	0.179	0.586	84.1	0.132	3.963	64.4	**0.306**	0.613	3.4
DINOv2† [27]	ViT-S/14	800	72.7	0.184	0.583	**86.6**	0.112	**3.757**	63.2	0.332	0.631	2.6
CODI (ours)	ViT-S/14	800	**74.1**	**0.176**	**0.570**	86.5	**0.109**	3.834	**64.5**	0.323	**0.610**	**1.3**
Base Models												
MAE [18]	ViT-B/16	800	65.3	0.224	0.673	84.3	0.123	4.087	59.1	0.350	0.688	8.0
MoCo v3 [8]	ViT-B/16	300	72.5	0.181	0.585	85.3	0.122	3.954	65.9	0.308	0.606	5.9
DINO [3]	ViT-B/16	800	71.7	0.182	0.593	86.5	0.117	3.771	62.8	0.336	0.644	6.7
DINO [3]	ViT-B/8	800	72.8	0.178	0.579	86.9	0.115	3.697	65.1	0.314	0.613	5.3
iBOT [48]	ViT-B/16	400	77.7	**0.160**	0.523	**88.3**	0.110	3.514	**69.1**	0.299	**0.565**	2.1
Mugs [49]	ViT-B/16	400	74.9	0.164	0.550	87.7	0.111	3.620	66.6	**0.294**	0.585	3.6
DINOv2† [27]	ViT-B/14	400	78.1	**0.160**	0.519	88.0	0.112	3.502	68.0	0.310	0.570	2.8
CODI (ours)	ViT-B/14	400	**78.3**	**0.160**	**0.516**	**88.3**	**0.109**	**3.487**	67.8	0.306	0.568	**1.7**
Large Models												
iBOT [48]	ViT-L/16	250	80.4	0.147	0.492	90.5	**0.096**	3.253	70.1	0.303	0.551	2.7
Mugs [49]	ViT-L/16	250	76.7	0.158	0.524	88.4	0.106	3.485	68.0	0.299	0.574	4.1
I-JEPA [1]	ViT-H/14	300	76.5	0.170	0.534	86.3	0.119	3.725	68.4	0.303	0.568	4.7
DINOv2† [27]	ViT-L/14	200	81.4	0.147	0.481	90.4	0.097	3.266	71.5	0.299	0.529	2.4
CODI (ours)	ViT-L/14	200	**82.0**	**0.145**	**0.472**	**90.6**	**0.096**	**3.244**	**71.7**	**0.297**	**0.528**	**1.1**

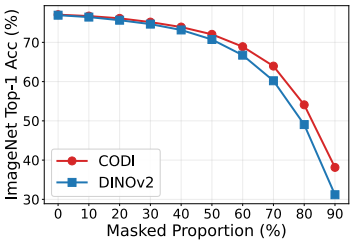
Table 3: DAVIS semi-supervised VOS results.

Methods	Arch.	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_{\text{mean}}$	$\mathcal{F}_{\text{mean}}$
DINOv3	ViT-7B/16	71.1	–	–
DINO	ViT-B/16	63.2	61.5	64.9
iBOT	ViT-B/16	64.3	63.2	65.5
CAE	ViT-L/16	65.1	62.8	67.5
iBOT	ViT-L/16	63.8	62.1	65.5
Mugs	ViT-L/16	62.1	60.3	63.0
DINOv2	ViT-L/14	64.2	62.2	66.3
CODI (ours)	ViT-B/14	65.7	**63.6**	67.9
CODI (ours)	ViT-L/14	**66.0**	63.3	**68.7**

Table 4: ImageNet-1K full fine-tuning Top-1 accuracy.

Method	Arch.	Epochs	Acc.
MoCo v3	ViT-S/16	300	81.4
DINO	ViT-S/16	800	82.0
iBOT	ViT-S/16	800	82.3
CODI (ours)	ViT-S/14	800	**82.7**
MoCo v3	ViT-B/16	300	83.2
DINO	ViT-B/16	400	83.6
iBOT	ViT-B/16	400	84.0
CODI (ours)	ViT-B/14	400	**84.3**

Figure 6: ImageNet-1K k-NN classification under occlusion.



Appendix C.2. As shown in Tab. 3, CODI achieves 66.0 on $\mathcal{J}\&\mathcal{F}$ with the ViT-L/14 model, surpassing all comparable (non-DINOv3) baselines. Notably, CODI outperforms the original DINOv2, which is distilled from a larger model, by a clear margin of $+1.8$ on $\mathcal{J}\&\mathcal{F}$. These results suggest that CODI features are more effective for affinity-based mask propagation.

5.2.2 Evaluation on Image-level Tasks

Image Classification with Frozen Backbone. To provide a comprehensive analysis of the effect of CODI, we evaluate the models on image classification tasks. Following the standard protocol [3, 27], we use two classifiers, a parameter-free k-NN classifier and a linear classifier, and report Top-1 and Top-5 accuracy. We also sweep the hyperparameters of these heads and report the best results. Details are provided in Appendix C.2. As shown in Tab. 1, we observe a consistent improvement in Top-1 classification performance. Notably, the Top-1 performance gain scales with model size: with ViT-Large, CODI achieves improvements of $+0.8$ and $+1.0$ in the k-NN and linear classification settings, respectively, compared with DINOv2.

Full Fine-Tuning on ImageNet-1K. Following the evaluation protocol of iBOT [48], we further fine-tune the model on ImageNet-1K for 100 epochs. As shown in Tab. 4, although the performance

Table 5: Comparison of ViT-S/14 pretraining and fine-tuning settings on ImageNet-1K. Gram anchoring is applied to a DINOv2 checkpoint.

Method	Pretraining iterations	Fine-tuning iterations	ImageNet k-NN (%)	VOC mIoU (%)
DINOv2	1M	–	76.9	68.3
DINOv2 (best checkpoint)	625K	–	75.9	69.4
DINOv2 + Gram anchoring	1M	100K	76.7	70.2
DINOv2 + CODI fine-tuning	1M	100K	76.8	71.4
CODI	1M	–	**77.1**	**73.1**

Table 6: Ablation of the CODI mixing ratio α and target distribution.

Target	α	COCO	VOC	ADE	k-NN
None	0.0	48.1	68.3	29.6	76.9
Averaged patch	0.1	48.9	69.8	29.9	76.6
`[CLS]`	0.1	**50.3**	**73.1**	**30.8**	77.1
`[CLS]`	0.2	49.2	71.5	29.7	**77.2**
`[CLS]`	0.5	42.7	60.5	25.6	76.1
Uniform	0.1	45.8	62.5	27.3	75.8

Table 7: Semantic alignment and regularization after 100 epochs on ImageNet-1K.

Method	k-NN	VOC
DINOv2	71.1	61.4
Random-image `[CLS]`	70.6	62.1
Entropy-matched patch	70.8	62.9
CODI (same-image `[CLS]`)	**71.3**	**65.2**

gain is reduced after fine-tuning, CODI still improves Top-1 accuracy by $+0.4$, showing that it provides a better initialization.

Classification with Patch Occlusion. To study how patch-level target mixing improves image-level performance, we conduct an additional experiment on classification under occlusion. For each input image, we randomly mask a proportion of patches and use the k-NN classifier. The results are shown in Fig. 6. Compared with DINOv2, CODI achieves better performance at every masking ratio from 0% to 90%, and the performance gap increases as the masked ratio grows. This suggests that CODI reduces reliance on a small set of highly discriminative patches. As a result, CODI is more robust when representative patches are missing.

5.3 Comparison with Gram Anchoring

We compare CODI with DINOv3's Gram anchoring [33] under an aligned ViT-S/14 setting. The Gram-anchoring results are obtained by fine-tuning a DINOv2 checkpoint for 100K iterations. For a fair comparison, we tune the resolution in $\{224 \times 224, 448 \times 448\}$, whether to distill from the previous best or final checkpoint, and whether to update the Gram-anchor teacher, and report the best result. We also fine-tune DINOv2 with the CODI loss for the same number of iterations. As shown in Tab. 5, although Gram anchoring recovers degraded dense performance through distillation, CODI fine-tuning outperforms it by $+1.2$ mIoU under the aligned setting. Training CODI from scratch further improves performance, surpassing the Gram-anchoring baseline by $+2.9$ mIoU with fewer training iterations. Fine-tuning does not improve classification performance, and training CODI from scratch still yields the best results. Comparisons with the distilled DINOv3 and SigLIP2 checkpoints are provided in Appendix D.4.

5.4 The Effect of Cross Distillation

How does cross distillation affect downstream performance? We study the target distribution and mixing strength α, with additional dataset sweeps in Appendix D.6. As shown in Tab. 6, introducing global semantics into the patch target improves dense performance, with the `[CLS]` token outperforming the averaged patch target. Mixing a uniform distribution instead reduces both classification and segmentation performance. The results also show that $\alpha = 0.1$ is an appropriate mixing strength: when $\alpha = 0$, CODI reduces to DINOv2; when α is large, performance degrades.

On COCO, $\alpha = 0.05$ and 0.075 achieve the best k-NN accuracy, whereas $\alpha = 0.1$ gives the best VOC mIoU. Increasing α to 0.2 reduces both metrics below DINOv2. On ImageNet-22K, excluding

ImageNet-1K and using a 100K-iteration schedule, $\alpha = 0.1$ achieves the best results on both metrics. These findings support $\alpha = 0.1$ as a strong starting point, though not a universal optimum for every downstream metric or evidence of a uniformly wide operating range. The final choice depends on whether image-level or dense performance is prioritized.

Global semantic guidance or regularization? To disentangle global semantic guidance from regularization, we conduct two controlled ablations after 100 epochs of ImageNet-1K pretraining. For *random-image `[CLS]`*, we keep $\alpha = 0.1$, but replace the teacher `[CLS]` distribution with that of another randomly selected image from the same mini-batch. This preserves a valid teacher `[CLS]` distribution while breaking image-level semantic alignment. For the *entropy-matched patch target*, we first compute the entropy of the standard CODI target after mixing its patch distribution with the same-image `[CLS]` distribution at $\alpha = 0.1$. We then omit `[CLS]` mixing and adjust only the patch-target temperature so that the resulting patch distribution matches this entropy. This preserves CODI's target-entropy change while keeping the target entirely local and excluding global `[CLS]` information.

As shown in Tab. 7, the entropy-matched patch control improves VOC mIoU by $+1.5$ over DINOv2 while reducing k-NN by 0.3, showing that matching CODI's target entropy alone provides a regularization benefit. Random-image `[CLS]` yields a smaller improvement in dense prediction performance and also reduces k-NN. In contrast, same-image `[CLS]` (CODI) improves both metrics and yields the largest mIoU gain. Thus, regularization contributes, but semantically aligned `[CLS]`-to-patch guidance is the dominant source of improvement.

Does the performance gain arise from non-representative patches? To examine whether the performance improvement comes from improving non-representative patches, we visualize the performance gain on patches with different attention rankings. We rank patches by the final-layer `[CLS]`-to-patch attention mass averaged over heads, excluding register tokens. Low-ranked patches have relatively low `[CLS]`-to-patch attention mass and thus correspond to non-representative patches. As shown in Fig. 7, the performance gain of CODI mainly comes from these non-representative patches, which is consistent with our design goal.

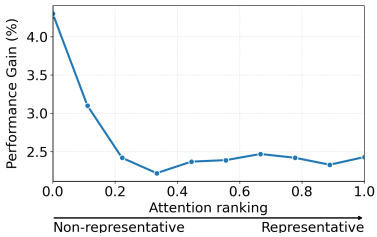


Figure 7: Performance gain by `[CLS]` attention rank.

How does CODI influence the representation structure? Since CODI mixes a small amount of the `[CLS]` distribution into the patch targets, a natural question arises: although this operation acts like a centering operation in the representation space, why does it improve representation separability instead of confusing representations across classes? To answer this question, we conduct an experiment. Using the pixel labels from Pascal VOC, we first aggregate patch-level labels by majority voting. Then, we compute the inter-class and intra-class distances between patch representations based on the labels. As shown in Tab. 8, although both the intra-class and inter-class distances shrink as expected, the gap between them, which measures class separability, increases. These results also show a trade-off between inter-class and intra-class distances, suggesting that we need to choose α carefully to prevent collapse.

Table 8: Representation distance analysis.

Methods	Intra-$d \downarrow$	Inter-$d \uparrow$	Gap $\uparrow$
DINOv2	0.6821	**0.9168**	0.2347
CODI	**0.5864**	0.8430	**0.2566**

6 Conclusion

This paper studies the phenomenon that the `[CLS]` token in ViTs tends to gather information from a set of *representative patches* rather than the whole object. We provide a local gradient analysis showing that, through the final `[CLS]` attention readout, image-level semantic gradients are weighted by `[CLS]`-to-patch attention mass. As a result, non-representative patches, i.e., patches with low attention scores, receive weaker image-level semantic gradients through this path, which can contribute to less discriminative patch features. To address this issue, we propose a simple yet effective cross-distillation (CODI) design. By mixing a small amount of teacher class-token information into masked patch targets, CODI ensures that every masked patch supervised by the patch loss receives a target-level global semantic correction. We empirically evaluate CODI on multiple benchmarks and evaluation settings. The results demonstrate its effectiveness and scalability.

Acknowledgments and Disclosure of Funding

This work was supported in part by National Natural Science Foundation of China: 62525212, U23B2051, 62236008, 62441232, 62521007, 62411540034, 62132006, and U21B2038, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB0680201, in part by the China National Postdoctoral Program for Innovative Talents under Grant BX20250377, in part by the China Postdoctoral Science Foundation under Grant 2026M791710, and in part by the project ZR2025ZD01 supported by Shandong Provincial Natural Science Foundation.

The authors declare no other competing interests.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[6] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, 2024.

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

[9] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. PiCIE: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021.

[10] Siran Dai, Qianqian Xu, Peisong Wen, Yang Liu, and Qingming Huang. Exploring structural degradation in dense representations for self-supervised learning. *arXiv preprint arXiv:2510.17299*, 2025.

[11] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88: 303–338, 2010.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[16] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[21] Yang Liu, Qianqian Xu, Peisong Wen, Siran Dai, and Qingming Huang. When the future becomes the past: Taming temporal correspondence for self-supervised video representation learning, 2025. URL https://arxiv.org/abs/2503.15096.

[22] Yang Liu, Qianqian Xu, Peisong Wen, Siran Dai, Xilin Zhao, and Qingming Huang. From static to dynamic: Exploring self-supervised image-to-video representation transfer learning, 2026. URL https://arxiv.org/abs/2603.26597.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[24] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-jepa 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

[25] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[26] Dantong Niu, Xudong Wang, Xinyang Han, Long Lian, Roei Herzig, and Trevor Darrell. Unsupervised universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22744–22754, 2024.

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[28] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[30] Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16536–16547, 2023.

[31] Hyun Seok Seong, WonJun Moon, SuBeen Lee, and Jae-Pil Heo. Leveraging hidden positives for unsupervised semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19540–19549, 2023.

[32] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[33] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[34] Qing Su, Anton Netchaev, Hai Li, and Shihao Ji. Flsl: Feature-level self-supervised learning. *Advances in Neural Information Processing Systems*, 36:6568–6581, 2023.

[35] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[36] Feng Wang, Huiyu Wang, Chen Wei, Alan Yuille, and Wei Shen. Cp 2: Copy-paste contrastive pretraining for semantic segmentation. In *European Conference on Computer Vision*, pages 499–515. Springer, 2022.

[37] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *International Conference on Computer Vision*, pages 568–578, 2021.

[38] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3024–3033, 2021.

[39] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3124–3134, 2023.

[40] Peisong Wen, Qianqian Xu, Siran Dai, Runmin Cong, and Qingming Huang. Semantic concentration for self-supervised dense representations learning, 2025. URL https://arxiv.org/abs/2509.09429.

[41] Peisong Wen, Qianqian Xu, Minye Lei, Siran Dai, Yang Liu, and Qingming Huang. Slot-jepa: Object-structured prediction interfaces for latent world models, 2026. URL https://arxiv.org/abs/2610.28074.

[42] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[43] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[44] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing*, 29:9316–9329, 2020.

[45] Sukmin Yun, Hankook Lee, Jaehyung Kim, and Jinwoo Shin. Patch-level representation learning for self-supervised vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8354–8363, 2022.

[46] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017.

[48] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[49] Pan Zhou, Yichen Zhou, Chenyang Si, Weihao Yu, Teck Khim Ng, and Shuicheng Yan. Mugs: A multi-granular self-supervised learning framework. *arXiv preprint arXiv:2203.14415*, 2022.

[50] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14502–14511, 2022.

Appendix Contents

A Detailed Theory and Proofs **16**

A.1 Why a Local [CLS] Readout Is Sufficient for the Argument 16

A.2 Exact Gradients of the [CLS] Attention Readout 16

A.3 Attention Amplification of Representative Patches 17

A.4 Sparse Sufficiency of an Image-Level Readout 18

A.5 CODI Target Mixing 19

A.6 Register Tokens as Softmax Competitors 20

B Pseudo Code **21**

C Detailed Settings **21**

C.1 Pretraining Settings 21

C.2 Evaluation Settings 22

D Additional Experiments **23**

D.1 Unsupervised Semantic Segmentation 23

D.2 Object Detection and Instance Segmentation 24

D.3 Results on Smaller Models 24

D.4 Comparison with DINOv3 and SigLIP2 24

D.5 Additional Experiments on Model Robustness 25

D.6 Sensitivity to the Mixing Ratio Across Pretraining Datasets 25

E Qualitative Results **26**

F Limitations **26**

A Detailed Theory and Proofs

This appendix provides the formal statements and proofs underlying Sec. 3.2 and Sec. 4.3. We deliberately keep the model local. The goal is not to describe every computational path in a multi-layer ViT, but to analyze one explicit path: the final [CLS] attention readout through which the image-level DINO loss aggregates patch information and sends gradients back to patch tokens.

A.1 Why a Local [CLS] Readout Is Sufficient for the Argument

A full ViT contains multiple self-attention blocks, residual connections, MLPs, normalization layers, and projection heads. Our main claim, however, concerns the following specific question: along the final [CLS] attention row, how is the gradient from the image-level teacher target distributed across patch tokens? This question can be answered exactly with a one-step local readout.

Let P denote image patch tokens in the final block, and let T denote all tokens competing in the final [CLS] attention row, including register tokens when used. Let a_{cls} be the [CLS] query, and let k_i and v_i be the key and value of token $i \in T$. The attention score, attention mass, and readout are

$$s_i = \frac{a_{\text{cls}}^\top k_i}{\sqrt{d}}, \qquad \pi_i = \frac{\exp(s_i)}{\sum_{j \in T} \exp(s_j)}, \qquad z = \sum_{j \in T} \pi_j v_j. \tag{12}$$

The output projection can be absorbed into v_i, since it is a shared linear map applied after the weighted sum. During the local calculation, upstream representations, teacher targets, and all other paths are held fixed. Under this convention, for $i \in P$, the derivatives with respect to v_i and s_i measure how the image-level loss allocates its gradient through this attention readout.

This abstraction does not imply that patch tokens receive no gradients from other network components. It only isolates a mechanism that is directly relevant to the representative-patch observation: when the [CLS] loss depends on a softmax-weighted readout, the gradient flowing through that readout is multiplied by the attention mass.

A.2 Exact Gradients of the [CLS] Attention Readout

Treat the teacher class-token target q_{cls}^t as fixed, and define

$$\mathcal{L}_{\text{cls}} = \text{CE}\left(q_{\text{cls}}^t, \text{softmax}(\phi_s(z)/\tau_s)\right), \qquad g := \nabla_z \mathcal{L}_{\text{cls}}. \tag{13}$$

For the following proof, the exact form of the head ϕ_s is not important. We only need the gradient of the scalar loss with respect to the local readout.

Proof of Theorem 1. We prove the value-path identity first. In Eq. (12), the readout is

$$z = \sum_{j \in T} \pi_j v_j. \tag{14}$$

When differentiating with respect to a value vector v_i, the attention weights π_j are fixed because they depend on the scores s_j, not on the values v_j. Hence

$$\frac{\partial z}{\partial v_i} = \pi_i I, \tag{15}$$

where I is the identity matrix. By the chain rule,

$$\nabla_{v_i} \mathcal{L}_{\text{cls}} = \left(\frac{\partial z}{\partial v_i}\right)^\top \nabla_z \mathcal{L}_{\text{cls}} = \pi_i g. \tag{16}$$

This proves the first identity.

We next differentiate with respect to the score s_i. The derivative of the softmax is

$$\frac{\partial \pi_j}{\partial s_i} = \pi_j(\mathbf{1}_{i=j} - \pi_i). \tag{17}$$

Therefore,

$$\frac{\partial z}{\partial s_i} = \sum_{j \in T} \frac{\partial \pi_j}{\partial s_i} v_j \tag{18}$$

$$= \pi_i(1 - \pi_i)v_i + \sum_{j \neq i}(-\pi_j \pi_i)v_j \tag{19}$$

$$= \pi_i v_i - \pi_i \sum_{j \in T} \pi_j v_j \tag{20}$$

$$= \pi_i(v_i - z). \tag{21}$$

Applying the chain rule again gives

$$\frac{\partial \mathcal{L}_{\text{cls}}}{\partial s_i} = \left\langle \nabla_z \mathcal{L}_{\text{cls}}, \frac{\partial z}{\partial s_i} \right\rangle \tag{22}$$

$$= \pi_i \langle g, v_i - z \rangle. \tag{23}$$

Defining $u_i = -\langle g, v_i - z \rangle$, we obtain

$$-\frac{\partial \mathcal{L}_{\text{cls}}}{\partial s_i} = \pi_i u_i. \tag{24}$$

The same calculation also yields the corresponding ratio identities. For any two patches $i, r \in P$ with $g \neq 0$, since

$$\|\nabla_{v_i} \mathcal{L}_{\text{cls}}\| = \pi_i \|g\|, \tag{25}$$

we have

$$\frac{\|\nabla_{v_i} \mathcal{L}_{\text{cls}}\|}{\|\nabla_{v_r} \mathcal{L}_{\text{cls}}\|} = \frac{\pi_i}{\pi_r}. \tag{26}$$

Similarly, when $u_i, u_r > 0$, the negative score gradients are positive and

$$\frac{-\partial \mathcal{L}_{\text{cls}}/\partial s_i}{-\partial \mathcal{L}_{\text{cls}}/\partial s_r} = \frac{\pi_i u_i}{\pi_r u_r}. \tag{27}$$

□

A.3 Attention Amplification of Representative Patches

The previous proof gives an instantaneous gradient statement. The next result explains how a set of useful representative patches can gain more attention under local score-gradient flow.

Theorem 5 (Local attention amplification)**.** *Consider the score dynamics*

$$\dot{s}_i = -\frac{\partial \mathcal{L}_{\text{cls}}}{\partial s_i} = \pi_i u_i. \tag{28}$$

Let $\mathcal{R} \subseteq \mathcal{P}$ be a set of representative patches, and define its total attention mass

$$\Pi_{\mathcal{R}} = \sum_{i \in \mathcal{R}} \pi_i. \tag{29}$$

If

$$u_i \geq \gamma > 0 \quad \textit{for } i \in \mathcal{R}, \qquad u_j \leq 0 \quad \textit{for } j \notin \mathcal{R}, \tag{30}$$

then

$$\dot{\Pi}_{\mathcal{R}} \geq (1 - \Pi_{\mathcal{R}})\gamma \sum_{i \in \mathcal{R}} \pi_i^2 \geq 0. \tag{31}$$

Proof. We first compute how softmax probabilities change when scores change over time. Since

$$\pi_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}, \tag{32}$$

the time derivative is

$$\dot{\pi}_i = \sum_j \frac{\partial \pi_i}{\partial s_j} \dot{s}_j \tag{33}$$

$$= \sum_j \pi_i (\mathbf{1}_{i=j} - \pi_j) \dot{s}_j \tag{34}$$

$$= \pi_i \dot{s}_i - \pi_i \sum_j \pi_j \dot{s}_j \tag{35}$$

$$= \pi_i \left(\dot{s}_i - \bar{s} \right), \tag{36}$$

where

$$\bar{s} := \sum_j \pi_j \dot{s}_j \tag{37}$$

is the attention-weighted average score velocity.

Now sum over the representative set $\mathcal{R}$:

$$\dot{\Pi}_{\mathcal{R}} = \sum_{i \in \mathcal{R}} \dot{\pi}_i \tag{38}$$

$$= \sum_{i \in \mathcal{R}} \pi_i \dot{s}_i - \left(\sum_{i \in \mathcal{R}} \pi_i \right) \left(\sum_j \pi_j \dot{s}_j \right) \tag{39}$$

$$= \sum_{i \in \mathcal{R}} \pi_i \dot{s}_i - \Pi_{\mathcal{R}} \sum_j \pi_j \dot{s}_j. \tag{40}$$

Using $\dot{s}_i = \pi_i u_i$, we obtain

$$\dot{\Pi}_{\mathcal{R}} = \sum_{i \in \mathcal{R}} \pi_i^2 u_i - \Pi_{\mathcal{R}} \left(\sum_{i \in \mathcal{R}} \pi_i^2 u_i + \sum_{j \notin \mathcal{R}} \pi_j^2 u_j \right) \tag{41}$$

$$= (1 - \Pi_{\mathcal{R}}) \sum_{i \in \mathcal{R}} \pi_i^2 u_i - \Pi_{\mathcal{R}} \sum_{j \notin \mathcal{R}} \pi_j^2 u_j. \tag{42}$$

By assumption, $u_i \geq \gamma$ for $i \in \mathcal{R}$, so

$$\sum_{i \in \mathcal{R}} \pi_i^2 u_i \geq \gamma \sum_{i \in \mathcal{R}} \pi_i^2. \tag{43}$$

Also $u_j \leq 0$ outside $\mathcal{R}$, so

$$-\Pi_{\mathcal{R}} \sum_{j \notin \mathcal{R}} \pi_j^2 u_j \geq 0. \tag{44}$$

Combining these two inequalities gives

$$\dot{\Pi}_{\mathcal{R}} \geq (1 - \Pi_{\mathcal{R}}) \gamma \sum_{i \in \mathcal{R}} \pi_i^2 \geq 0. \tag{45}$$

Thus, under this local sign condition, the total attention mass on representative patches increases. □

A.4 Sparse Sufficiency of an Image-Level Readout

The attention-amplification result shows how representative patches can be reinforced. The next lemma explains why such a sparse explanation can still be sufficient for an image-level loss. Since the loss observes the pooled readout z, it does not directly constrain every patch value to be used.

Lemma 6 (Sparse sufficiency)**.** *Assume $\ell(z)$ is L-Lipschitz on the convex hull $\mathrm{conv}\{v_i : i \in \mathcal{P}\}$, and assume $\|v_i\| \leq B$ for all i. Suppose there exists a subset $\mathcal{R} \subseteq \mathcal{P}$ and a distribution $\bar{\pi}$ supported on $\mathcal{R}$ such that*

$$\ell\left(\sum_{i \in \mathcal{R}} \bar{\pi}_i v_i \right) \leq \min_{\pi \in \Delta_{\mathcal{P}}} \ell\left(\sum_{i \in \mathcal{P}} \pi_i v_i \right) + \varepsilon. \tag{46}$$

Then, for any $\eta > 0$, there exists a softmax attention distribution π^η satisfying

$$\sum_{i \notin \mathcal{R}} \pi_i^\eta \leq \eta \tag{47}$$

and

$$\ell\left(\sum_{i \in \mathcal{P}} \pi_i^\eta v_i\right) \leq \min_{\pi \in \Delta_\mathcal{P}} \ell\left(\sum_{i \in \mathcal{P}} \pi_i v_i\right) + \varepsilon + 2LB\eta. \tag{48}$$

Proof. Let

$$z_\mathcal{R} = \sum_{i \in \mathcal{R}} \bar{\pi}_i v_i. \tag{49}$$

By assumption, $z_\mathcal{R}$ is already within ε of the best image-level readout obtainable from all patches.

Choose a distribution π^η whose mass outside $\mathcal{R}$ is at most η, and whose restriction to $\mathcal{R}$ is proportional to $\bar{\pi}$. Such a distribution can be approximated by softmax scores because assigning very negative scores to patches outside $\mathcal{R}$ makes their total softmax mass arbitrarily small.

Define

$$z^\eta = \sum_{i \in \mathcal{P}} \pi_i^\eta v_i. \tag{50}$$

The only difference between z^η and $z_\mathcal{R}$ is that at most η probability mass has been moved from vectors in $\mathcal{R}$ to vectors outside $\mathcal{R}$, and then the remaining mass on $\mathcal{R}$ has been renormalized. Since every vector has norm at most B, the perturbation is bounded by

$$\|z^\eta - z_\mathcal{R}\| \leq 2B\eta. \tag{51}$$

Because ℓ is L-Lipschitz,

$$\ell(z^\eta) \leq \ell(z_\mathcal{R}) + L\|z^\eta - z_\mathcal{R}\| \leq \ell(z_\mathcal{R}) + 2LB\eta. \tag{52}$$

Using the assumed ε-optimality of $z_\mathcal{R}$, we get

$$\ell(z^\eta) \leq \min_{\pi \in \Delta_\mathcal{P}} \ell\left(\sum_i \pi_i v_i\right) + \varepsilon + 2LB\eta. \tag{53}$$

This proves the lemma. □

A.5 CODI Target Mixing

We now prove the logit-level statement used in Sec. 4.3. The result is independent of the attention readout because it concerns the masked patch prediction head directly.

Proof of Theorem 3. Use the notation of Theorem 3, and write $\mathcal{L}_i^0 = \mathcal{L}_{\text{patch}}^0(i)$, $\mathcal{L}_i^\alpha = \mathcal{L}_{\text{patch}}^\alpha(i)$, and $\widetilde{q}_i^t = (1-\alpha)q_i^t + \alpha q_{\text{cls}}^t$. For any target distribution y, define the softmax cross-entropy

$$\ell(\xi; y) = -y^\top \log \text{softmax}(\xi). \tag{54}$$

The standard derivative is

$$\nabla_\xi \ell(\xi; y) = \text{softmax}(\xi) - y. \tag{55}$$

Applying this identity to the original patch target q_i^t, we obtain

$$-\nabla_{\xi_i} \mathcal{L}_i^0 = q_i^t - p_i^s. \tag{56}$$

Applying it to the mixed target $\widetilde{q}_i^t$, we get

$$-\nabla_{\xi_i} \mathcal{L}_i^\alpha = \widetilde{q}_i^t - p_i^s \tag{57}$$

$$= (1-\alpha)q_i^t + \alpha q_{\text{cls}}^t - p_i^s \tag{58}$$

$$= q_i^t - p_i^s + \alpha(q_{\text{cls}}^t - q_i^t). \tag{59}$$

Thus, compared with the original patch target, CODI adds the term

$$\alpha(q_{\text{cls}}^t - q_i^t) \tag{60}$$

directly at the patch logits. This term is defined by the two teacher target distributions and does not involve the `[CLS]` attention mass π_i. □

A.6 Register Tokens as Softmax Competitors

The main text includes register tokens as softmax competitors through T. Equivalently, if $\mathcal{G}$ is the set of register indices, the local readout becomes

$$z = \sum_{i \in \mathcal{P}} \pi_i v_i + \sum_{r \in \mathcal{G}} \pi_r v_r, \qquad \sum_{i \in \mathcal{P}} \pi_i + \sum_{r \in \mathcal{G}} \pi_r = 1. \tag{61}$$

The same gradient identities apply to every token in the enlarged softmax. Let

$$\Pi_{\mathcal{G}} = \sum_{r \in \mathcal{G}} \pi_r \tag{62}$$

be the total attention mass assigned to registers. Then the value-path gradients satisfy

$$\sum_{i \in \mathcal{P}} \|\nabla_{v_i} \mathcal{L}_{\text{cls}}\| = (1 - \Pi_{\mathcal{G}}) \|g\|, \qquad \sum_{r \in \mathcal{G}} \|\nabla_{v_r} \mathcal{L}_{\text{cls}}\| = \Pi_{\mathcal{G}} \|g\|. \tag{63}$$

Registers can therefore be useful for storing global or internal computation, but in this local readout they also participate in the same softmax competition as patch tokens. In contrast, the CODI correction in Appendix A.5 is applied to masked patch logits, so it is not reduced when the `[CLS]` readout assigns attention to registers.

B Pseudo Code

Algorithm 1 Self-Distillation with CODI Target Mixing

Require: Dataset $\mathcal{D}$; student $F_\theta = (f_\theta, h_\theta)$; EMA teacher $F_\xi = (f_\xi, h_\xi)$
Require: Student views $\mathcal{U}$; teacher views $\mathcal{G}$; masked patch sampler $\mathcal{M}$; mixing ratio α
for each training iteration t **do**
  Sample images $x \sim \mathcal{D}$ and generate augmented views
  Sample masked patch positions $\mathcal{M}(u)$ for masked student views $u \in \mathcal{U}$
  Get teacher temperature τ_t and EMA momentum m_t
  Teacher targets, no gradient
  for each teacher view $v \in \mathcal{G}$ **do**
    $(z_v^{T,\mathrm{cls}}, \{z_{v,i}^{T,\mathrm{p}}\}_{i=1}^N) \leftarrow f_\xi(v)$
    $q_v^{\mathrm{cls}} \leftarrow \mathrm{softmax}((h_\xi(z_v^{T,\mathrm{cls}}) - c_{\mathrm{cls}})/\tau_t)$
    for each patch position i **do**
      $q_{v,i}^{\mathrm{p}} \leftarrow \mathrm{softmax}((h_\xi(z_{v,i}^{T,\mathrm{p}}) - c_{\mathrm{patch}})/\tau_t)$
    end for
  end for
  for each paired masked student view u and teacher view v **do**
    for each masked patch $i \in \mathcal{M}(u)$ **do**
      $\widetilde{q}_{v,i}^{\mathrm{p}} \leftarrow (1-\alpha) q_{v,i}^{\mathrm{p}} + \alpha q_v^{\mathrm{cls}}$
    end for
  end for
  Update centers c_{cls} and c_{patch} from unmixed teacher logits
  Student forward and losses
  for each student view $u \in \mathcal{U}$ **do**
    $(z_u^{S,\mathrm{cls}}, \{z_{u,i}^{S,\mathrm{p}}\}_{i=1}^N) \leftarrow f_\theta(u, \mathcal{M}(u))$
    $s_u^{\mathrm{cls}} \leftarrow h_\theta(z_u^{S,\mathrm{cls}})$
    $s_{u,i}^{\mathrm{p}} \leftarrow h_\theta(z_{u,i}^{S,\mathrm{p}})$ for $i \in \mathcal{M}(u)$
  end for
  $\mathcal{L}_{\mathrm{DINO}} \leftarrow \sum_{(u,v)\in\mathcal{V}_{\mathrm{cls}}} \mathrm{CE}(q_v^{\mathrm{cls}}, s_u^{\mathrm{cls}})$
  $\mathcal{L}_{\mathrm{CODI}} \leftarrow \sum_{(u,v)\in\mathcal{V}_{\mathrm{patch}}} \frac{1}{|\mathcal{M}(u)|} \sum_{i\in\mathcal{M}(u)} \mathrm{CE}(\widetilde{q}_{v,i}^{\mathrm{p}}, s_{u,i}^{\mathrm{p}})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{DINO}} + \lambda \mathcal{L}_{\mathrm{CODI}}$
  Update student parameters θ using $\nabla_\theta \mathcal{L}$
  $\xi \leftarrow m_t \xi + (1-m_t)\theta$
end for
$\mathcal{V}_{\mathrm{cls}}$ *and* $\mathcal{V}_{\mathrm{patch}}$ *denote the usual cross-view class-token and masked-patch distillation pairs.*
$\mathrm{CE}(q, s) = -\sum_k q_k \log \mathrm{softmax}(s/\tau_s)_k$.

C Detailed Settings

C.1 Pretraining Settings

Pretraining protocol. Unless otherwise specified, we pretrain on ImageNet-1K with a DINOv2/iBOT-style teacher-student framework. The default run uses ViT-S/14 backbones with 4 register tokens and a shared DINO/iBOT projection head. For the backbone, LayerScale is set to 1×10^{-5}, and uniform stochastic depth is set to 0.1. For the projection head, we use 65,536 prototypes, hidden dimension 2048, bottleneck dimension 256, and 3 MLP layers. Each image is augmented into two 224×224 global crops sampled from scale $[0.32, 1.0]$ and eight 98×98 local crops sampled from scale $[0.05, 0.32]$, together with the standard DINOv2/iBOT augmentations: bicubic random resized crop, horizontal flip, color jitter, grayscale conversion, Gaussian blur, solarization, and ImageNet normalization. We train for 800 epochs on 8 GPUs with batch size 128 per GPU, giving a global batch size of 1024 and 1250 iterations per epoch. For compute resources, the ViT-S/14 experiments are run on 8 NVIDIA RTX 4090 GPUs, and the 800-epoch pretraining run takes about 7 days. The ViT-B/14 and ViT-L/14 experiments are run on 8 NVIDIA A100 GPUs.

For optimization, we use AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and cosine learning-rate decay from the sqrt-scaled base learning rate 2×10^{-3} to 1×10^{-6} after 10 warmup epochs. Weight decay follows a cosine schedule from 0.04 to 0.4. We clip gradients at 3.0, freeze the last layer for one epoch, set the patch-embedding learning-rate multiplier to 0.2, and use layer-wise learning-rate decay

0.9. The teacher is updated with EMA, whose momentum is cosine-scheduled from 0.992 to 1.0. The teacher temperature is warmed from 0.04 to 0.07 over 30 epochs, while the student temperature is 0.1 and the centering momentum is 0.9.

For the training losses, DINO and iBOT are both weighted by 1.0, KoLeo regularization uses a weight of 0.1, and prototype coding regularization is disabled, while iBOT masks 50% of global-view samples with mask ratios in $[0.1, 0.5]$. CODI keeps the protocol unchanged except for the patch target: for each masked teacher patch, we mix the teacher class-token distribution into the original patch-token target with a class-to-patch ratio of 0.1.

C.2 Evaluation Settings

ImageNet k-NN. We evaluate image-level frozen representations with weighted k-NN classification on ImageNet-1K. The training split is used as the retrieval bank, and each validation image is classified by a temperature-weighted vote over its nearest training features. To make the comparison deterministic, both bank and query images are resized on the shorter side to 256, center-cropped to 224×224, converted to tensors, and normalized with ImageNet statistics. We use the last-block CLS feature, apply ℓ_2 normalization, and evaluate with global batch size 512. The standard setting keeps the first nearest neighbor, uses no patch masking, sweeps $k \in \{10, 20, 100, 200\}$ at temperature $\tau = 0.07$, and reports the best top-1/top-5 accuracy over the sweep.

ImageNet linear. We train linear classifiers on top of frozen ImageNet-1K backbones, optimizing only the classification heads. The probe uses the last-block CLS feature without average pooling. Training images are randomly resized and cropped to 224×224, horizontally flipped with probability 0.5, and normalized with ImageNet statistics; validation images are resized to 256 and center-cropped to 224×224. We train the heads for all candidate learning rates jointly for 10 epochs, or 12,500 updates, with SGD, momentum 0.9, zero weight decay, cosine learning-rate decay, and global batch size 1024. The base learning-rate candidates are $\{1 \times 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 2 \times 10^{-2}\}$, scaled in implementation by global batch size divided by 256. We select the head with the best final validation top-1 accuracy and report its top-1/top-5 accuracy.

ImageNet full fine-tuning. For full fine-tuning, we update the entire backbone together with a randomly initialized linear classification head. We use an iBOT-style 100-epoch recipe with 20 warmup epochs, repeated augmentation, global training batch size 512, and global evaluation batch size 1024. Training images are randomly resized and cropped to 224×224, horizontally flipped with probability 0.5, processed by RandAugment with 2 operations and magnitude 9, normalized with ImageNet statistics, and randomly erased with probability 0.25; evaluation uses resizing to 256 followed by center cropping to 224×224. Optimization uses AdamW with learning rate 5×10^{-4}, minimum learning rate 1×10^{-6}, weight decay 0.05, betas $(0.9, 0.999)$, epsilon 1×10^{-8}, layer-wise learning-rate decay 0.65, and drop-path rate 0.1. Mixup and CutMix use $\alpha = 0.8$ and $\alpha = 1.0$, respectively, with mixup/CutMix probability 1.0, switch probability 0.5, and label smoothing 0.1. These hyperparameters are not tuned; we report the top-1/top-5 accuracy of the best validation top-1 checkpoint.

Linear semantic segmentation. We evaluate dense transfer with a frozen-backbone linear segmentation probe on COCO, ADE20K, and Pascal VOC. For ViT backbones, the probe reads normalized final-block patch tokens and predicts per-pixel labels with a 1×1 convolution trained by cross-entropy loss. Training images are randomly resized and cropped to 336×336 with scale range $[0.8, 1.0]$, horizontally flipped with probability 0.5, and normalized with ImageNet statistics; validation images are resized to 448×448 and normalized. Each probe is trained for 2000 steps with global batch size 256 using Adam with betas $(0.9, 0.999)$ and zero weight decay. We choose the learning rate independently for each dataset by validation mIoU from $\{1 \times 10^{-4}, 3 \times 10^{-4}, 8 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 3 \times 10^{-3}, 1 \times 10^{-2}\}$, then report mIoU and pixel accuracy.

Linear depth estimation. We evaluate geometric transfer with a frozen-backbone linear depth probe. Dense patch features are passed to a depth head with optional batch normalization followed by a 1×1 convolution that predicts 256 depth bins, which are converted to metric depth by linear binning and linear normalization over bins. The head is trained with SigLoss for 38,400 steps using global batch size 16, AdamW, weight decay 1×10^{-4}, betas $(0.9, 0.99)$, gradient clipping at 35, and a warmup-one-cycle schedule with 12,800 warmup iterations and final division factor 1000. We select

the head learning rate by validation RMSE from $\{1\times10^{-4}, 2\times10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$. Dataset-specific settings follow the standard crops and masks: NYUv2 uses random crop 416×544, max depth 10m, and the NYU Eigen evaluation mask; KITTI uses random crop 352×704, max depth 80m, and the KITTI Garg evaluation mask; SUNRGBD transfer trains on NYUv2 and evaluates on SUNRGBD with max depth 10m. We report RMSE, absolute relative error, and δ_1.

Video object segmentation. We evaluate video object segmentation by propagating masks over frozen dense features. The first-frame annotation provides the only supervision, and later masks are inferred from affinities to the first frame and a queue of previous predicted frames. Features are ℓ_2-normalized before affinity computation; affinities are temperature-scaled, optionally restricted by a spatial neighborhood, sparsified by top-k source patches, and then used to propagate one-hot masks. We select hyperparameters on a 15-video DAVIS train holdout sampled with seed 0, using J&F-Mean as the selection metric, and keep the selected setting unchanged for final evaluation. The holdout candidates are listed in Tab. 9 following DINOv3 [33]. Unless otherwise stated, all runs use a frame short-side resolution of 480, nearest frame alignment, and matrix multiplication chunk size 2048.

Table 9: VOS hyperparameter candidates used in the DAVIS holdout search.

Top-k	Last frames	Neighborhood	Mask shape	Temperature
5	7	12	square	0.2
5	7	12	circle	0.2
5	7	5	square	0.2
5	7	24	square	0.2
5	7	∞	none	0.2
5	7	12	square	0.01
5	7	12	square	0.1
5	7	12	square	0.7
5	4	12	square	0.2
5	10	12	square	0.2
5	15	12	square	0.2
3	7	12	square	0.2
10	7	12	square	0.2
15	7	12	square	0.2
10	15	12	circle	0.1
10	15	24	circle	0.1
10	15	36	circle	0.1
10	15	∞	none	0.1

D Additional Experiments

D.1 Unsupervised Semantic Segmentation

We evaluate CODI and the baseline representations under the STEGO framework [16] on COCO-Stuff-27. As shown in Tab. 10, CODI outperforms DINOv2 by $+1.6$ and $+1.2$ mIoU in the unsupervised and linear-probing settings, respectively. Despite using a larger patch size, CODI also outperforms STEGO with DINO ViT-S/8 by $+1.1$ mIoU in unsupervised segmentation, demonstrating that it learns a stronger dense representation structure.

Table 10: COCO-Stuff-27 validation mIoU under the STEGO protocol.

Method	Unsupervised mIoU (%)	Linear-probe mIoU (%)
PiCIE+H [9]	14.4	14.8
STEGO (DINO ViT-S/8)	24.5	38.3
DINOv2	24.0	39.4
CODI	**25.6**	**40.6**

D.2 Object Detection and Instance Segmentation

We evaluate our models on COCO object detection and instance segmentation following the DINO and iBOT evaluation protocols [3, 48]. As shown in Tab. 11, CODI achieves the best performance across all reported metrics, indicating that its pretrained representation provides a stronger initialization for object detection and instance segmentation.

Table 11: Object detection and instance segmentation performance on COCO.

Method	Architecture	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Supervised	Swin Transformer	48.1	67.1	52.5	41.7	64.4	45.0
MoBY [42]	Swin Transformer	48.1	67.1	52.1	41.5	64.0	44.7
Supervised	ViT-S/16	46.2	65.9	49.6	40.1	62.9	42.8
DINO	ViT-S/16	48.4	67.6	52.4	42.0	64.8	45.1
iBOT	ViT-S/16	49.4	68.7	53.3	42.6	65.6	45.8
DINOv2	ViT-S/14	49.9	69.0	53.8	43.0	66.0	46.4
CODI (ours)	ViT-S/14	**50.4**	**69.4**	**54.5**	**43.6**	**66.7**	**47.2**

D.3 Results on Smaller Models

To validate CODI on smaller models, we pretrain a ViT-T/14 model, where T denotes Tiny, on ImageNet-1K for 200 epochs. As shown in Tab. 12, CODI outperforms DINOv2 by $+0.3$ in ImageNet classification and $+3.2$ mIoU in VOC segmentation, demonstrating that CODI generalizes to smaller models.

Table 12: ViT-T/14 results after 200 epochs of ImageNet-1K pretraining.

Method	ImageNet k-NN (%)	VOC mIoU (%)
DINOv2	63.2	51.6
CODI	**63.5**	**54.8**

D.4 Comparison with DINOv3 and SigLIP2

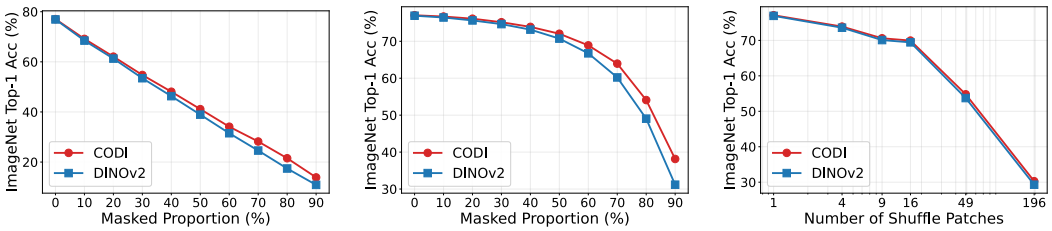
For the aligned Gram-anchoring comparison in Sec. 5.3, we tune the resolution in $\{224 \times 224, 448 \times 448\}$, whether to distill from the previous best or final checkpoint, and whether to update the Gram-anchor teacher, and report the best result. These results are obtained by fine-tuning a DINOv2 checkpoint for 100K iterations with DINOv3's Gram anchoring.

We evaluate the distilled DINOv3 [33] and SigLIP2 [35] checkpoints under the same frozen-feature classification and semantic segmentation protocols and compare them with CODI ViT-Large in Tab. 13. These results are provided for reference, with the architecture and pretraining dataset reported for each model. CODI outperforms SigLIP2 by $+6.1$ mIoU while using substantially less data and no distillation from a larger model, highlighting the potential of our design.

Table 13: Reference comparison with distilled DINOv3 and SigLIP2 checkpoints. Pretraining data and architectures differ across models.

Method	Architecture	Dataset	Epochs	ImageNet k-NN (%)	VOC mIoU (%)
DINOv3 (distilled)	ViT-L/16	LVD-1689M	–	**85.3**	**85.1**
SigLIP2 (distilled)	ViT-L/16	WebLI-10B	–	84.1	74.0
CODI	ViT-L/14	IN1K	200	81.2	80.1

We also pretrain DINOv3 from scratch on ImageNet-1K using RoPE, a patch size of 16, and DINOv3's improved schedule and LayerNorm design. After 100 epochs, DINOv3 ViT-S/16 obtains



(a) Attention occlusion (b) Random occlusion (c) Patch shuffling

Figure 8: Robustness analysis on ImageNet k-NN classification under patch-level perturbations.

70.9 ImageNet k-NN accuracy and 61.3 VOC mIoU, compared with 71.1 and 61.4 for DINOv2 ViT-S/14. We hypothesize that these DINOv3 design choices, developed for large-scale pretraining, require further hyperparameter tuning for small models trained from scratch on datasets at the scale of ImageNet-1K, and leave this adaptation to future work.

D.5 Additional Experiments on Model Robustness

We further evaluate the robustness of CODI under patch-level perturbations. Specifically, we measure ImageNet k-NN classification accuracy under three complementary stress tests that remove representative evidence, remove random visual evidence, or disrupt spatial structure.

Robustness on Attention-based Patch Occlusion. In this setting, we first compute the `[CLS]`-to-patch attention score for each patch. We then mask the patches with the highest attention scores and use the k-NN classifier to evaluate the resulting representations. This protocol tests whether the model can preserve recognition performance when the most representative patches are not visible.

Robustness on Random Patch Occlusion. Instead of masking high-attention patches, we also evaluate the model when randomly selected patches are masked. This protocol measures robustness to input-agnostic information loss.

Robustness on Random Patch Shuffling. Following [25], we shuffle the image using different grid sizes. A grid size of 1 keeps the image unchanged, while a grid size of 196 shuffles every individual patch. This protocol measures how well the model preserves classification performance when spatial structure is disrupted.

As shown in Fig. 8, CODI consistently improves robustness across all three stress tests. Under attention-based occlusion, CODI retains a clear advantage when high-attention patches are removed, indicating that its classification representation relies less on a small set of representative patches. Under random occlusion, the advantage becomes larger at high masking ratios, showing that CODI better exploits distributed patch evidence when the image is severely incomplete. Under patch shuffling, CODI remains slightly but consistently stronger as spatial order is increasingly disrupted. These results suggest that CODI learns more distributed and perturbation-tolerant image representations, rather than improving only the clean-input setting.

D.6 Sensitivity to the Mixing Ratio Across Pretraining Datasets

We conduct additional ablations on α when pretraining on COCO and ImageNet-22K. For COCO, we pretrain for 800 epochs. For ImageNet-22K, we exclude the ImageNet-1K subset to avoid overlap with the downstream classification benchmark and use a 100K-iteration schedule. We evaluate ImageNet-1K k-NN classification and VOC semantic segmentation for both pretraining datasets.

As shown in Tab. 14, on COCO, $\alpha = 0.075$ matches the k-NN performance obtained with $\alpha = 0.05$, while its segmentation performance lies between those obtained with $\alpha = 0.05$ and $\alpha = 0.1$, rather than improving both metrics. This indicates a stable range of $[0.05, 0.1]$, with the final choice depending on whether image-level or dense performance is prioritized. Increasing α to 0.2 reduces both metrics below the DINOv2 baseline. On ImageNet-22K, $\alpha = 0.1$ achieves the best results on both metrics. These findings support $\alpha = 0.1$ as a strong starting point, though not a universal optimum for every downstream metric; CODI does not have a uniformly wide operating range.

Table 14: Sensitivity to α across pretraining datasets. ImageNet-22K excludes the ImageNet-1K subset. All downstream results are percentages.

Method	α	ImageNet k-NN	VOC mIoU
COCO pretraining: 800 epochs			
DINOv2	–	41.1	49.2
CODI	0.05	**41.8**	50.8
CODI	0.075	**41.8**	50.9
CODI	0.1	41.6	**51.3**
CODI	0.2	40.9	48.4
ImageNet-22K pretraining: 100K iterations			
DINOv2	–	63.2	66.1
CODI	0.05	63.5	67.9
CODI	0.1	**63.8**	**68.2**
CODI	0.2	63.4	67.2

E Qualitative Results

We provide additional qualitative results to complement the quantitative dense-prediction evaluations. We group the visualizations by task: semantic segmentation visualizes the quality of learned patch semantics, depth estimation examines whether the learned dense features preserve spatial structure, and sparse correspondence examines local feature alignment across images.

Semantic Segmentation Visualizations. Fig. 9 provides additional frozen-backbone linear segmentation results. The baseline predictions often contain fragmented object regions or background activations. CODI produces more coherent object masks in these examples, indicating that cross distillation improves the semantic quality of patch representations.

Depth Estimation Visualizations. Fig. 10 shows additional depth predictions from frozen dense representations. Although depth estimation is less directly tied to discriminative semantics than segmentation, the visualizations show that CODI preserves coherent indoor layouts and object boundaries. This suggests that the additional class-token guidance does not sacrifice local geometric structure when improving dense semantic transfer.

Sparse Correspondence Visualizations. We further include sparse correspondence results in Fig. 11 to examine whether local features remain aligned across different images of related concepts. The matched points often connect semantically consistent object parts despite changes in instance, viewpoint, scale, and background, indicating that CODI learns patch representations that support transferable local matching.

F Limitations

CODI is motivated by ImageNet-style pretraining views where a dominant object often exists. In highly multi-object scenes, the `[CLS]` target can mix semantics from multiple objects, so overly strong class-token mixing may confuse patch targets. This is consistent with our ablation showing that large mixing ratios can reduce patch discriminability. Additional experiments on COCO and ImageNet-22K support $\alpha = 0.1$ as a strong starting point, but not a universal optimum or evidence of a uniformly wide operating range (Appendix D.6). ImageNet-22K pretraining uses only 100K iterations; full-scale training on substantially larger corpora and adaptive or multi-object-aware mixing remain future work.



Figure 9: Additional semantic segmentation visualizations from frozen-backbone linear probes. For each block, rows from top to bottom show the input image, iBOT, DINOv2, and CODI, while columns show different examples. Colored regions denote predicted semantic labels. CODI produces more complete and less fragmented object regions in these cases.

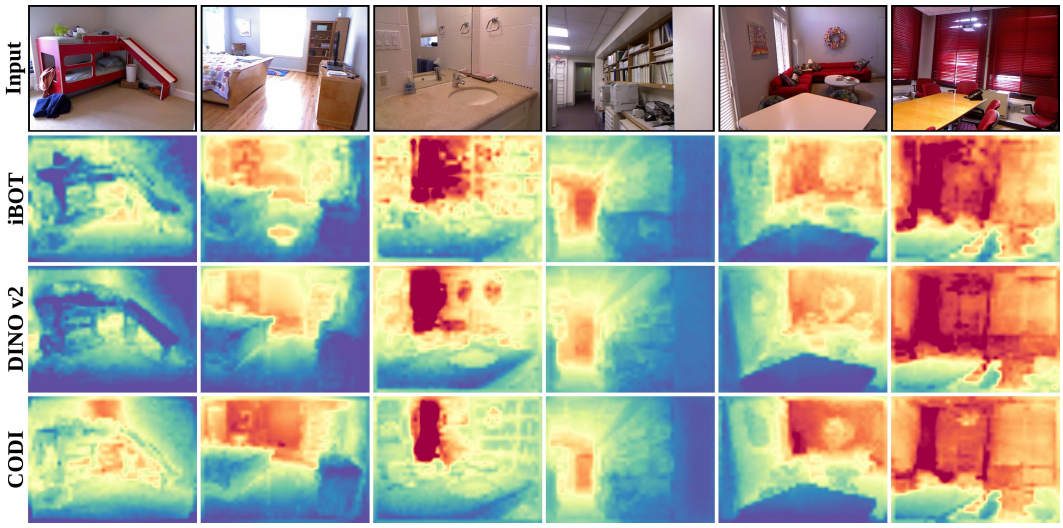


Figure 10: Additional depth estimation visualizations from frozen-backbone linear depth probes. Rows from top to bottom show the input image, iBOT, DINOv2, and CODI, while columns show different indoor examples. Colors visualize relative predicted depth. CODI maintains coherent scene layouts and depth discontinuities while improving dense representation quality.

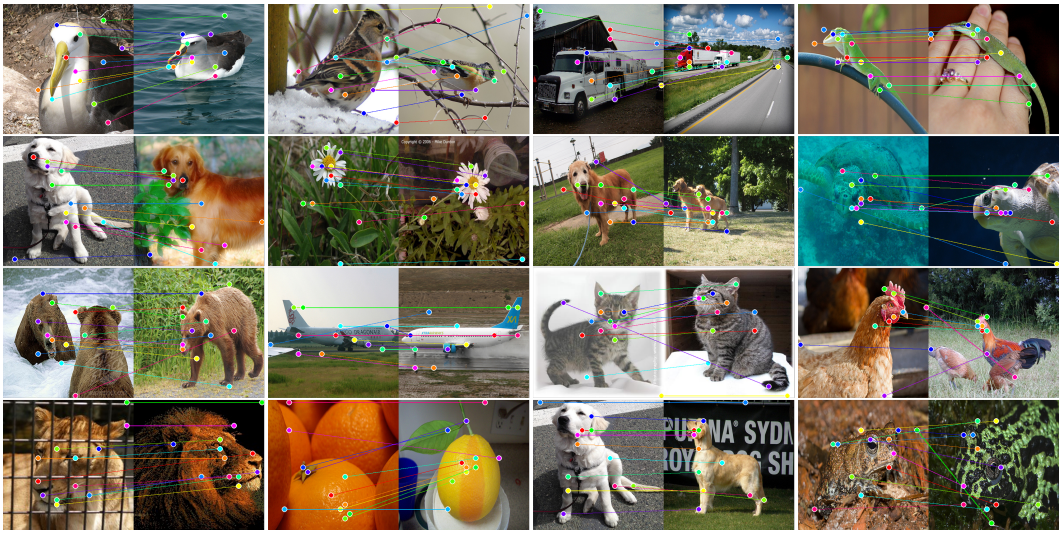


Figure 11: Additional sparse-correspondence visualizations. Colored points and lines indicate matched local features between image pairs. CODI produces many semantically aligned matches across different instances, viewpoints, scales, and backgrounds, suggesting that its dense features generalize beyond the specific image context.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Introduction state the representative-patch failure mode, the CODI target-mixing method, and the empirical scope of dense and image-level evaluations. The claims are supported by the theory in Sec. 3 and Sec. 4 and by the experiments in Sec. 5.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section in Appendix F. It discusses the possible confusion caused by strong class-token mixing on multi-object images, sensitivity to the mixing ratio, and the limited scope of large-scale pretraining experiments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results and assumptions are stated in Sec. 3.2 and Sec. 4.3. Full formal statements and proofs are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The method is specified in Sec. 4 and Algorithm 1, while pretraining and evaluation settings are described in Sec. 5.1 and Appendix C.2. These details include datasets, model variants, optimization settings, augmentation, probe protocols, and hyperparameter sweeps needed to reproduce the reported claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and instructions for reproducing the main experiments are available at https://github.com/EldercatSAM/CODI. The experiments use public datasets, with pretraining and evaluation settings detailed in Appendix C.1 and Appendix C.2.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5.1 specifies the main pretraining setup, including dataset, optimizer, batch size, model scales, epoch counts, learning-rate schedule, and CODI mixing ratio. Appendix C.1 and Appendix C.2 provide detailed pretraining and downstream evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper reports single-run benchmark results and hyperparameter sweeps, but it does not report error bars, confidence intervals, or statistical significance tests. This is mainly due to the high computational cost of repeated large-scale self-supervised pretraining and downstream evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix C.1 reports GPU types and counts and the approximate wall-clock time for the ViT-S/14 800-epoch pretraining run. Memory requirements and execution times for all pretraining and downstream evaluation runs are not fully reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work uses established public vision datasets and standard self-supervised pretraining/evaluation protocols, and it does not involve human subjects, private data collection, or deployment on users. The study is intended as foundational representation-learning research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper is foundational representation-learning work and does not include a separate broader-impact discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk assets such as language models, image generators, or scraped datasets. The proposed method is a training objective for self-supervised vision backbones evaluated on standard benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The datasets, baselines, and model components are credited in Sec. 5.2 and Appendix D. However, the paper does not explicitly enumerate the licenses and terms of use for all existing assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The CODI implementation and accompanying usage and reproduction documentation are provided in the project repository linked in the abstract.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or other research with human subjects. All experiments are conducted on existing vision benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or other research with human subjects. Therefore, IRB approval or an equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing, editing, and formatting, and are not part of the core research methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.