REVISITING JIGSAW OBJECTIVES FOR FINE-GRAINED VISUAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current self-supervised models focus on semantic invariance, capturing similarity of content while ignoring fine-grained variations in the input. This is useful for tasks such as semantic segmentation and retrieval, but physical scene understanding, anomaly detection, and the guidance and evaluation of generative models require representations that capture fine details and spatial structure of the input. This motivates unsupervised objectives that teach models more about the structure of what they encode. Jigsaw-style objectives are a natural candidate. Decomposing an image and asking a model to reassemble it requires attending to boundaries and local structure, and the supervision comes entirely from the decomposition. Such objectives were explored in early self-supervised learning but largely abandoned after contrastive methods proved stronger on classification, and they have not been revisited at the scale or in the architectures now standard. In this paper, we revisit this family in a modern self-distillation framework. We study a range of formulations of the Jigsaw puzzle objective, several of which are new, and evaluate them on several benchmarks. Across tasks requiring fine detail, including object discovery, tracking, and anomaly detection, these objectives outperform prior self-supervised models. Code is available at `https://anonymous.4open.science/r/jigsaw-AE82/`.

1 INTRODUCTION

Self-supervised learning (SSL) has made it possible to learn general-purpose visual representations without human annotation (Zhou et al., 2021; Chen et al., 2020; Caron et al., 2021; Grill et al., 2020; Oquab et al., 2023; Siméoni et al., 2025). Frozen features from these models now support tasks ranging from segmentation and depth estimation to video tracking, and increasingly serve as perceptual representations for generative models and 3D reconstruction pipelines (Yu et al., 2024; Wang et al., 2025a; Chen et al., 2025; Zhu et al., 2024). A central challenge is therefore to determine what information SSL objectives preserve in their representations. Different families of objectives impose different constraints: pretext tasks predict transformations such as rotation, colour, missing regions, or spatial arrangement (Gidaris et al., 2018; Zhang et al., 2016; Pathak et al., 2016; Doersch et al., 2015; Noroozi & Favaro, 2016); joint-embedding methods encourage representations of different views to agree through contrastive learning, clustering, self-distillation, or redundancy reduction (Chen et al., 2020; He et al., 2020; Caron et al., 2018; 2020; Grill et al., 2020; Caron et al., 2021; Zbontar et al., 2021; Bardes et al., 2021); and masked modeling predicts missing pixels, tokens, or latent features (He et al., 2022; Xie et al., 2022; Bao et al., 2021; Baevski et al., 2022; Assran et al., 2023). More recently, these objectives have been combined and scaled into visual foundation models such as iBOT, DINOv2, and DINOv3 (Zhou et al., 2021; Oquab et al., 2023; Siméoni et al., 2025).

Despite their strong performance, modern SSL objectives do not directly supervise spatial arrangement. Joint-embedding objectives encourage invariance between views, which can leave the precise organization of local features weakly constrained. Siméoni et al. (2025) address this issue with Gram anchoring, which helps preserve the relationships between dense features. Masked modeling provides a spatially anchored prediction problem, but its target is the missing content rather than the arrangement of that content. Likewise, recent approaches improve spatial features indirectly by masking boundary regions (Fu et al., 2026) or by asking a student to reproduce teacher-defined patch

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

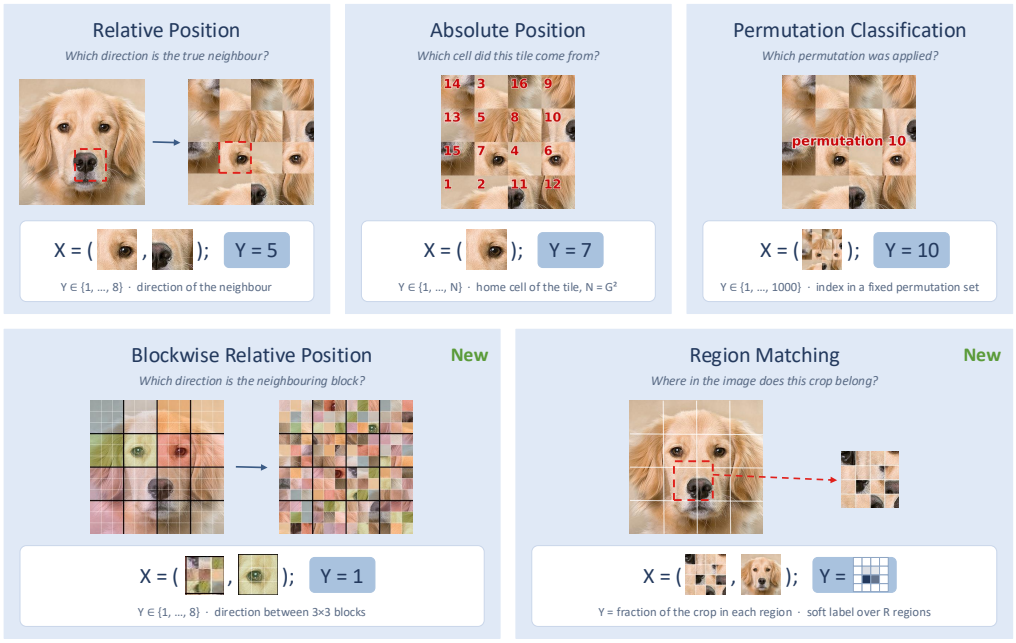


Figure 1: Jigsaw objectives. Each panel shows the puzzle, the input X, and the label Y. **Relative position** (Doersch et al., 2015) predicts the direction of a neighbouring tile among 8 classes. **Absolute position** (Zhai et al., 2022; Wang et al., 2023) predicts the absolute position of each tile. **Permutation classification** (Noroozi & Favaro, 2016; Noroozi et al., 2018) predicts which of 1000 fixed permutations was applied. **Blockwise relative position** (ours) uses two levels of permutation: a 4×4 outer permutation and a 3×3 local permutation within each block, and solves the outer permutation. **Region matching** (ours) shuffles a small crop and predicts where it came from in the full image, using the fraction of the crop in each region as the target.

rankings (Pariza et al., 2025). These methods demonstrate that spatial quality can be improved, but the spatial relationships that the representation should preserve are not themselves the prediction target.

This raises a simple question: *what happens if spatial arrangement is made an explicit self-supervised objective?* Jigsaw learning provides a natural answer. By decomposing an image and asking the model to recover where its pieces belong, the supervision is generated entirely from the image itself while directly specifying a spatial relationship. Importantly, this does not require assuming that jigsaw solving entails high-level scene understanding. A jigsaw can be solved using local edge continuity, colour, texture, or object cues. Rather, the key property of the objective is that these cues must ultimately be used to predict a *spatial relation*: which tile belongs where, how two tiles are positioned relative to one another, or which permutation generated the observed arrangement. This makes jigsaw objectives a controlled way to study whether explicitly supervising spatial arrangement changes the information encoded by visual representations.

Jigsaw objectives were among the earliest forms of self-supervision, including relative position prediction between patch pairs (Doersch et al., 2015), classification of tile permutations (Noroozi & Favaro, 2016; Noroozi et al., 2018), and differentiable permutation recovery (Santa Cruz et al., 2017; Mena et al., 2018). Their use declined as contrastive objectives became dominant, in part because spatial invariance was found to be more useful for recognition than explicitly predicting spatial arrangement (Misra & Maaten, 2020). Position prediction later reappeared in vision transformers: Zhai et al. (2022) showed that transformers trained to recover token positions can learn useful representations even without positional embeddings, while Wang et al. (2023) reconstructed the positions of dropped patch embeddings and reported improvements in classification and dense transfer. These studies suggest that spatial prediction can influence learned representations, but jigsaw formulations

have largely been studied separately, using different architectures, training procedures, and evaluation settings. Consequently, it remains unclear how different forms of spatial supervision compare under a common modern SSL framework, and whether they consistently improve spatially sensitive downstream tasks.

In this paper, we revisit jigsaw objectives in a modern self-distillation framework. We compare relative position, absolute position, permutation classification, and two new formulations, blockwise relative position and region matching, under a common DINOv3 training setup. Across diverse downstream tasks, we find that jigsaw objectives primarily improve spatially sensitive tasks, with different objectives and grid sizes exhibiting distinct trade-offs between spatial detail and object-level representations. These results highlight jigsaw objectives as a direct and controllable form of spatial supervision for modern visual representation learning.

2 Related Work

2.1 Dense Self-Supervised Representations

Existing self-supervised objectives differ in the type of information they ask the model to preserve. Joint-embedding objectives train representations of different views to agree (Chen et al., 2020; He et al., 2020; Caron et al., 2020; 2021; Grill et al., 2020), which enforces semantic invariance in the representation. This is useful for many tasks, but the performance of tasks that depend on fine detail and spatial information can be degraded. Siméoni et al. (2025) observe that the quality of patch tokens can decline over long training even as global features improve, and introduce Gram anchoring, which matches the patch similarity matrix of the student to that of an earlier teacher. Masked modeling masks part of the image and asks the model to predict what was there, as pixels (He et al., 2022; Xie et al., 2022), discrete tokens (Bao et al., 2021), or latent features (Baevski et al., 2022; Assran et al., 2023). Fu et al. (2026) recently showed that the choice of what to mask matters, since random masking hides mostly flat regions that neighboring patches can explain, and concentrating the mask on boundary tokens gives better dense representations. A third group supervises correspondence between local features. Dense contrastive methods match locations across two views of an image (Wang et al., 2021), others treat neighboring patches (Yun et al., 2022) or discovered regions (Hénaff et al., 2022; Wen et al., 2022) as the unit of supervision, and several methods refine an existing backbone by aligning patch features across crops (Ziegler & Asano, 2022), across time (Salehi et al., 2023), or across images (Lebailly et al., 2024). Pariza et al. (2025) rank reference patches drawn from other images by their distance from each patch and train the student to reproduce the teacher's ordering. Matching the ranks teaches the model the relative structure of the patch features, which improves performance on dense tasks.

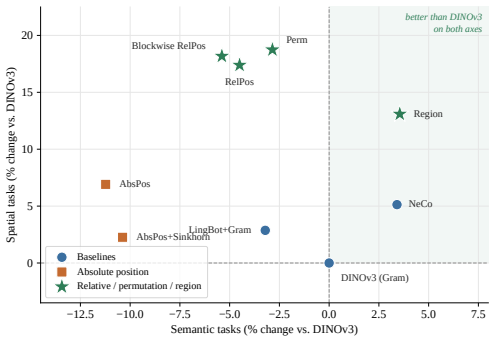


Figure 2: Semantic and spatial performance relative to DINOv3. Each axis is the mean over benchmarks. The semantic axis covers global recognition, in-context segmentation, and dense segmentation, and the spatial axis covers object discovery, saliency, video tracking, point tracking, and 3D correspondence, for nine and ten benchmarks respectively. Most jigsaw objectives improve the spatial axis at a cost on the semantic one, and region matching is the only objective that improves both.

2.2 Jigsaw and Position Prediction Objectives

Jigsaw objectives directly supervise spatial arrangement by decomposing an image into tiles, shuffling them, and asking the model to recover their positions. Early formulations differ in the spatial structure they predict. Doersch et al. (2015) predict the relative position of one tile with respect to another using eight directional classes, while Noroozi & Favaro (2016) formulate the task as classification over a fixed set of global tile permutations. Subsequent work extends these ideas with occluded or missing tiles (Noroozi et al., 2018), joint puzzle and classification training (Car-

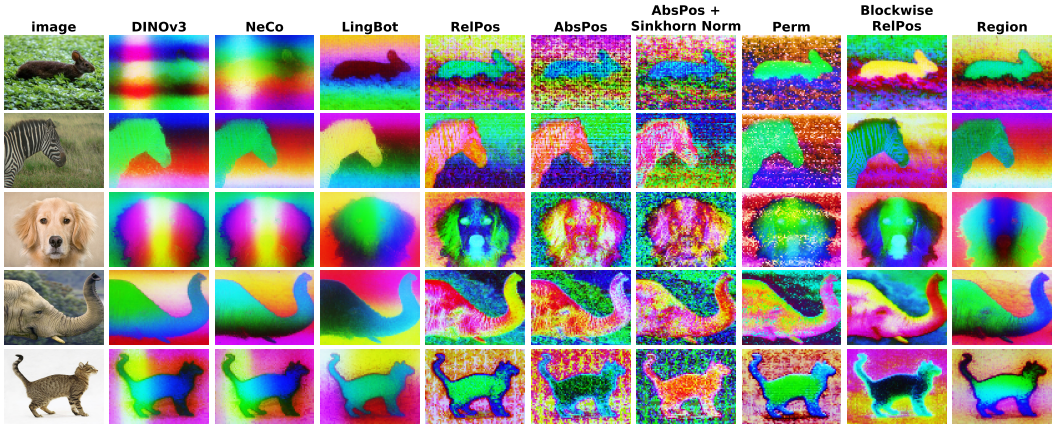


Figure 3: Comparison of dense features. For each panel, patch features are projected to RGB dimensions with PCA. The baselines give smooth maps in which color varies gradually across the object. RelPos, Perm, and Blockwise RelPos separate object from background more sharply and resolve parts within it, while overlaying a periodic texture at the scale of the puzzle cells. The texture is strongest for the absolute-position variants, where it covers the object as well. Region alone is both sharp and clean. All jigsaw objectives use a 12×12 grid.

Table 1: Object discovery, saliency, boundary detection, and anomaly detection. All results use frozen features. Object discovery applies TokenCut Wang et al. (2022) to the patch features and reports CorLoc, saliency reports F_β, boundary detection reports ODS, and anomaly detection follows PatchCore. All jigsaw objectives use a 12×12 grid. Best in **bold**, second best underlined.

Method	Object discovery			Saliency		BSDS500	MVTec-AD		VisA	BTAD
	VOC07 ↑	VOC12 ↑	COCO-20k ↑	DUTS ↑	ECSSD ↑	ODS ↑	AUROC ↑	AUPRO ↑	AUROC ↑	AUROC ↑
Baselines										
DINOv3 Siméoni et al. (2025)	36.28	40.33	29.93	29.75	51.38	59.16	94.03	81.11	84.81	90.17
NeCo Pariza et al. (2025)	36.16	42.44	31.47	29.21	52.62	58.08	94.03	81.13	84.18	90.88
LingBot Fu et al. (2026)	39.73	44.91	33.73	31.26	58.21	55.00	93.08	79.90	84.26	92.28
Jigsaw objectives										
RelPos	44.22	48.47	37.87	40.54	65.40	60.23	94.88	86.21	**86.72**	**92.59**
AbsPos	38.48	42.47	33.27	33.03	56.66	50.29	93.95	83.79	85.30	89.84
AbsPos + Sinkhorn Norm	36.48	40.40	30.61	31.41	56.94	48.36	93.89	83.28	85.27	90.25
Perm	**48.17**	**52.20**	**41.96**	**43.99**	**69.05**	57.30	94.27	83.12	85.31	89.64
Blockwise RelPos	44.28	49.51	38.32	38.82	62.96	**60.28**	**95.51**	**86.28**	84.50	90.65
Region	41.75	46.72	36.18	35.42	59.38	59.06	95.01	82.49	86.40	87.94

lucci et al., 2019), and clustering objectives (Chen et al., 2021). Other approaches recover arbitrary permutations by predicting a soft assignment between tiles and positions (Santa Cruz et al., 2017; Mena et al., 2018). These studies also show that jigsaw tasks can be solved using low-level cues such as edge continuity, chromatic aberration, and intensity statistics (Doersch et al., 2015; Noroozi & Favaro, 2016), motivating augmentations and preprocessing designed to reduce such shortcuts.

With vision transformers, position prediction was reformulated at the token level. Zhai et al. (2022) train transformers to predict the position of each token without positional embeddings, while Wang et al. (2023) recover the positions of patches whose positional information has been removed. More recently, Shen et al. (2026) predict the relative position and scale of one crop with respect to another. Position prediction has also been explored for multimodal language models and reinforcement learning, where spatial reconstruction provides annotation-free training signals (Wu et al., 2026; Wang et al., 2025b). Despite this range of formulations, existing methods use different architectures, training procedures, and evaluation settings, making their effects difficult to compare. We study these objectives under a common modern self-distillation framework, introduce two additional formulations, and evaluate their effects across dense and object-level tasks.

Table 2: Dense probing. A single linear layer is trained on the frozen patch features of the last block for semantic segmentation and monocular depth estimation, and in-context segmentation retrieves labels from a memory of frozen features with nothing trained. Segmentation and in-context segmentation report mIoU, and depth reports RMSE on NYUv2 and KITTI together with δ_1 (%) on NYUv2. Heads are trained for 4k iterations at batch 32, 16k on COCO-Stuff, and 8k at batch 8 for depth, identically for every method, and we report the best checkpoint. All jigsaw objectives use a 12×12 grid, and Blockwise RelPos shuffles 3×3 blocks of tiles, giving 16 movable pieces. Best in **bold**, second best underlined.

Method	Segmentation					Depth			In-context	
	ADE20K ↑	City. ↑	VOC ↑	COCO-St. ↑	P-Ctx ↑	NYUv2 ↓	NYUv2 δ_1 ↑	KITTI ↓	VOC ↑	ADE20K ↑
Baselines										
DINOv3	19.82	47.94	53.78	24.58	36.69	0.650	**67.79**	4.93	36.08	9.50
NeCo	20.70	48.49	54.42	24.73	37.20	**0.649**	67.77	4.94	**38.11**	10.60
LingBot	19.15	45.02	53.45	23.41	35.74	0.702	64.09	**4.92**	36.46	9.73
Jigsaw objectives										
RelPos	19.06	45.76	51.51	23.64	35.23	0.701	62.68	5.53	31.37	8.78
AbsPos	17.85	44.01	49.07	22.87	34.17	0.715	61.22	5.41	26.56	6.90
AbsPos + Sinkhorn Norm	17.98	45.11	50.50	23.50	34.85	0.704	63.01	5.23	27.31	6.62
Perm	19.30	47.50	52.26	23.63	35.73	0.696	63.93	5.21	34.01	9.18
Blockwise RelPos	18.70	45.29	51.47	23.23	34.98	0.755	59.12	5.75	30.55	8.48
Region	**21.09**	**48.67**	**54.66**	**24.92**	**37.33**	0.666	66.01	5.03	36.14	**10.61**

Table 3: Correspondence and tracking. All results use frozen features with nothing trained. Correspondence follows the Probe3D protocol, reporting recall at 2 cm on NAVI for keypoints across views of the same object and recall at 0.1 on SPair across instances of the same category, with our bounding-box normalised PCK@0.1 alongside for reference. Video segmentation propagates the first-frame mask through the sequence by a similarity-weighted vote over patches in preceding frames, and reports $\mathcal{J}\&\mathcal{F}$. Point tracking reports AJ on TAP-Vid. All jigsaw objectives use a 12×12 grid, and Blockwise RelPos shuffles 3×3 blocks of tiles, giving 16 movable pieces. Best in **bold**, second best underlined.

Method	Correspondence			Video segmentation tracking			Point tracking
	SPair rec. ↑	SPair PCK ↑	NAVI ↑	DAVIS ↑	YT-VOS ↑	MOSE ↑	TAP-Vid ↑
Baselines							
DINOv3	24.36	24.35	33.90	49.21	56.90	34.67	19.97
NeCo	**27.71**	**25.84**	34.51	50.82	58.91	35.94	23.30
LingBot	21.38	20.96	32.41	47.65	56.17	34.42	19.99
Jigsaw objectives							
RelPos	21.55	23.70	**34.84**	53.96	59.04	35.59	24.80
AbsPos	17.18	21.77	33.89	51.96	56.25	33.66	23.15
AbsPos + Sinkhorn Norm	14.36	20.42	32.56	50.41	54.51	32.89	21.64
Perm	24.25	24.06	34.56	53.71	59.53	36.18	22.06
Blockwise RelPos	24.43	23.90	34.82	**56.06**	**60.49**	36.36	**25.77**
Region	27.32	25.48	34.70	52.64	59.97	**36.52**	24.49

3 JIGSAW OBJECTIVES

We revisit three jigsaw formulations, relative position, absolute position, and permutation classification, and introduce two new ones, blockwise relative position, and region matching.

Notations: Let $\boldsymbol{x}$ be an input image, divided into a $G \times G$ grid of tiles with $N = G^2$. We index a tile by its original cell position $h \in \{1, \dots, N\}$ and write $g(h) \in \{1, \dots, G\}^2$ for its grid coordinates. A permutation π places the tile whose original position is $\pi(i)$ in the slot i, and the permuted tiles form the shuffled image the student receives. The student encodes the shuffled image and the teacher encodes the intact image. We write $\boldsymbol{s}_i \in \mathbb{R}^D$ for the student tile in slot i and $\boldsymbol{t}_h \in \mathbb{R}^D$ for the teacher tile at position h, both obtained by pooling the tokens they contain, and $\boldsymbol{W}_s, \boldsymbol{W}_t \in \mathbb{R}^{P \times D}$ for the projections applied to each side.

Table 4: Grid size and puzzle structure. Best within each block in **bold**. Gram is shown for reference.

Method	ADE20K	NYUv2		SPair	DAVIS	YT-VOS	VOC07	MVTec	BSDS
	mIoU ↑	RMSE ↓	δ_1 ↑	PCK ↑	$\mathcal{J}\&\mathcal{F}$ ↑	$\mathcal{J}\&\mathcal{F}$ ↑	CorLoc ↑	AUPRO ↑	ODS ↑
DINOv3	19.82	0.65	67.79	24.35	49.21	56.90	36.28	81.11	59.16
(a) Relative position grid size									
3×3	18.09	0.73	61.77	**26.08**	52.36	59.58	36.28	82.54	57.41
4×4	18.92	0.73	61.42	26.01	54.67	**61.32**	43.60	84.24	59.90
6×6	17.90	0.75	59.69	23.60	**55.93**	61.13	44.08	87.03	61.03
9×9	17.99	0.75	59.15	22.66	55.26	59.73	**44.62**	**87.08**	**61.04**
12×12	19.06	0.70	62.68	23.70	53.96	59.04	44.22	86.21	60.23
18×18	**19.11**	**0.68**	**64.48**	22.80	52.12	57.94	40.29	83.16	58.87
(b) Permutation grid size									
3×3	13.31	0.76	59.91	16.61	45.66	51.55	23.57	80.99	52.27
4×4	16.02	0.76	59.91	20.30	44.55	51.92	25.74	82.45	53.26
12×12	**19.30**	**0.70**	**63.93**	**24.06**	**53.71**	**59.53**	**48.17**	**83.12**	**57.30**

Table 5: Effect of the shortcut-suppression augmentations of Noroozi & Favaro (2016). Each pair shows the objective without and with augmentation, and the better value of each pair is in **bold**.

Method	ADE20K	NYUv2		SPair	DAVIS	YT-VOS	VOC07	MVTec	BSDS
	mIoU ↑	RMSE ↓	δ_1 ↑	PCK ↑	$\mathcal{J}\&\mathcal{F}$ ↑	$\mathcal{J}\&\mathcal{F}$ ↑	CorLoc ↑	AUPRO ↑	ODS ↑
RelPos 4×4	18.92	0.73	61.42	26.01	54.67	61.32	**43.60**	**84.24**	**59.90**
+ aug	**21.08**	**0.69**	**64.48**	**28.05**	**54.68**	**61.47**	42.53	83.10	59.36
RelPos 12×12	19.06	0.70	62.68	23.70	**53.96**	59.04	**44.22**	**86.21**	**60.23**
+ aug	**20.56**	**0.65**	**67.42**	**25.24**	51.93	**59.20**	38.56	82.11	58.60
AbsPos	17.85	0.72	61.22	21.77	**51.96**	56.25	38.48	**83.79**	50.29
+ aug	**20.53**	**0.66**	**66.88**	**23.74**	51.38	**57.76**	**39.63**	82.74	**57.38**
Perm 12×12	19.30	0.70	63.93	24.06	**53.71**	**59.53**	**48.17**	**83.12**	57.30
+ aug	**20.64**	**0.66**	**66.57**	**25.26**	52.18	59.50	44.48	82.01	**58.37**
Blockwise	18.70	0.75	59.12	23.90	**56.06**	**60.49**	**44.28**	**86.28**	**60.28**
+ aug	**20.96**	**0.66**	**66.74**	**26.94**	53.22	60.36	40.29	82.73	58.91
Region	**21.09**	0.67	66.01	25.48	**52.64**	**59.97**	**41.75**	**82.49**	59.06
+ aug	21.08	**0.66**	**66.97**	**26.25**	51.95	59.94	39.95	82.16	**59.16**

Training objective. DINOv3 (Siméoni et al., 2025) pretrains with an image-level self-distillation term $\mathcal{L}_{\text{DINO}}$, a patch-level latent reconstruction term $\mathcal{L}_{\text{iBOT}}$, and a Koleo regulariser $\mathcal{L}_{\text{DKoleo}}$ that spreads features within a batch,

$$\mathcal{L}_{\text{SSL}} = \mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + 0.1\,\mathcal{L}_{\text{DKoleo}}. \tag{1}$$

Since dense features degrade over long training, a refinement phase adds Gram anchoring, which pulls the patch similarity matrix of the student towards that of an earlier teacher,

$$\mathcal{L}_{\text{Gram}} = \left\| \boldsymbol{X}_S \boldsymbol{X}_S^\top - \boldsymbol{X}_G \boldsymbol{X}_G^\top \right\|_F^2, \tag{2}$$

where $\boldsymbol{X}_S$ and $\boldsymbol{X}_G$ hold the L2-normalised patch features of the student and of the Gram teacher. We keep this refinement phase and replace Gram anchoring with a jigsaw objective,

$$\mathcal{L} = \mathcal{L}_{\text{SSL}} + \lambda\,\mathcal{L}_{\text{jigsaw}}, \tag{3}$$

so that every model differs only in the term that supervises dense features.

Relative position. For a tile and one of its true neighbors in direction d, where $h \oplus \delta_d$ denotes the neighbor of cell h in direction d,

$$\boldsymbol{z}_{i,d} = \text{MLP}\big([\boldsymbol{W}_s \boldsymbol{s}_i\,;\, \boldsymbol{W}_t \boldsymbol{t}_{\pi(i)\oplus\delta_d}]\big) \in \mathbb{R}^8, \qquad \mathcal{L}_{\text{rel}} = \frac{1}{|\mathcal{V}|} \sum_{(i,d)\in\mathcal{V}} \text{CE}(\boldsymbol{z}_{i,d},\, d), \tag{4}$$

Table 6: Global features. ImageNet columns report top-1 accuracy, with k-NN ($k = 20$) on the frozen CLS token and all other columns from a linear probe trained on ImageNet-1k. ImageNet-C reports mean corruption error. Fine-S averages linear-probe accuracy over 12 datasets. Instance recognition reports mAP, and GAP for Met. All jigsaw objectives use a 12×12 grid, and Blockwise RelPos shuffles 3×3 blocks of tiles, giving 16 movable pieces. Best in **bold**, second best underlined.

Method	ImageNet				Rendition		Hard			Fine-grained	Instance recognition			
	k-NN ↑	Val ↑	V2 ↑	ReaL ↑	R ↑	Sketch ↑	A ↑	C ↓	ObjectNet ↑	Fine-S ↑	Oxford-H ↑	Paris-H ↑	Met ↑	AmsterTime ↑
Baselines														
DINOv3	40.19	49.80	36.69	55.87	14.56	6.07	3.04	116.01	12.30	56.06	1.974	**9.41**	7.41	**14.04**
NeCo	40.15	50.28	37.07	56.55	15.22	**6.38**	**3.29**	113.81	12.39	56.62	1.940	8.96	6.65	13.35
LingBot	37.03	47.90	34.69	53.79	12.37	4.89	2.81	116.66	10.50	52.97	1.576	6.17	1.60	9.18
Jigsaw objectives														
RelPos	40.57	50.41	37.38	56.60	14.32	5.92	3.25	114.72	12.14	57.71	1.965	8.47	7.55	13.50
AbsPos	40.74	49.40	35.91	55.55	13.37	5.31	2.77	116.62	11.46	56.32	1.914	8.58	5.86	13.59
AbsPos + Sinkhorn Norm	**41.12**	50.03	36.61	56.03	13.71	5.60	3.27	116.19	11.88	56.83	**2.245**	8.64	6.38	13.38
Perm	39.61	50.43	37.57	56.46	14.97	5.85	3.24	113.40	11.88	57.22	1.969	7.95	4.53	12.35
Blockwise RelPos	41.00	50.98	**38.53**	57.06	14.78	5.89	3.19	112.95	12.28	**58.77**	1.809	8.46	6.56	13.16
Region	41.07	**51.11**	38.00	**57.28**	**15.42**	6.32	**3.29**	**111.93**	**13.12**	58.55	1.951	8.81	**8.08**	13.56

where $\mathcal{V}$ excludes pairs whose neighbour falls outside the grid.

Absolute position. The model predicts which cell each tile came from by matching it against the teacher's view of the intact image. The teacher tile $\boldsymbol{t}_h$ is the content at cell h, the predicted cell is the one with the highest similarity. Both sides are projected and compared by a scaled dot product,

$$\ell_{i,h} = \frac{(\boldsymbol{W}_s \boldsymbol{s}_i)^\top (\boldsymbol{W}_t \boldsymbol{t}_h)}{\sqrt{P}}. \tag{5}$$

The alternative is a head that classifies each tile into one of N cells, which requires the cell to be recoverable from the tile representation alone. Matching against the teacher instead asks the student to produce tile representations that agree with the teacher's representation of the same content, and the teacher has seen the intact image with its tiles in place. The objective therefore aligns with the other self-distillation terms in DINOv3.

A one-hot target would treat every incorrect cell as equally wrong, so we give cell h a weight that decreases with its distance from the correct cell $\pi(i)$,

$$q_{i,h} = \frac{\exp\big(-\|g(h) - g(\pi(i))\|^2/\sigma^2\big)}{\sum_{h'} \exp\big(-\|g(h') - g(\pi(i))\|^2/\sigma^2\big)}, \qquad \mathcal{L}_{\text{abs}} = -\frac{1}{N} \sum_i \sum_h q_{i,h} \, \log \operatorname{softmax}_h(\ell_{i,\cdot}), \tag{6}$$

where σ sets the width of the target and gives the one-hot case as $\sigma \to 0$. Each tile chooses a cell independently, so several tiles can claim the same one, although the true answer assigns exactly one tile to each cell. The Sinkhorn Santa Cruz et al. (2017) variant enforces this by normalizing ℓ over rows and columns alternately in log space until every tile and every cell hold the unit mass, giving a matrix $\boldsymbol{Q}$ and the loss

$$\mathcal{L}_{\text{sink}} = -\frac{1}{N} \sum_i \log Q_{i,\pi(i)}. \tag{7}$$

Permutation classification. The permutation is drawn from a fixed set $\mathcal{P} = \{\pi_1, \ldots, \pi_K\}$, with $\pi = \pi_y$ and $y \sim \mathcal{U}\{1, \ldots, K\}$. The tile vectors are concatenated in slot order and passed to a head $f : \mathbb{R}^{ND} \to \mathbb{R}^K$ that predicts which permutation was applied,

$$\mathcal{L}_{\text{perm}} = \mathrm{CE}\big(f\big([\boldsymbol{s}_1; \ldots; \boldsymbol{s}_N]\big),\, y\big). \tag{8}$$

Unlike the other objectives, this one uses the student alone, since the arrangement is identified from the tiles themselves.

Blockwise relative position. Cell positions are grouped into $(G/\beta)^2$ blocks of $\beta \times \beta$ cells, indexed by m. Tiles are permuted within their block, and the blocks are permuted among themselves, so the model solves a coarse puzzle over blocks. Both sides are pooled within each block. The student pools by position in the shuffled image and the teacher by position in the intact image,

$$\bar{\boldsymbol{s}}_m = \frac{1}{\beta^2} \sum_{i \in \mathcal{S}_m} \boldsymbol{s}_i, \qquad \bar{\boldsymbol{t}}_m = \frac{1}{\beta^2} \sum_{h \in \mathcal{B}_m} \boldsymbol{t}_h, \tag{9}$$

where $\mathcal{S}_m$ are the slots of block m in the shuffled image and $\mathcal{B}_m$ the cells of block m in the intact image. Blocks are paired by their shuffled position, and the direction is predicted as in $\mathcal{L}_{\text{rel}}$,

$$\mathcal{L}_{\text{block}} = \frac{1}{|\mathcal{V}_B|} \sum_{(m,d)\in\mathcal{V}_B} \text{CE}\Big(\text{MLP}\big([\boldsymbol{W}_s \bar{\boldsymbol{s}}_m \,;\, \boldsymbol{W}_t \bar{\boldsymbol{t}}_{\rho(m)\oplus\delta_d}]\big),\, d\Big), \tag{10}$$

where ρ is the permutation over blocks. This is not the same as relative position on a coarser grid. With a $(G/\beta)\times(G/\beta)$ grid, each unit is a single tile whose content is spatially intact and encoded as one region. Here each unit is an average of β^2 small tiles that have been permuted among themselves, so the block carries the content of its region without its internal arrangement, and the model has to place it from content alone.

Region matching. The image is partitioned into R regions $\mathcal{R}_1, \ldots, \mathcal{R}_R$, each a contiguous group of cells. A crop $\mathcal{C}$ is taken from the image, cut into n_c tiles, shuffled, and encoded by the student, giving tile vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_c}$. The model predicts which region the crop came from. Each crop tile is scored against every teacher tile, and the scores are summed within each region,

$$r_k = \frac{1}{n_c} \sum_{j=1}^{n_c} \sum_{h\in\mathcal{R}_k} \frac{(\boldsymbol{W}_s \boldsymbol{u}_j)^\top (\boldsymbol{W}_t \boldsymbol{t}_h)}{\sqrt{P}}. \tag{11}$$

The crop spans more than one region, so the target is the fraction of its area that falls in each,

$$p_k = \frac{\text{area}(\mathcal{C}\cap\mathcal{R}_k)}{\text{area}(\mathcal{C})}, \qquad \mathcal{L}_{\text{reg}} = -\sum_k p_k \log \text{softmax}_k(\boldsymbol{r}). \tag{12}$$

4 Experimental Setup

Training. All models use a ViT-L/16 backbone and the DINOv3 recipe. We pretrain once on ImageNet-1k for 100k iterations with batch size 64, then refine each model from that checkpoint for a more 20k training iteration. Unless stated otherwise, jigsaw objectives use a 12×12 grid over a 576-pixel view of 36×36 patches, with three-patch cells holding a two-patch tile and a one-patch gap, and the loss weight $\lambda = 0.3$. Projection heads are 256-dimensional and trained without weight decay. The teacher's pass over the intact view is shared with the DINO and iBOT terms, so each objective adds one student forward and backward per step. We also vary the relative-position grid over 3×3 to 18×18, adjusting tile size so the source region stays fixed, and the permutation grid over 3×3, 4×4, and 12×12 results shown in table 4. Each objective is also trained with and without the shortcut-suppression augmentations shown in table 5, everything else held fixed. All other training and implementation details are provided in Appendix A.

Baselines. We compare against three baselines, all trained using the same DINOv3 codebase and settings where applicable: DINOv3 (Siméoni et al., 2025), which uses Gram anchoring; NeCo (Pariza et al., 2025), which supervises patch-neighbour ordering; and LingBot (Fu et al., 2026), which adds masked boundary modeling to Gram anchoring. Further details on baseline implementations and training are provided in Appendix A.

Evaluation. We evaluate frozen teacher features across dense prediction, correspondence, video segmentation and tracking, object discovery, saliency, boundary detection, anomaly localization, and global recognition. We follow the DINOv3 evaluation protocol, with full details on datasets, metrics, resolutions, probes, and evaluation settings in Appendix A. All results use a single training run per model.

5 Results

Table 1 shows that the effect of spatial supervision depends on the objective and task. Permutation classification performs best on object discovery and saliency, improving CorLoc on VOC07 by 11.9 points and F_β on DUTS by 14.2 points over DINOv3. Relative position and its blockwise variant perform best on anomaly localization, reaching 86.21 and 86.28 AUPRO on MVTec. Region

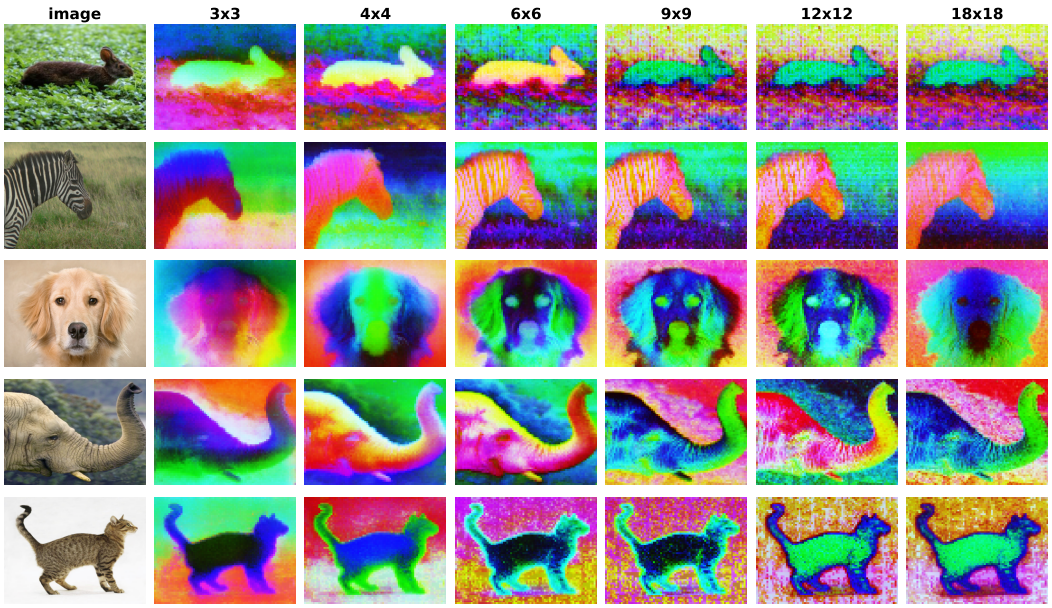


Figure 4: Effect of grid size on dense features. We project the patch features of the relative position objective to three dimensions with PCA and map them to RGB. Columns show grids from 3×3 to 18×18, with the tile size adjusted so that the source region stays fixed. Coarse grids give more detailed patterns and object boundaries, with visible tile structure in the background, and fine grids give smoother maps.

matching consistently improves over the jigsaw baselines, although it does not lead on these tasks. Absolute position is the weakest variant, including on boundary detection.

For dense prediction, region matching is the only jigsaw objective that consistently improves over DINOv3, leading on all five semantic segmentation datasets in Table 2. Correspondence results in Table 3 depend on the type of matching: most objectives improve correspondence across views of the same object on NAVI, whereas region matching is the only jigsaw variant competitive with NeCo on SPair. Video segmentation follows a similar pattern, with blockwise relative position achieving the best results on DAVIS, YouTube-VOS, and TAP-Vid, while region matching leads on MOSE. Jigsaw objectives also preserve global recognition performance, collectively leading on twelve of fourteen benchmarks in Table 6. The effects of the objective also depend on the spatial granularity. As shown in Table 4, relative position favors different grid sizes depending on the task, whereas permutation classification consistently performs best at 12×12. The shortcut-suppression experiments in Table 5 show that removing low-level cues does not uniformly improve performance: it shifts the representation toward dense prediction and away from object-level tasks. Overall, Figure 2 shows a clear trade-off between semantic and spatial performance for most jigsaw objectives, while region matching is the only formulation that improves over DINOv3 on both.

6 Conclusion

We revisited jigsaw objectives in a modern self-distillation framework and systematically compared relative position, absolute position, permutation classification, and two new formulations across semantic and spatial tasks. We find that explicit spatial supervision improves dense and spatially sensitive tasks, but its effect depends strongly on the formulation. Permutation classification favors figure-ground separation, while relative position and its blockwise variant are particularly effective for boundaries, anomaly localization, and tracking. Absolute position provides weaker improvements, whereas region matching is the only formulation that consistently benefits both semantic and spatial tasks. Overall, our results show that jigsaw objectives provide a competitive and controllable way to improve dense visual representations, with the choice of spatial supervision playing a central role.

AI USE STATEMENT

We used generative AI tools to help refine the writing, to set up the evaluation scripts, and to monitor experiments. For writing, we used them to edit prose for clarity and to draft LaTeX for tables and figures, working from results we produced. For engineering, we used them to help write evaluation and plotting code and to help locate related work, which we then read and cited from the primary sources. The core implementation and experimental design are done by the authors. We did not use generative AI to generate research ideas or to produce any experimental result. All code was reviewed and tested by the authors, every number in the paper comes from our own training and evaluation runs, and every citation was checked against the cited paper. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 3 gives the loss for every objective we study, and Section 4 gives the training recipe, the baselines, and the evaluation protocols. Appendix A.2 lists the objective-specific settings, including grid sizes, head architectures, teacher schedules, and the shortcut-suppression augmentations, and Appendix A.3 gives the metrics, resolutions, and probe settings for every benchmark. All models share one pretraining checkpoint and one refinement schedule, so the only difference between them is the objective. Code for the objectives is available at `https://anonymous.4open.science/r/jigsaw-AE82/`. All benchmarks use publicly available datasets.

REFERENCES

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning*, pp. 1298–1312. PMLR, 2022.

Ivana Balazevic, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, and Olivier Henaff. Towards in-context scene understanding. *Advances in Neural Information Processing Systems*, 36: 63758–63778, 2023.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2224–2233. Ieee, 2019.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European conference on computer vision*, pp. 139–156. Springer, 2018.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Pengguang Chen, Shu Liu, and Jiaya Jia. Jigsaw clustering for unsupervised visual representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11521–11530. IEEE, 2021.

Sherry X Chen, Misha Sra, and Pradeep Sen. Instruct-clip: Improving instruction-guided image editing with automated data refinement using contrastive learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28513–28522. IEEE, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21795–21806. IEEE, 2024.

Zelin Fu, Bin Tan, Changjiang Sun, Shaohui Liu, Kecheng Zheng, Yinghao Xu, Xing Zhu, Yujun Shen, and Nan Xue. Vision pretraining for dense spatial perception. *arXiv preprint arXiv:2607.05247*, 2026.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9726–9735. IEEE, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. In *European conference on computer vision*, pp. 123–143. Springer, 2022.

Tim Lebailly, Thomas Stegmüller, Behzad Bozorgtabar, Jean-Philippe Thiran, and Tinne Tuytelaars. Cribo: Self-supervised learning via cross-image object-level bootstrapping. In *International Conference on Learning Representations*, volume 2024, pp. 46794–46814, 2024.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6707–6717, 2020.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9359–9367. IEEE, 2018.

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.

Valentinos Pariza, Mohammadreza Salehi, Gertjan J Burghouts, Francesco Locatello, and Yuki Asano. Near, far: Patch-ordering enhances vision foundation models' scene understanding. In *International Conference on Learning Representations*, volume 2025, pp. 101552–101579, 2025.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14298–14308. IEEE, 2022.

Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16490–16501. IEEE, 2023.

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6044–6052. IEEE, 2017.

Yang Shen, Yusen Cai, Weronika Hryniewska-Guzik, Qing Lin, and Mengmi Zhang. Learning to perceive" where": Spatial pretext tasks for robust self-supervised learning. *arXiv preprint arXiv:2605.09963*, 2026.

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.

Haochen Wang, Junsong Fan, Yuxi Wang, Kaiyou Song, Tong Wang, and ZHAO-XIANG ZHANG. Droppos: Pre-training vision transformers by reconstructing dropped positions. *Advances in Neural Information Processing Systems*, 36:46134–46151, 2023.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3023–3032. IEEE, 2021.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14523–14533. IEEE, 2022.

Zifu Wang, Junyi Zhu, Bo Tang, Zhiyu Li, Feiyu Xiong, Jiaqian Yu, and Matthew B Blaschko. Jigsaw-r1: A study of rule-based visual reinforcement learning with jigsaw puzzles. *arXiv preprint arXiv:2505.23590*, 2025b.

Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. *Advances in neural information processing systems*, 35:16423–16438, 2022.

Penghao Wu, Yushan Zhang, Haiwen Diao, Bo Li, Lewei Lu, and Ziwei Liu. Visual jigsaw post-training improves mllms. In *International Conference on Learning Representations*, volume 2026, pp. 93221–93245, 2026.

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 *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9643–9653. IEEE, 2022.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Sukmin Yun, Hankook Lee, Jaehyung Kim, and Jinwoo Shin. Patch-level representation learning for self-supervised vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8344–8353. IEEE, 2022.

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*, pp. 12310–12320. PMLR, 2021.

Shuangfei Zhai, Navdeep Jaitly, Jason Ramapuram, Dan Busbridge, Tatiana Likhomanenko, Joseph Yitan Cheng, Walter Talbott, Chen Huang, Hanlin Goh, and Joshua Susskind. Position prediction as an effective pretraining strategy. *arXiv preprint arXiv:2207.07611*, 2022.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

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.

Yilun Zhu, Jianxin Zhang, Aditya Gangrade, and Clayton Scott. Label noise: Ignorance is bliss. *Advances in neural information processing systems*, 37:116575–116616, 2024.

Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14482–14491. IEEE, 2022.

A Appendix

A.1 Additional Results



Figure 5:

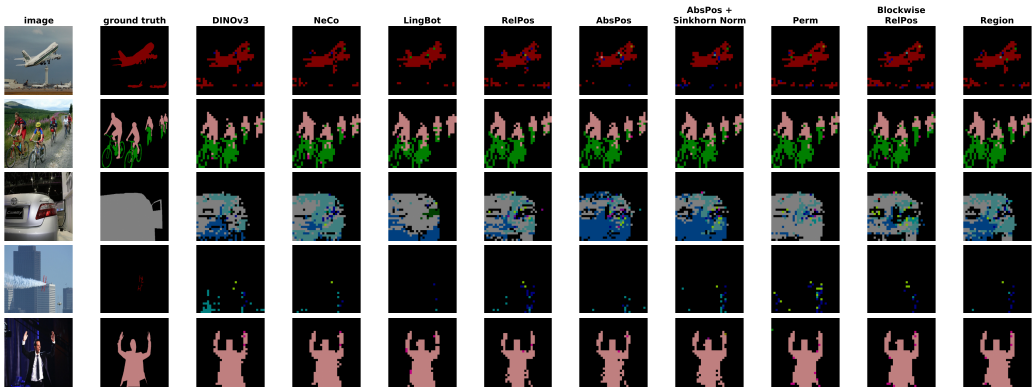


Figure 6:

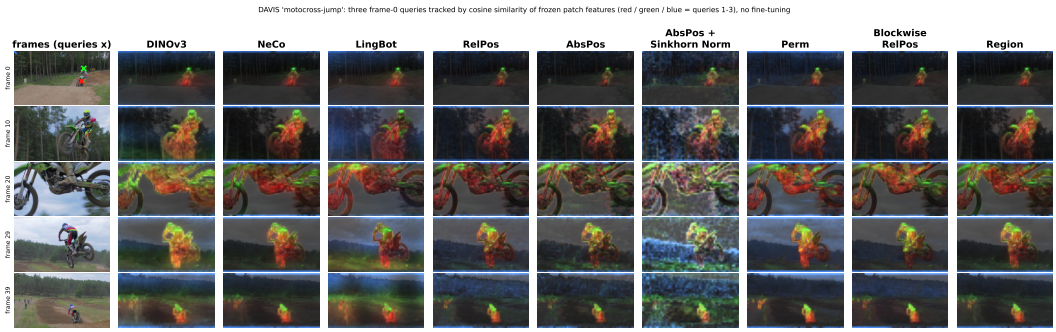


Figure 7:

A.2 Implementation details

Shared setup. All models use a ViT-L/16 backbone and the DINOv3 recipe. A single pretraining run of 100k iterations on ImageNet-1k at batch size 64 provides the common starting point, and

each model is refined from that checkpoint for a further 20k iterations (100k–120k). Every jigsaw objective is weighted $\lambda = 0.3$ against the DINO, iBOT and KoLeo terms, replaces Gram anchoring, and its head is trained with the weight-decay multiplier set to zero. The jigsaw view is a 576-pixel crop, 36×36 patches at patch size 16, cut into a 12×12 grid of three-patch cells. All objectives except Region place a two-patch tile inside each three-patch cell and discard the remaining one-patch gap, which prevents the low-level edge continuation that would otherwise solve the task. Region uses the full three-patch tile. Tile origins are jittered by a random offset each step. The teacher's forward pass over the intact view is shared with the DINO and iBOT terms, so each objective costs one additional student forward and backward per step.

Relative position and its block variants. The teacher used for the relative-position family is a frozen anchor and not the live EMA. The EMA weights are hard-copied in at iteration 100k and refreshed every 1250 iterations from 101 250 onward, the same ratchet Gram anchoring uses. One grid configuration is sampled per step. With `block_size` $= 1$ the student predicts the direction between neighbouring tiles. With `block_size` $= 3$ the 12×12 tiles group into $4 \times 4 = 16$ blocks and supervision moves to block pairs, so the puzzle has 16 movable pieces and not 144. The two block variants differ only in where the grouping lives: Blockwise shuffles all 144 tiles freely and lets the loss restore the blocks, while Blockwise RelPos shuffles blocks as units and permutes tiles only within their own block, putting the structure in the input.

Permutation classification. Following Noroozi & Favaro (2016), the tiles are permuted by one of a fixed set of 1000 permutations chosen to maximise mean Hamming distance (mean distance 8.00 over the set), and the network performs 1000-way classification over that set. Tiles are encoded independently and their features concatenated (4608) before two fully connected layers (4096, then 1000), so no tile sees another through the backbone. This objective is student-only: it uses no teacher branch.

Absolute position and the Sinkhorn variant. The student predicts each tile's absolute index. Targets are smoothed with a Gaussian of width σ over the grid, annealed linearly from $\sigma = 1.0$ at iteration 100k to $\sigma = 0$ at 120k, so the task begins as "roughly where" and ends as "exactly where". The Sinkhorn variant is the same objective with the score matrix projected toward doubly stochastic by 20 Sinkhorn iterations at temperature $\tau = 1.0$, which makes tiles compete for positions in place of being scored independently. It is therefore a variant of absolute position and not of permutation classification.

Region matching. A small crop of four tiles is taken from the teacher's view, cut up, shuffled and encoded. Student crop tiles and teacher tiles are projected to 256 dimensions and scored as $S = \mathrm{proj}_s(\text{crop})\,\mathrm{proj}_t(\text{teacher})^\top/\sqrt{d}$. The 144 teacher columns are collapsed into 16 regions by summing the nine tiles of each region, and the result is trained with a soft cross-entropy against the normalised box overlap between the crop and each region. Nothing is averaged in feature space, so a region is identified by the conjunction of what it contains and not by a mean of it.

Shortcut-suppression augmentations. The augmented experiments of Table 5 add the three countermeasures of Noroozi & Favaro (2016), applied to the jigsaw view only: per-tile normalisation, a random chroma shift of up to 2 pixels per colour channel, and conversion to grayscale with probability 0.3. Everything else is held fixed.

A.3 Evaluation Setup

All evaluations use frozen features from the teacher network. Only dense probing and the ImageNet-1k and Fine-S linear probes train anything on top of the backbone, and in every case that is a single linear layer; the remaining protocols train nothing.

Dense probing. A single linear layer is trained on the last-layer patch tokens. Semantic segmentation covers ADE20K, Cityscapes, VOC, COCO-Stuff and Pascal-Context and reports mIoU, evaluated at 512×512 except Cityscapes at 1024×1024. Monocular depth covers NYUv2 and KITTI and reports RMSE, with δ_1 also reported on NYUv2. Heads are trained for 4k iterations at

Table 7: Objective-specific settings. All objectives use a 12×12 grid over a 576-pixel view, $\lambda = 0.3$, and heads trained without weight decay. The shuffled unit is what the student's view is permuted over, and the supervised unit is what the loss asks about.

Objective	Prediction target	Shuffled / supervised unit	Settings
RelPos	8-way direction between a student tile and one true teacher neighbour	tile / tile	proj. 256; 3-layer head, hidden 512; frozen teacher
AbsPos	absolute tile index, label-smoothed	tile / tile	proj. 256; σ annealed $1.0 \to 0$ over 100k–120k
AbsPos + Sinkhorn Norm	absolute tile index, scores projected to a doubly stochastic matrix	tile / tile	as AbsPos; 20 Sinkhorn iterations, $\tau = 1.0$
Perm	which of 1000 permutations was applied	tile / whole puzzle	context-free encoding, concat $4608 \to 4096 \to 1000$
Blockwise RelPos	8-way direction between adjacent 3×3 blocks of tiles	3×3 block / block	as RelPos; tiles move only within their block, blocks move as units
Region	which of 16 regions an unpooled crop came from	tile / 3×3 region	proj. 256; 4 crop tiles; jitter $p = 0.8$, grayscale $p = 0.2$

batch size 32, 16k iterations on COCO-Stuff, and 8k iterations at batch size 8 for depth, identically for every model, and we report the best checkpoint.

In-context segmentation. We follow the Hummingbird protocol (Balazevic et al., 2023) on VOC and ADE20K. A memory bank is built by sampling 512 patch features from each of 2000 training images; every validation patch is labelled by a temperature-weighted vote ($\tau = 0.02$) over its $k = 30$ nearest bank entries, and the result is upsampled to the image and scored as mIoU. Nothing is trained.

Correspondence. We follow Probe3D (El Banani et al., 2024). NAVI matches keypoints across views of the same object and reports recall at a 2 cm threshold. SPair matches across instances of the same category and reports recall at 0.1, and we report our bounding-box normalised PCK@0.1 alongside it. As in Probe3D, the ratio test weights candidate matches rather than filtering them, and each pair contributes a fixed number of correspondences.

Video segmentation and point tracking. The first-frame mask is propagated through the sequence by a similarity-weighted vote over patches in preceding frames, following DINOv3 (Siméoni et al., 2025), on DAVIS-2017, YouTube-VOS and MOSE at a short side of 480 pixels. Each patch votes over its $k = 5$ most similar patches within a spatial radius of 12 patches, at temperature 0.2, using the first frame plus the six preceding frames as context. We report $\mathcal{J}\&\mathcal{F}$. Point tracking reports AJ on TAP-Vid, at the occlusion threshold that maximises AJ for each model, swept over $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.

Object discovery, saliency and boundaries. Object discovery applies TokenCut (Wang et al., 2022) to the patch features on VOC07, VOC12 and COCO-20k and reports CorLoc, counting a prediction correct when its box overlaps a ground-truth box by IoU ≥ 0.5. Saliency reports F_β with $\beta^2 = 0.3$ on DUTS and ECSSD from the same bipartition, and boundary detection reports ODS on BSDS500. The TokenCut threshold τ is swept over $\{0.0, \ldots, 0.8\}$ and the best value reported for each model, as is standard for ODS, so these columns compare each model at its own operating point.

Anomaly detection. We follow PatchCore (Roth et al., 2022). For each category, patch features from layers 12 and 18 of the normal training images at 448 pixels form a coreset memory bank retaining 1% of patches, and a test patch is scored by its distance to the three nearest entries. MVTec-AD reports image-level AUROC and AUPRO; VisA and BTAD report image-level AUROC.

Global features. Classification features are taken at a shorter side of 512 with a centre crop, and retrieval descriptors at 224 pixels. k-NN classification uses the frozen CLS token with $k = 20$ and temperature 0.07. A linear probe is trained on ImageNet-1k with its learning rate selected per model on the ImageNet validation set, and that same classifier is then evaluated on ImageNet-V2, ReaL, R, Sketch, A, C and ObjectNet without refitting. ImageNet columns report top-1 accuracy; ImageNet-C reports mean corruption error, each corruption normalised by AlexNet's error rate on it and averaged over five severities and fifteen corruptions. Fine-S averages accuracy over twelve fine-grained datasets (Food101, CIFAR-10, CIFAR-100, SUN397, Cars, Aircraft, VOC2007, DTD, Pets, Caltech101, Flowers, CUB), each fit with L2 logistic regression whose regularisation is chosen on a validation split and refit on train and validation together. Instance retrieval reports mAP on revisited Oxford and Paris (Hard) and on AmsterTime, and GAP on Met, where each query takes the exhibit label of its nearest database image and queries are ranked by that cosine similarity.