DINOv3

Oriane Siméoni* **Huy V. Vo*** **Maximilian Seitzer*** **Federico Baldassarre*** **Maxime Oquab***
Cijo Jose **Vasil Khalidov** **Marc Szafraniec** **Seungeun Yi** **Michaël Ramamonjisoa**
Francisco Massa **Daniel Haziza** **Luca Wehrstedt** **Jianyuan Wang**
Timothée Darcet **Théo Moutakanni** **Leonel Sentana** **Claire Roberts**
Andrea Vedaldi **Jamie Tolan** **John Brandt**[1] **Camille Couprie**
Julien Mairal[2] **Hervé Jégou** **Patrick Labatut** **Piotr Bojanowski**

Meta AI Research [1] *WRI* [2] *Inria*

* *corresponding authors:* {osimeoni,huyvvo,seitzer,baldassarre,qas}@meta.com

Presented by:

John Tan Chong Min

Key Question: How can we learn from massive unlabelled image / video data?

Self-Supervised Learning Learns without Labels

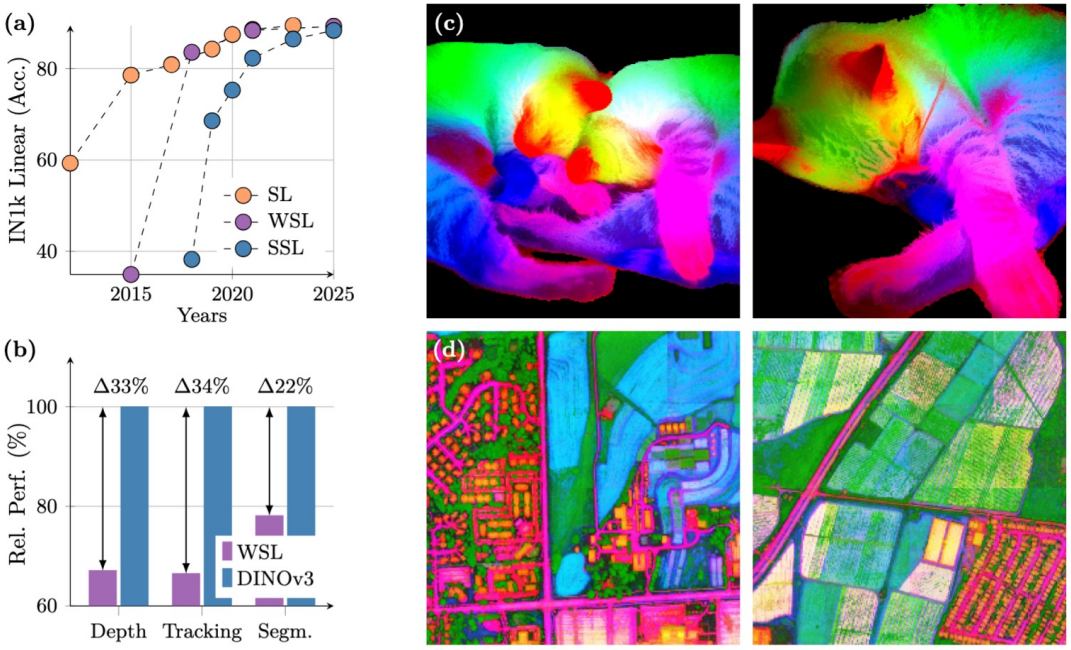
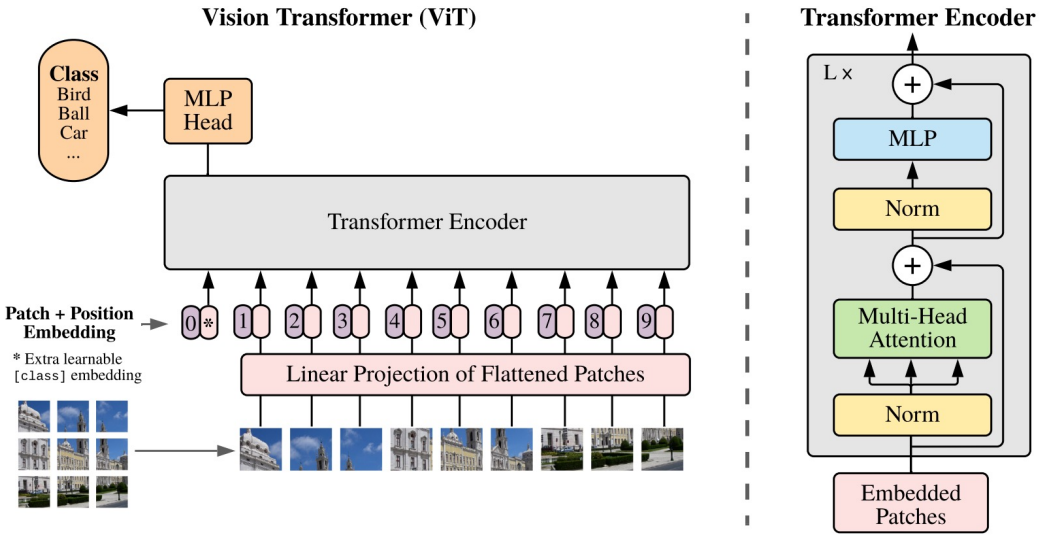


Figure 1: (a) Evolution of linear probing results on ImageNet1k (IN1k) over the years, comparing fully- (SL), weakly- (WSL) and self-supervised learning (SSL) methods. Despite coming into the picture later, SSL has quickly progressed and now reached the Imagenet accuracy plateau of recent years. On the other hand, we demonstrate that SSL offers the unique promise of high-quality dense features. With DINOv3, we markedly improve over weakly-supervised models on dense tasks, as shown by the relative performance of the best-in-class WSL models to DINOv3 (b). We also produce PCA maps of features obtained from high resolution images with DINOv3 trained on natural (c) and aerial images (d).

- Supervised Learning – Human Labels
- Weakly Supervised Learning – Labels are from metadata
- Self-Supervised Learning – No labels

Vision Transformer (ViT): Can we process images like word tokens?



TLDR: Possible to learn from images by splitting them up into patches and encoding them as tokens

Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

An image is worth 16x16 words. Transformers for Image Recognition at Scale. Dosovitskiy et al. 2021.

Key Takeaway:

"A single frozen SSL backbone can serve as a universal visual encoder that achieves state-of-the-art performance on challenging downstream tasks, outperforming supervised and metadata-reliant pre-training strategies"

Brief Overview of DINO: Teacher-student learning

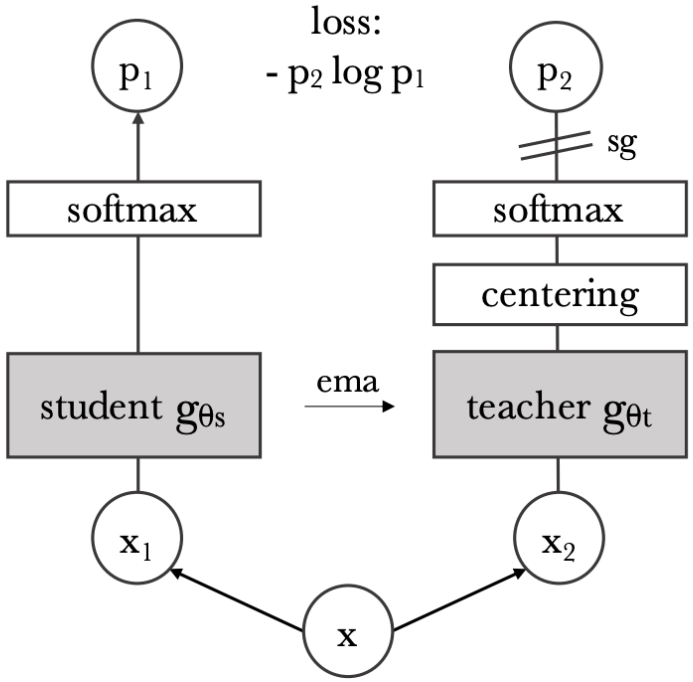


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

Emerging Properties in Self-Supervised Vision Transformers (DINO). Caron et al. 2021.

Key Improvements of DINOv3: Gram anchoring to earlier version of model

The DINOv3 Family of Models Solving the degradation of dense feature map with Gram anchoring unlocks the power of scaling. As a consequence, training a much larger model with SSL leads to significant performance improvements. In this work, we successfully train a DINO model with 7B parameters. Since such a large model requires significant resources to run, we apply distillation to compress its knowledge into smaller variants. As a result, we present the *DINOv3 family of vision models*, a comprehensive suite designed to address a wide spectrum of computer vision challenges. This model family aims to advance the state of the art by offering scalable solutions adaptable to diverse resource constraints and deployment scenarios. The distillation process produces model variants at multiple scales, including Vision Transformer (ViT) Small, Base, and Large, as well as ConvNeXt-based architectures. Notably, the efficient and widely adopted ViT-L model achieves performance close to that of the original 7B teacher across a variety of tasks. Overall, the DINOv3 family demonstrates strong performance on a broad range of benchmarks, matching or exceeding the accuracy of competing models on global tasks, while significantly outperforming them on dense prediction tasks, as visible in Fig. 2.

Data Curation is very important

Data Collection and Curation We build our large-scale pre-training dataset by leveraging a large data pool of web images collected from public posts on Instagram. These images already went through platform-level content moderation to help prevent harmful contents and we obtain an initial data pool of approximately 17 billions of images. Using this raw data pool, we create three dataset *parts*. We construct the first part by applying the automatic curation method based on hierarchical k-means from Vo et al. (2024). We employ DINOv2 as image embeddings, and use 5 levels of clustering with the number of clusters from the lowest to highest levels being 200M, 8M, 800k, 100k, and 25k respectively. After building the hierarchy of clusters, we apply the balanced sampling algorithm proposed in Vo et al. (2024). This results in a curated subset of 1,689 million images (named LVD-1689M) that guarantees a balanced coverage of all visual concepts appearing on the web. For the second part, we adopt a retrieval-based curation system similar to the procedure proposed by Oquab et al. (2024). We retrieve images from the data pool that are similar to those from selected seed datasets, creating a dataset that covers visual concepts relevant for downstream tasks. For the third part, we use raw publicly available computer vision datasets including ImageNet1k (Deng et al., 2009), ImageNet22k (Russakovsky et al., 2015), and Mapillary Street-level Sequences (Warburg et al., 2020). This final part allows us to optimize our model's performance, following Oquab et al. (2024).

DINOv3 has better feature map representation

Figure 3: High-resolution dense features. We visualize the cosine similarity maps obtained with DINOv3 output features between the patches marked with a red cross and all other patches. Input image at 4096×4096. *Please zoom in*, do you agree with DINOv3?

Feature map can capture high resolution details

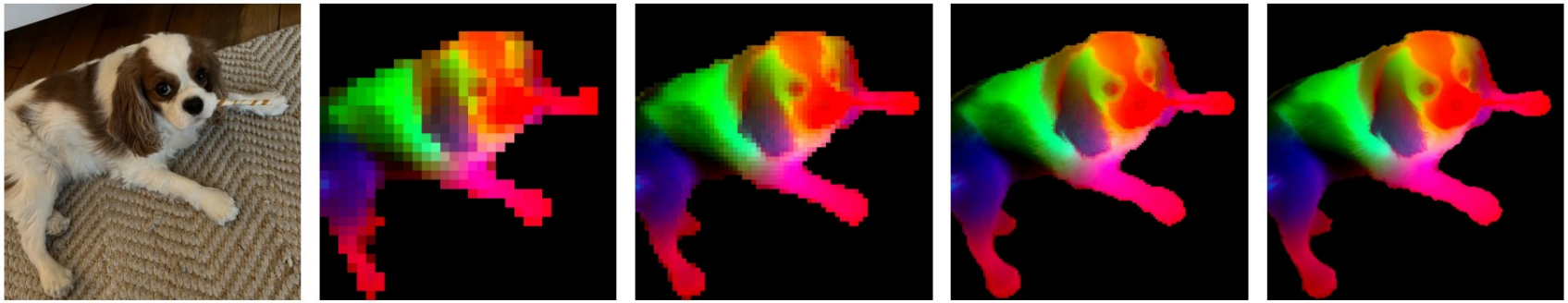


Figure 4: DINOv3 at very high resolution. We visualize dense features of DINOv3 by mapping the first three components of a PCA computed over the feature space to RGB. To focus the PCA on the subject, we mask the feature maps via background subtraction. With increasing resolution, DINOv3 produces crisp features that stay semantically meaningful. We visualize more PCAs in Sec. 6.1.1.

Although increased training iterations lead to better benchmark performance, the feature map representation becomes noisier

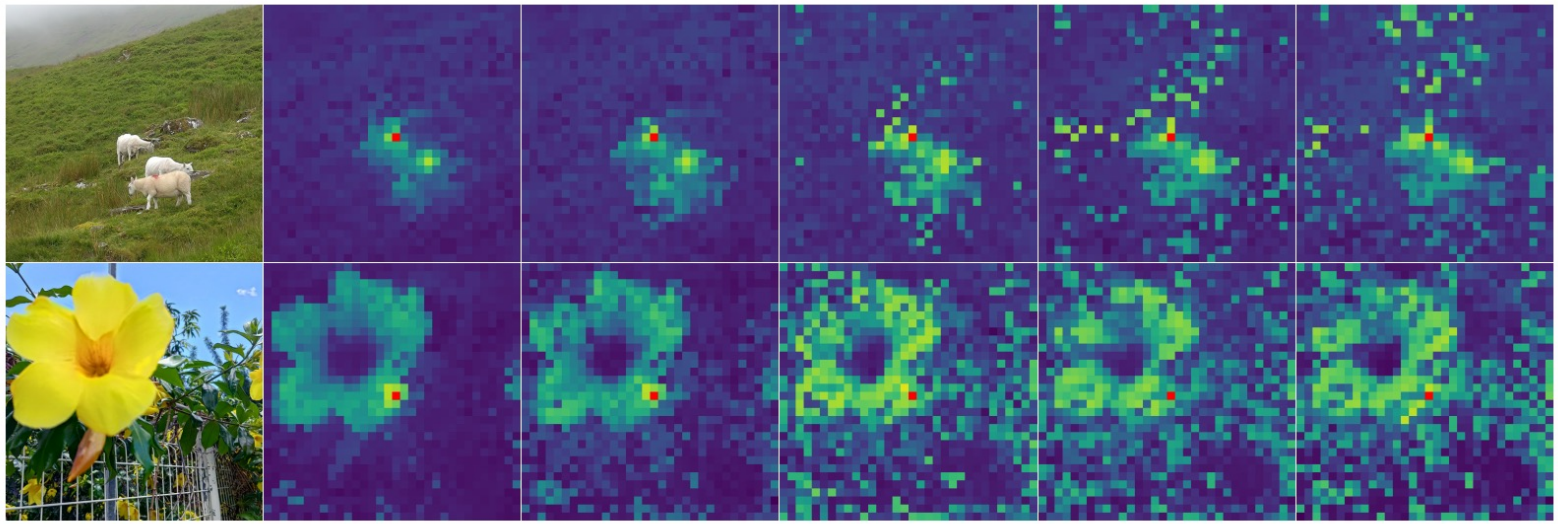


Figure 6: Evolution of the cosine similarity between the patch noted in red and all other patches. As training progresses, the features produced by the model become less localized and the similarity maps become noisier.

New Gram loss: Minimises difference in feature space anchored to earlier iteration of the model

We introduce a new objective which mitigates the degradation of patch-level consistency by enforcing the quality of the patch-level consistency, without impacting the features themselves. This new loss function operates on the Gram matrix: the matrix of all pairwise dot products of patch features in an image. We want to push the Gram matrix of the student towards that of an earlier model, referred to as the *Gram teacher*. We select the Gram teacher by taking an early iteration of the teacher network, which exhibits superior dense properties. By operating on the Gram matrix rather than the feature themselves, the local features are free to move, provided the structure of similarities remains the same. Suppose we have an image composed of P patches, and a network that operates in dimension d. Let us denote by $\mathbf{X}_S$ (respectively $\mathbf{X}_G$) the $P \times d$ matrix of $\mathbf{L}_2$-normalized local features of the student (respectively the Gram teacher). We define the loss $\mathcal{L}_{\text{Gram}}$ as follows:

$$\mathcal{L}_{\text{Gram}} = \left\| \mathbf{X}_S \cdot \mathbf{X}_S^\top - \mathbf{X}_G \cdot \mathbf{X}_G^\top \right\|_{\text{F}}^2 . \tag{2}$$

We only compute this loss on the global crops. Even though it can be applied early on during the training, for efficiency, we start only after 1M iterations. Interestingly, we observe that the late application of $\mathcal{L}_{\text{Gram}}$ still manages to "repair" very degraded local features. In order to further improve performance, we update the Gram teacher every 10k iterations at which the Gram teacher becomes identical to the main EMA teacher.

What is a Gram Matrix?

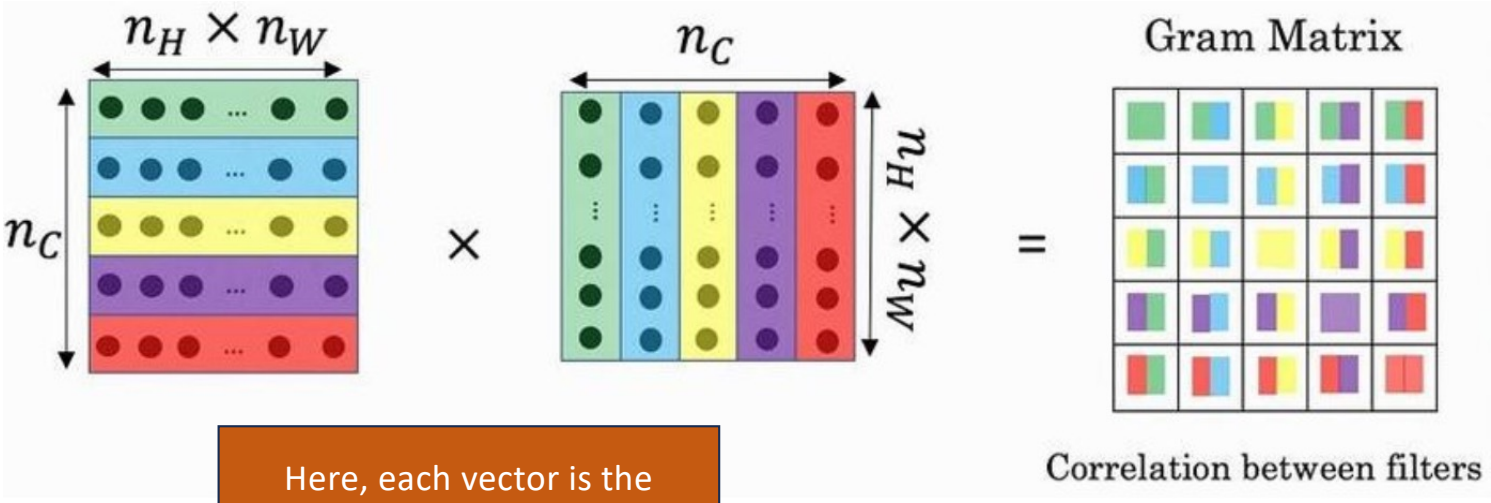
what is the mathematical equation for gram matrix

The mathematical equation for a Gram matrix, as described in the paper (Section 4.2), is derived from the pairwise dot products of patch features. Given a set of P patches and a network that operates in dimension d, let X be the $P \times d$ matrix of L_2-normalized local features. The Gram matrix, G, is then calculated as:

$$G = XX^T$$

This results in a $P \times P$ matrix where each element G_{ij} is the dot product (or cosine similarity, given L_2-normalization) between the feature vectors of patch i and patch j.

Gram Matrix



Here, each vector is the feature representation of each image patch of the ViT

https://www.researchgate.net/figure/The-Gram-Matrix-is-created-from-a-target-image-and-a-reference-image_fig4_356667127

Even though Gram loss is applied after 1M training iterations, it still repairs the damage

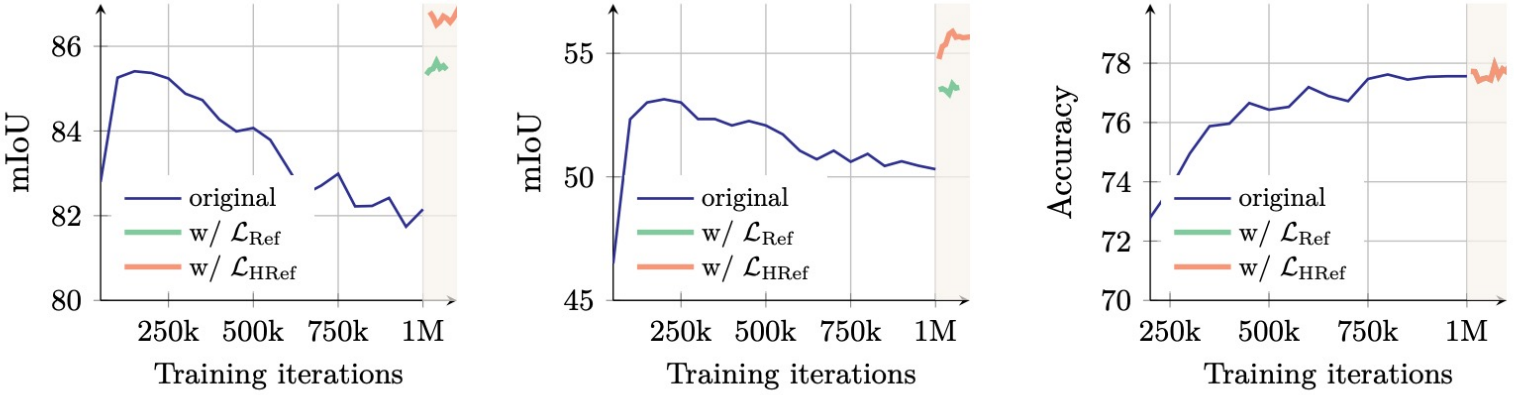
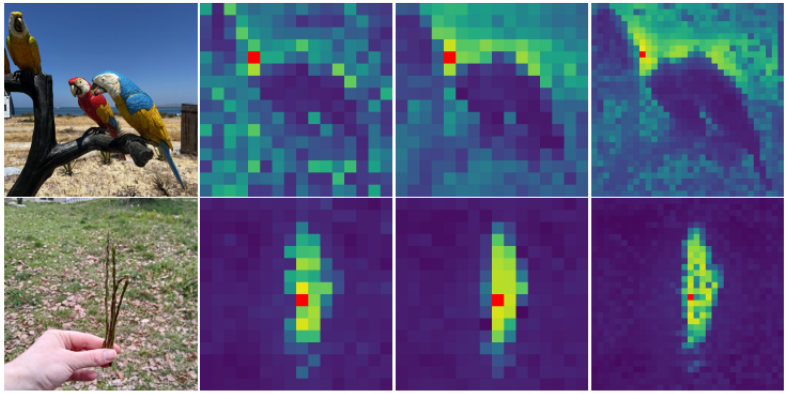


Figure 8: Evolution of the results on different benchmarks after applying our proposed *Gram anchoring* method. We visualize results when continuing the original training with our refinement step, noted '$\mathcal{L}_{\text{Ref}}$'. We also plot results obtained when using higher-resolution features for the Gram objective as introduced in following Sec. 4.3 and noted '$\mathcal{L}_{\text{HRef}}$'. We highlight the iterations which use the Gram objective.

Higher Input Resolution is Better



(a) Gram matrices at different input resolutions.

Method	Teacher Iteration	Res.	IN1k Linear	ADE mIoU	NYU RMSE
Baseline	—	—	**88.2**	50.3	0.307
GRAM	200k	×1	88.0	53.6	0.285
GRAM	200k	×2	88.0	**55.7**	**0.281**
GRAM	100k	×2	87.9	**55.7**	0.284
GRAM	1M	×2	88.1	54.9	0.290

(b) Ablation of Gram teachers and resolutions.

Figure 9: Quantitative and qualitative study of the impact of high-resolution Gram. We show (a) the improved cosine maps after down-sampling the high-resolution maps into smaller ones, and (b) the quantitative improvements brought by varying the training iteration and the resolution of the Gram teacher.

Using Gram Matrix Anchoring helps to Improve Feature Maps

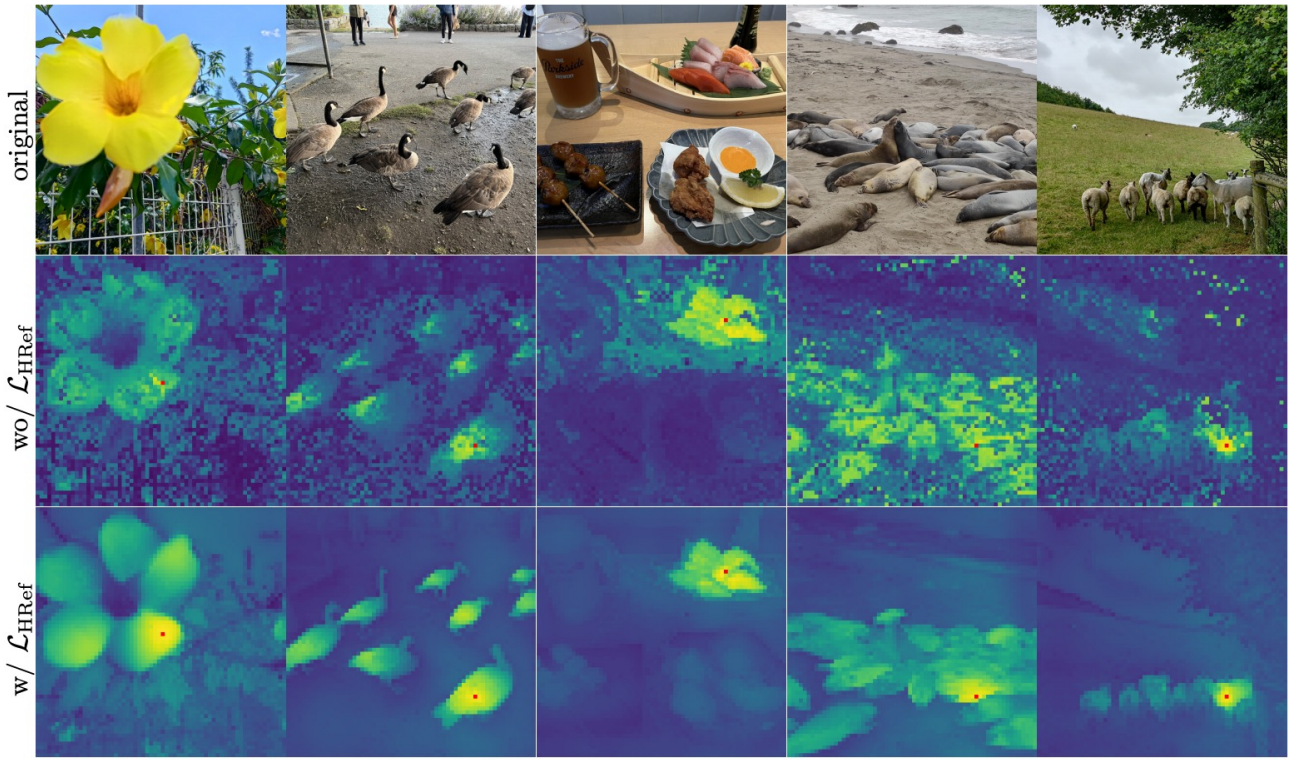


Figure 10: Qualitative effect of Gram anchoring. We visualize cosine maps before and after using the refinement objective $\mathcal{L}_{\text{HRef}}$. The input resolution of the images is 1024×1024 pixels.

DINOv3 has huge improvement in feature map representation

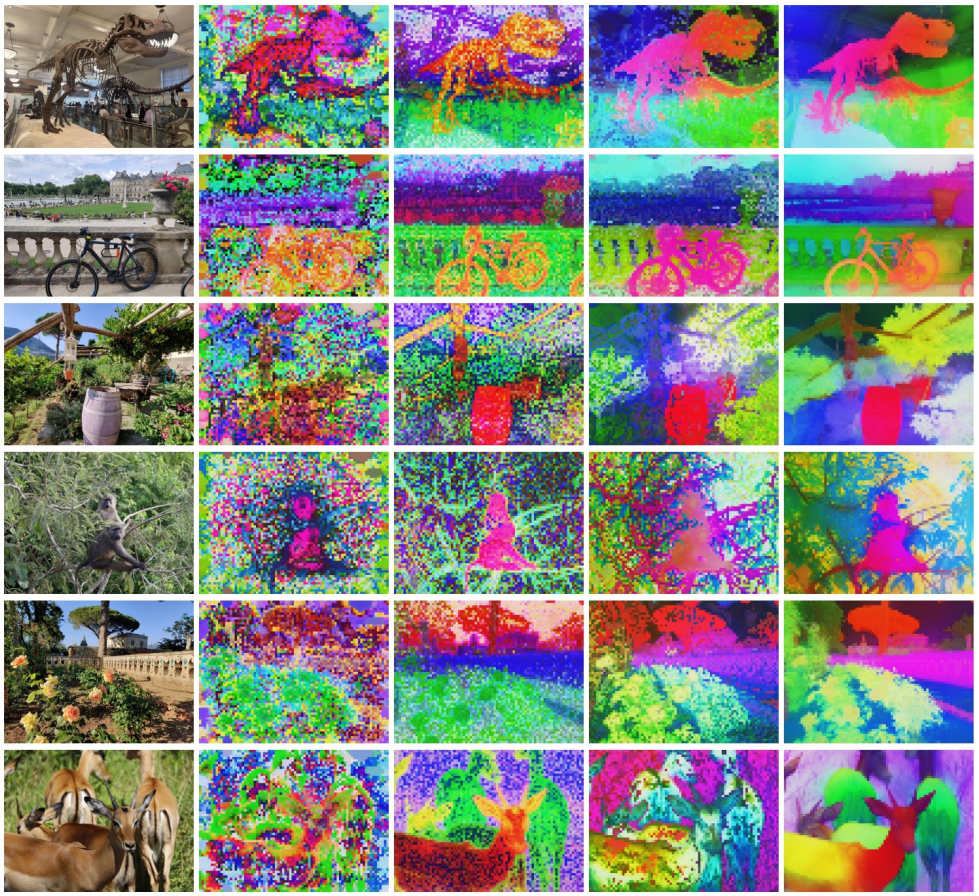


Figure 13: Comparison of dense features. We compare several vision backbones by projecting their dense outputs using PCA and mapping them to RGB. From left to right: SigLIP 2 ViT-g/16, PEspatial ViT-G/14, DINOv2 ViT-g/14 with registers, DINOv3 ViT-7B/16. Images are forwarded at resolution 1280×960 for models using patch 16 and 1120×840 for patch 14, *i.e.* all feature maps have size 80×60.

Results

SoTA for object detection

Table 10: Comparison with state-of-the-art systems on object detection. We train a detection adapter on top of a *frozen* DINOv3 backbone. We show results on the validation set of the COCO and COCO-O datasets, and report the mAP across IoU thresholds, as well as the effective robustness (ER). Our detection system based on DINOv3 sets a new state of the art. As the InternImage-G detection model has not been released, we were unable to reproduce their results or compute COCO-O scores.

			Parameters			COCO		COCO-O	
Model	Detector	FT	Encoder	Decoder	Trainable	Simple	TTA	mAP	ER
EVA-02	Cascade	🔥	300M	—	300M	64.1	—	63.6	34.7
InternImage-G	DINO	🔥	6B	—	6B	65.1	65.3	—	—
EVA-02	Co-DETR	🔥	300M	—	300M	65.4	65.9	63.7	34.3
PEspatial	DETA	🔥	1.9B	50M	2B	65.3	66.0	64.0	34.7
DINOv3	Plain-DETR	❄	7B	100M	100M	**65.6**	**66.1**	**66.4**	**36.8**

SoTA for semantic segmentation

Table 11: Comparison with state-of-the-art systems for semantic segmentation on ADE20k. We evaluate the model in a single- or multi-scale setup (respectively Simple and TTA). Following common practice, we run this evaluation at resolution 896 and report mIoU scores. BEIT3, ONE-PEACE and DINOv3 use a Mask2Former with ViT-Adapter architecture, and the decoder parameters take into account both. We report results on further datasets in Tab. 24

		Parameters			mIoU	
Model	FT	Encoder	Decoder	Trainable	Simple	TTA
BEIT3	🔥	1.0B	550M	1.6B	62.0	62.8
InternImage-H	🔥	1.1B	230M	1.3B	62.5	62.9
ONE-PEACE	🔥	1.5B	710M	2.2B	62.0	**63.0**
DINOv3	❄	7B	927M	927M	**62.6**	**63.0**

SoTA for monocular depth estimation

Table 12: Comparison with state-of-the-art systems for relative monocular depth estimation. By combining DINOv3 with Depth Anything V2 (Yang et al., 2024b), we obtain a SotA model for relative depth estimation.

		NYUv2		KITTI		ETH3D		ScanNet		DIODE	
Method	FT	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑	ARel ↓	δ_1 ↑
MiDaS	🔥	11.1	88.5	23.6	63.0	18.4	75.2	12.1	84.6	33.2	71.5
LeReS	🔥	9.0	91.6	14.9	78.4	17.1	77.7	9.1	91.7	27.1	76.6
Omnidata	🔥	7.4	94.5	14.9	83.5	16.6	77.8	7.5	93.6	33.9	74.2
DPT	🔥	9.8	90.3	10.0	90.1	7.8	94.6	8.2	93.4	**18.2**	75.8
Marigold	🔥	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	30.8	77.3
DAv2 (ViT-g)	🔥	4.4	97.9	7.5	94.7	13.1	86.5	—	—	—	—
DINOv3	❄️	**4.3**	**98.0**	**7.3**	**96.7**	**5.4**	**97.5**	**4.4**	**98.1**	25.6	**82.2**

SoTA for 3D understanding

Table 13: 3D understanding using Visual Geometry Grounded Transformer (VGGT) (Wang et al., 2025). Simply by swapping DINOv2 for DINOv3 ViT-L as the image feature extractor in the VGGT pipeline, we are able to obtain state-of-the-art results on various 3D geometry tasks. We reproduce baseline results from Wang et al. (2025). We also report methods using ground truth camera information, marked with *. Camera pose estimation results are reported with AUC@30.

(a) Camera pose estimation.

Method	Re10K	CO3Dv2
DUSt3R	67.7	76.7
MASt3R	76.4	81.8
VG GSfM v2	78.9	83.4
CUT3R	75.3	82.8
FLARE	78.8	83.3
VGGT	85.3	88.2
DINOv3	**86.3**	**89.6**

(b) Multi-view estimation on DTU.

Method	Acc.↓	Comp.↓	Overall↓
Gipuma*	0.283	0.873	0.578
CIDER*	0.417	0.437	0.427
MASt3R*	0.403	0.344	0.374
GeoMVSNet*	0.331	0.259	0.295
DUSt3R	2.677	0.805	1.741
VGGT	0.389	0.374	0.382
DINOv3	**0.375**	**0.361**	**0.368**

(c) View matching on ScanNet-1500.

Method	AUC@5	AUC@10
SuperGlue	16.2	33.8
LoFTR	22.1	40.8
DKM	29.4	50.7
CasMTR	27.1	47.0
Roma	31.8	53.4
VGGT	33.9	55.2
DINOv3	**35.2**	**56.1**

Strong results in video segmentation tasks

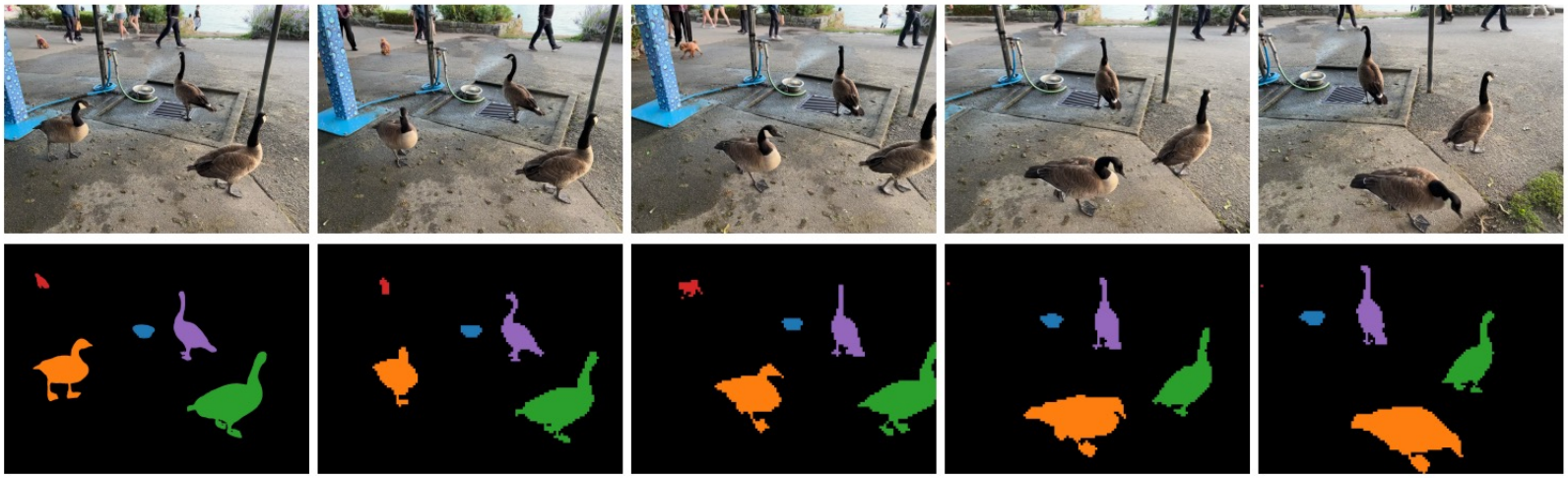


Figure 15: Segmentation tracking example. Given the ground-truth instance segmentation masks for the initial frame, we propagate the instance labels to subsequent frames according to patch similarity in the feature space of DINOv3. The input resolution is 2048×1536 pixels, resulting in 128×96 patches.

Comparable performance to supervised baseline on ImageNet classification

Table 7: Classification accuracy of linear probes trained on ImageNet1k with frozen backbones. Weakly- and self-supervised models are evaluated with image resolution adapted to 1024 patch tokens (*i.e.* 448×448 for patch size 14, 512×512 for patch size 16). For reference, we also list results from Dehghani et al. (2023) using a different evaluation protocol (marked with *).

		ImageNet			Rendition		Hard		
Method	ViT	Val	V2	ReaL	R	S	A	C ↓	Obj.
Supervised backbones									
Zhai et al. (2022a)*	G/14	89.0	81.3	90.6	91.7	—	78.8	—	69.6
Chen et al. (2023)*	e/14	89.3	82.5	90.7	94.3	—	81.6	—	71.5
Dehghani et al. (2023)*	22B/14	89.5	83.2	90.9	94.3	—	83.8	—	74.3
Agglomerative backbones									
AM-RADIOv2.5	g/14	88.0	80.2	90.3	83.8	67.1	81.3	27.1	68.4
Weakly-supervised backbones									
PEcore	G/14	**89.3**	**81.6**	90.4	**92.2**	**71.9**	**89.0**	22.7	**80.2**
SigLIP 2	g/16	89.1	81.6	**90.5**	**92.2**	71.8	84.6	30.0	78.6
AIMv2	3B/14	87.9	79.5	89.7	82.3	67.1	74.5	29.5	69.0
EVA-CLIP	18B/14	87.9	79.3	89.5	85.2	64.0	81.6	33.0	71.9
Self-supervised backbones									
Web-DINO	7B/14	85.9	77.1	88.6	75.6	64.0	71.6	31.2	69.7
Franca	g/14	84.8	75.3	89.2	67.6	49.5	56.5	40.0	54.5
DINOv2	g/14	87.3	79.5	89.9	81.1	65.4	81.7	24.1	66.4
DINOv3	7B/16	88.4	81.4	90.4	91.1	71.3	86.9	**19.6**	79.0

Question to Ponder

- What could be the cause of feature map degradation as training iterations progress?
- Can synthetic data help to increase data for SSL?
- Can synthetic high-resolution data help improve feature map resolution?

Other Information

Image-level SSL objective

Image-level objective (Caron et al., 2021). We consider the cross-entropy loss between the features extracted from a student and a teacher network. Both features are coming from the class token of a ViT, obtained from different crops of the same image. We pass the student class token through the student DINO head. This head is an MLP model outputting a vector of scores, that we call "prototype scores". We then apply a softmax to obtain p_s. Similarly, we apply the teacher DINO head to the teacher class token to obtain teacher prototype scores. We then apply a softmax followed by a centering with moving average (or a Sinkhorn-Knopp centering as detailed thereafter) to obtain p_t. The DINO loss term corresponds to:

$$\mathcal{L}_{DINO} = -\sum p_t \log p_s$$

We learn the parameters of the student and build the teacher head with an exponential moving average of past iterates (He et al., 2020).

Patch-level SSL objective

Patch-level objective (Zhou et al., 2022a). We randomly mask some of the input patches given to the student, but not to the teacher. We then apply the student iBOT head to the student mask tokens. Similarly, we apply the teacher iBOT head to the (visible) teacher patch tokens corresponding to the ones masked in the student. We then apply the softmax and centering steps as above, and obtain the iBOT loss term:

$$\mathcal{L}_{iBOT} = -\sum_{i} p_{ti} \log p_{si}$$

, where i are patch indices for masked tokens. Similarly to above, we learn the parameters of the student, and build the teacher head through exponential moving average.

KoLeo regularizer

KoLeo regularizer (Sablayrolles et al., 2019). The KoLeo regularizer derives from the Kozachenko-Leonenko differential entropy estimator (see Beirlant et al. (1997); Delattre & Fournier (2017)) and encourages a uniform span of the features within a batch. Given a set of n vectors $(x_1, \ldots, x_n)$, it is defined as

$$\mathcal{L}_{\text{koleo}} = -\frac{1}{n}\sum_{i=1}^{n} \log(d_{n,i}),$$

where $d_{n,i} = \min_{j \neq i} \|x_i - x_j\|$ is the minimum distance between x_i and any other point within the batch. We also ℓ_2-normalize the features before computing this regularizer.

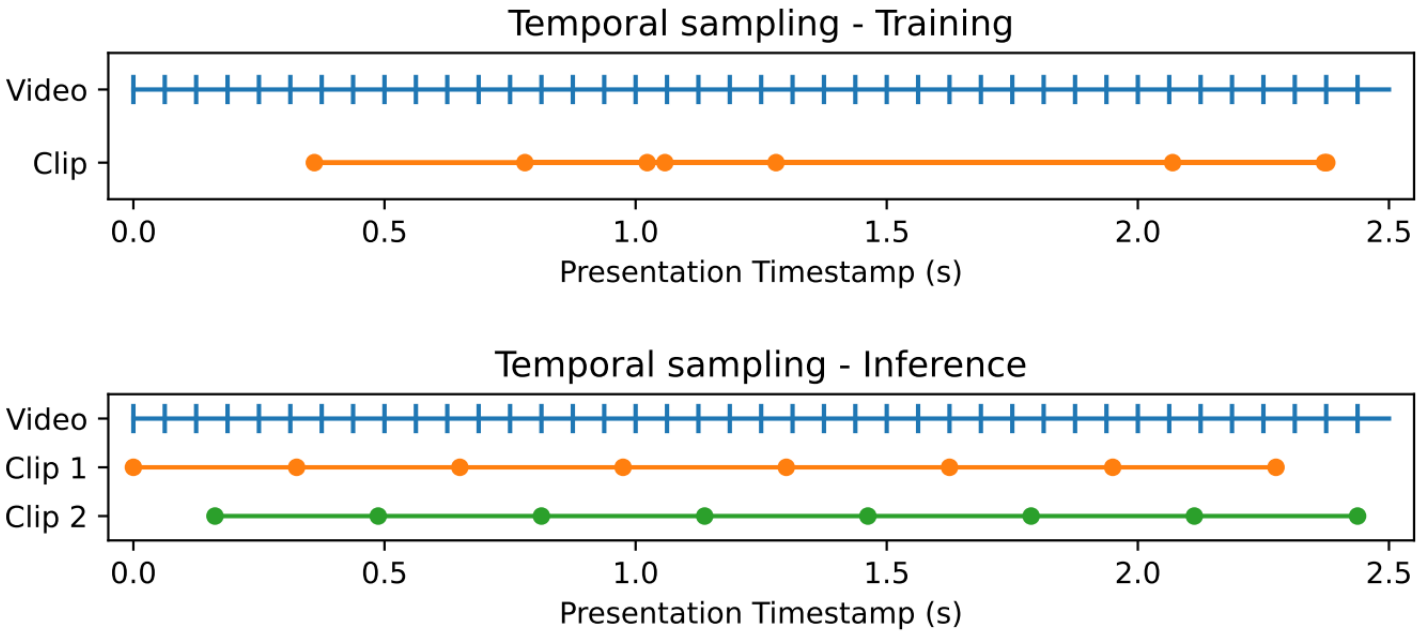
Initial Training Loss Function

$$\mathcal{L}_{\text{Pre}} = \mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + 0.1 * \mathcal{L}_{\text{DKoleo}}.$$

Refinement Step Loss Function (with Gram anchoring loss)

$$\mathcal{L}_{\text{Ref}} = w_{\text{D}}\mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + w_{\text{DK}}\mathcal{L}_{\text{DKoleo}} + w_{\text{Gram}}\mathcal{L}_{\text{Gram}}.$$

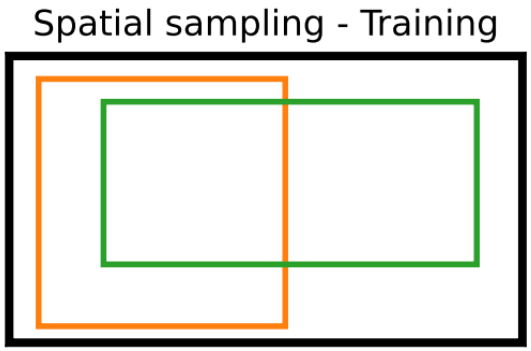
Temporal Sampling in Video Tasks

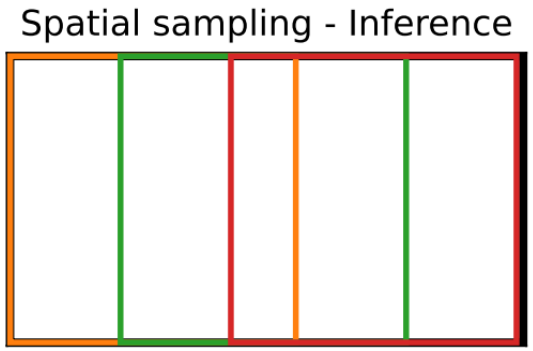


Training Time (Random): Sample clips at random by choosing random frames from the whole video

Test Time (Deterministic): Take two overlapping sets of frames such that they cover as much of the video as possible and their timestamps interleave

Spatial Sampling in Video Tasks





Training Time (Random): Samplerandom spatial crops that covers ≥ 40% of the area

Test Time (Deterministic): Take the three largest square crops aligned to the left, middle and right