

# DINOv2: Learning Robust Visual Features without Supervision

Maxime Oquab<sup>\*\*</sup>, Timothée Darcet<sup>\*\*</sup>, Théo Moutakanni<sup>\*\*</sup>,  
 Huy V. Vo<sup>\*</sup>, Marc Szafraniec<sup>\*</sup>, Vasil Khalidov<sup>\*</sup>, Pierre Fernandez, Daniel Haziza,  
 Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba,  
 Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat,  
 Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal<sup>1</sup>,  
 Patrick Labatut<sup>\*</sup>, Armand Joulin<sup>\*</sup>, Piotr Bojanowski<sup>\*</sup>

*Meta AI Research*      <sup>1</sup>*Inria*

<sup>\*</sup>core team      <sup>\*\*</sup>equal contribution

Reviewed on OpenReview: <https://openreview.net/forum?id=a68SUt6zFt>

## Abstract

The recent breakthroughs in natural language processing for model pretraining on large quantities of data have opened the way for similar foundation models in computer vision. These models could greatly simplify the use of images in any system by producing general-purpose visual features, i.e., features that work across image distributions and tasks without finetuning. This work shows that existing pretraining methods, especially self-supervised methods, can produce such features if trained on enough curated data from diverse sources. We revisit existing approaches and combine different techniques to scale our pretraining in terms of data and model size. Most of the technical contributions aim at accelerating and stabilizing the training at scale. In terms of data, we propose an automatic pipeline to build a dedicated, diverse, and curated image dataset instead of uncurated data, as typically done in the self-supervised literature. In terms of models, we train a ViT model (Dosovitskiy et al., 2021) with 1B parameters and distill it into a series of smaller models that surpass the best available general-purpose features, OpenCLIP (Ilharco et al., 2021) on most of the benchmarks at image and pixel levels.

## 1 Introduction

Learning task-agnostic pretrained representations have become the standard in Natural Language Processing (NLP) (Radford et al., 2019; Raffel et al., 2020; Chowdhery et al., 2022; Hoffmann et al., 2022; Touvron et al., 2023). One can use these features “as they are”, i.e., without fine-tuning, and achieve performances on downstream tasks that are significantly better than those produced by task-specific models (Brown et al., 2020). This success has been fueled by pretraining on large quantities of raw text using pretext objectives, such as language modeling (Radford et al., 2017) or word vectors (Devlin et al., 2019), that require no supervision.

Following this paradigm shift in NLP, we expect similar “foundation” models to appear in computer vision (Bommasani et al., 2021). These models should generate visual features that work out of the box on any task, both at the image level, e.g., image classification, and pixel level, e.g., segmentation. Most promising efforts towards these foundation models focus on text-guided pretraining, i.e., using a form of textual supervision to guide the training of the features (Joulin et al., 2016; Mahajan et al., 2018; Radford et al.,

All the authors are affiliated to Meta, except Julien Mairal who is affiliated to Inria. Timothée Darcet and Pierre Fernandez have a co-affiliation with Inria. Théo Moutakanni has a co-affiliation with Université Paris Saclay. Alaaeldin El-Nouby has a co-affiliation with Inria and ENS-PSL. Correspondence: {qas, timdarcet, theomoutakanni, ajoulin, bojanowski}@meta.com

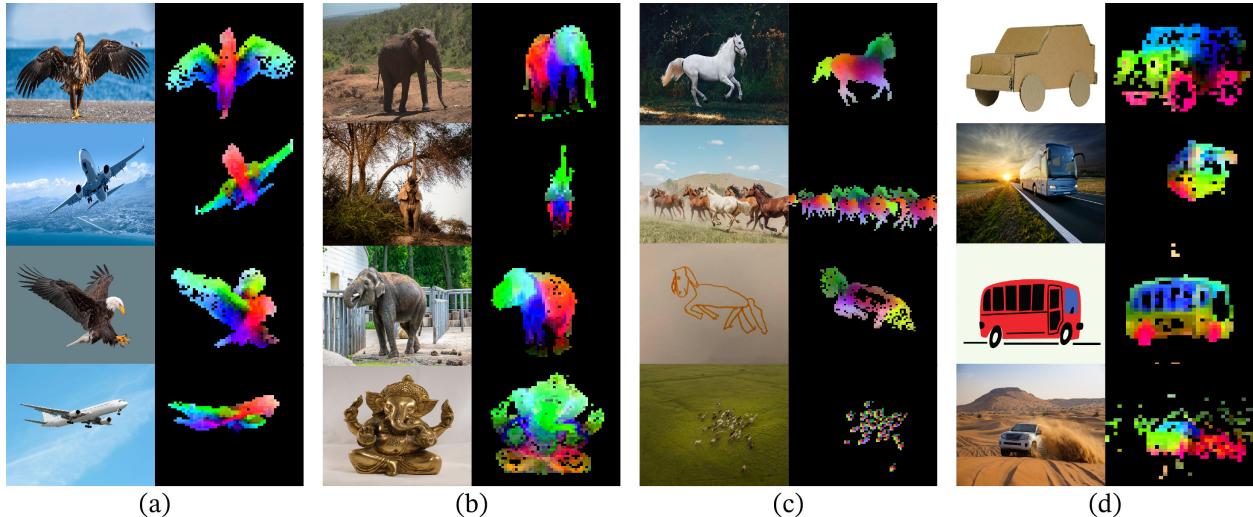


Figure 1: **Visualization of the first PCA components.** We compute a PCA between the patches of the images from the same column (a, b, c and d) and show their first 3 components. Each component is matched to a different color channel. Same parts are matched between related images despite changes of pose, style or even objects. Background is removed by thresholding the first PCA component.

2021). This form of text-guided pretraining limits the information that can be retained about the image since captions only approximate the rich information in images, and complex pixel-level information may not surface with this supervision. Furthermore, these image encoders require aligned text-image corpora and hence, do not offer the flexibility of their text counterparts, that is, to learn from raw data alone.

An alternative to text-guided pretraining is self-supervised learning (Caron et al., 2018; Chen et al., 2020; He et al., 2022) where features are learned from images alone. These approaches are conceptually closer to pretext tasks such as language modeling and can capture information at the image and pixel level (Caron et al., 2021). Additionally, the features output by self-supervised models have been shown to exhibit various useful properties, and have enabled a wide variety of applications (Amir et al., 2022; Tumanyan et al., 2022; Ofri-Amar et al., 2023; Hamilton et al., 2022). However, despite their potential to learn general-purpose features, most of the advances in self-supervised learning were made in the context of pretraining on a small curated dataset, ImageNet-1k (Russakovsky et al., 2015). Some efforts on scaling these approaches beyond ImageNet-1k have been attempted (Caron et al., 2019; Goyal et al., 2021; 2022a), but they focused on uncurated datasets, which typically lead to a significant drop in the quality of the features. This is explained by the lack of control over the data quality and diversity, which are essential to produce good features.

In this work, we explore if self-supervised learning has the potential to learn general-purpose visual features if pretrained on a large quantity of curated data. We revisit existing discriminative self-supervised approaches that learn features at both the image and patch level, such as iBOT (Zhou et al., 2022a), and we reconsider some of their design choices under the lens of a larger dataset. Most of our technical contributions are tailored toward stabilizing and accelerating discriminative self-supervised learning when scaling in model and data sizes. These improvements make our approach around  $2\times$  faster and require  $3\times$  less memory than similar discriminative self-supervised methods, allowing us to leverage longer training with larger batch sizes.

Regarding pretraining data, we have built an automatic pipeline to filter and rebalance datasets from an extensive collection of uncurated images. This pipeline is inspired by pipelines used in NLP (Wenzek et al., 2020), where data similarities are used instead of external metadata and do not require manual annotation. A major difficulty when dealing with images in the wild is to rebalance concepts and avoid overfitting on a few dominant modes. In this work, a naive clustering approach works reasonably well to resolve this issue. We gathered a small but diverse corpus of 142M images to validate our approach.

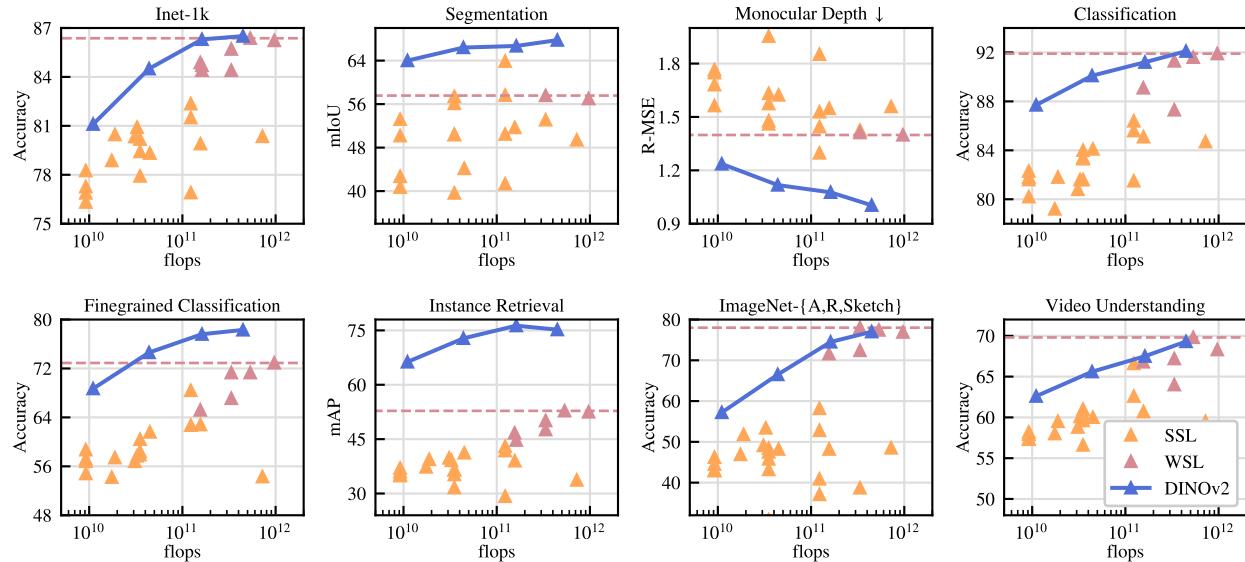


Figure 2: **Evolution of performance when scaling in parameters.** We show performance on eight types of vision tasks, as presented in Sec. 7, and average metrics with each type. Features are extracted from our self-supervised encoders, DINOV2 (dark blue), and we compare them with self-supervised methods (pale orange), as well as weakly-supervised methods (dark pink). We report the best-performing weakly-supervised model’s performance as a dashed horizontal line. Our family of models drastically improves over the previous state of the art in self-supervised learning and reaches performance comparable with weakly-supervised features. See Sec. 7 for a detailed analysis.

Finally, we provide a variety of pretrained visual models, called DINOV2, trained with different Vision Transformers (ViT) (Dosovitskiy et al., 2016) architectures on our data. We release all the models and the code to retrain DINOV2 on any data. We validate the quality of DINOV2 on various computer vision benchmarks at both image and pixel levels as we scale them, as summarized in Fig. 2. We conclude that self-supervised pretraining alone is a good candidate for learning transferable frozen features that are competitive with the best openly available weakly-supervised models.

## 2 Related Work

**Intra-image self-supervised training.** A first family of self-supervised methods focuses on pretext tasks built from the image, i.e., extracting a signal from the image to be predicted from the rest of the image. This idea has become prevalent with the work of Doersch et al. (2015), where they train by predicting the context of a given patch. Many other pretext tasks were introduced based on, for example, re-colorizing images (Zhang et al., 2016), predicting transformations (Gidaris et al., 2018), inpainting (Pathak et al., 2016) or patch re-ordering (Noroozi & Favaro, 2016; Misra & Maaten, 2020). Recently, the emergence of patch-based architectures, like ViTs, has led to a revisit of inpainting for pre-training (He et al., 2022; Bao et al., 2021; El-Nouby et al., 2021), potentially in feature space (Assran et al., 2023; Baevski et al., 2022). Of particular interest, He et al. (2022) show that a masked auto-encoder (MAE) learns features that provide substantial improvements when finetuned on downstream tasks. This property of MAEs has been further validated on video (Tong et al., 2022), audio (Xu et al., 2022), and across other modalities (Girdhar et al., 2023). However, their features require supervised finetuning, while our features perform well out of the box.

**Discriminative self-supervised learning.** The second line of work, closer to ours, is using discriminative signals between images or groups of images to learn features. This family of methods has roots in early deep learning work (Hadsell et al., 2006) but became popular with the emergence of instance classification methods (Dosovitskiy et al., 2016; Bojanowski & Joulin, 2017; Wu et al., 2018). Several improvements

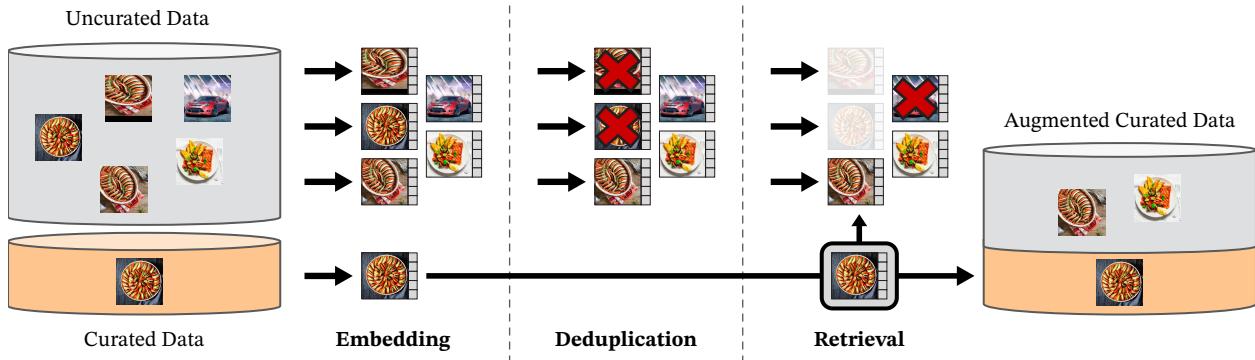


Figure 3: **Overview of our data processing pipeline.** Images from curated and uncurated data sources are first mapped to embeddings. Uncurated images are then deduplicated before being matched to curated images. The resulting combination augments the initial dataset through a self-supervised retrieval system.

were made based either on instance-level objectives (Hénaff et al., 2019; He et al., 2020; Chen & He, 2021; Chen et al., 2020; Grill et al., 2020; Caron et al., 2021) or clustering (Caron et al., 2018; Asano et al., 2020; Caron et al., 2020). These methods provide performant frozen features on standard benchmarks like ImageNet (Russakovsky et al., 2015), but they are hard to scale to larger model sizes (Chen et al., 2021). In this work, we revisit the training of these approaches in the context of large pretraining datasets and models. In particular, we build on top of Zhou et al. (2022a) that we find particularly suited for scaling.

**Scaling self-supervised pretraining.** A growing body of work has focused on the scaling abilities of self-supervised learning in terms of data and model size (Caron et al., 2019; Goyal et al., 2019; Tian et al., 2021; Goyal et al., 2022a). Most of these works use large quantities of uncurated data to train models without supervision. They show evidence that discriminative methods scale with data, but because of the poor quality of the pretraining data, most of the results are obtained by finetuning the features. Of particular interest, Goyal et al. (2021) have also shown that these methods benefit from scaling in model size given enough pretrained data. This line of work questions the ability of self-supervised methods to work on any data while we focus on producing the best pretrained encoders.

**Automatic data curation.** Our dataset construction borrows from the image retrieval community (Weinzaepfel et al., 2021; Radenović et al., 2018b; Berman et al., 2019; Douze et al., 2009; Tolias et al., 2016; Revaud et al., 2019). In particular, the use of retrieval to augment the training set has been studied in the context of semi-supervised learning (Yalniz et al., 2019). Similarly, others have used hashtags or other metadata (Mahajan et al., 2018; Radford et al., 2021) or pretrained vision encoders (Schuhmann et al., 2021; 2022) to filter uncurated datasets. Unlike these works, we use no pretrained encoders, metadata nor supervision to filter images and leverage visual similarity between images. Our approach is inspired by text curation pipelines (Wenzek et al., 2020), where a language model is trained on Wikipedia to score texts extracted from an uncurated source.

### 3 Data Processing

We assemble our curated LVD-142M dataset by retrieving, from a large pool of uncurated data, images that are close to those in several curated datasets. We describe below the main components in our data pipeline including the curated/uncurated data sources, the image deduplication step and the retrieval system. Our pipeline does not require any metadata or text and directly works with images, as shown in Fig. 3. We refer the reader to appendix A for more details on our approach.

**Data sources.** Our selection of curated datasets is detailed in the appendix (Table 15) and contains ImageNet-22k, the train split of ImageNet-1k, Google Landmarks and several fine-grained datasets. For the

uncurated data source, we collect a raw unfiltered dataset of images from a publicly available repository of crawled web data. From each web page in the repository, we extract URL links of images from `<img>` tags. We discard URLs that are unsafe or restricted by domains, and post-process the downloaded images (PCA hash deduplication, NSFW filtering, and blurring identifiable faces). This results in 1.2B unique images.

**Deduplication.** We apply the copy detection pipeline of Pizzi et al. (2022) to the uncurated data and remove near-duplicate images. This reduces redundancy and increases diversity among images. We also remove near-duplicates of images contained in the test or validation set of any benchmark used in this work.

**Self-supervised image retrieval.** We build our curated pretraining dataset by retrieving images from our uncurated data source that are close to images in our curated sources. In order to do this, we first compute an image embedding using a self-supervised ViT-H/16 network pretrained on ImageNet-22k, and use cosine-similarity as a distance measure between images. Then, we perform k-means clustering of the uncurated data. Given a query dataset for retrieval, if it is large enough we retrieve  $N$  (typically 4) nearest neighbors for each query image. If it is small, we sample  $M$  images from the cluster corresponding to each query image. Although visual inspection seemed to indicate good retrieval quality for  $N$  much larger than 4, this leads to more collisions (images that are nearest-neighbor retrievals of multiple queries). We choose  $N = 4$  as it provides a good tradeoff in that sense.

**Implementation Details.** The deduplication and retrieval stages of our pipeline rely on the Faiss library (Johnson et al., 2019) to efficiently index and compute batch searches of nearest embeddings. In particular, we heavily leverage its support for GPU-accelerated indices, using inverted file indices with product quantization codes (Jegou et al., 2010). The whole processing is distributed on a compute cluster of 20 nodes equipped with 8 V100-32GB GPUs and takes less than two days to produce the LVD-142M dataset.

## 4 Discriminative Self-supervised Pre-training

We learn our features with a discriminative self-supervised method that can be seen as a combination of DINO and iBOT losses with the centering of SwAV (Caron et al., 2020). We also add a regularizer to spread features and a short high-resolution training phase. We rapidly introduce each of these approaches, but more details can be found in the related papers, or in our open-sourced code.

- **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 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.

- **Untying head weights between both objectives.** Both the DINO and the iBOT loss use a learnable MLP projection head. It is applied to the output tokens and the loss is compute atop. In Zhou et al. (2022a), an ablation study shows that sharing parameters between the DINO and iBOT heads leads to better performance. At scale, we observed that the opposite is true, and we therefore use two separate heads in all our experiments.
- **Sinkhorn-Knopp centering (Caron et al., 2020).** Ruan et al. (2023) recommend to replace the teacher softmax-centering step of DINO and iBot by the Sinkhorn-Knopp (SK) batch normalization of SwAV (Caron et al., 2020). We run the Sinkhorn-Knopp algorithm steps for 3 iterations. For the student, we apply the softmax normalization.
- **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  $(x_1, \dots, x_n)$ , 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  $\ell_2$ -normalize the features before computing this regularizer.

- **Adapting the resolution (Touvron et al., 2019).** Increasing image resolution is key to pixel-level downstream tasks such as segmentation or detection, where small objects disappear at low resolutions. However, training at high resolution is time and memory demanding, and instead, we increase the resolution of images to  $518 \times 518$  during a short period at the end of pretraining. This is also similar to UniViT training from Likhomanenko et al. (2021) and FlexiViT training from Beyer et al. (2023).

## 5 Efficient implementation

We consider several improvements to train models at a larger scale. We train models on A100 GPUs using PyTorch 2.0. The code and pretrained models are made available under Apache 2.0 license <sup>1</sup>. The details of our models are in the appendix, Table 17. With the same hardware, compared to the iBOT implementation, the DINOv2 code runs around 2 $\times$  faster using only 1/3 of the memory.

**Fast and memory-efficient attention.** We implemented our own version of FlashAttention (Dao et al., 2022) to improve memory usage and speed on the self-attention layers. Our version is on par with or better than the original on all cases considered, while covering more use-cases and hardware. Due to the GPU hardware specifics, the efficiency is best when the embedding dimension per head is a multiple of 64, and the matrix operations are even better when the full embedding dimension is a multiple of 256. As a consequence, our ViT-g architecture slightly differs from the architecture proposed by Zhai et al. (2022) in order to maximize compute efficiency, and we use an embedding dimension of 1536 with 24 heads (64 dim/head), rather than 1408 with 16 heads (88 dim/head). Our experiments did not show significant differences in final accuracy, and our ViT-g backbone counts 1.1B parameters.

**Sequence packing.** The DINO algorithm requires forwarding both large crops (at resolution 224) and small crops (resolution 98). When split into patches, these two groups are represented by token sequences of different lengths and cannot be forwarded together. In order to accelerate training, we use a trick called "sequence packing," which originates from NLP (Krell et al., 2022). The idea is simple: we concatenate the

<sup>1</sup><https://github.com/facebookresearch/dinov2>

sequences we must forward through the transformers into a single long sequence. We pass this sequence through the transformer blocks as usual. However, a block-diagonal mask is applied to the self-attention matrix in attention layers, preventing attention between different sequences. This way, the forward is strictly equivalent to forwarding each sequence separately. This trick gives us significant compute efficiency gains compared to using separate forward and backward passes, as in prior implementations. The lower-level components of our setup are available in the xFormers library<sup>2</sup> (Lefauze et al. (2022)).

**Efficient stochastic depth.** We implement an improved version of stochastic depth (Huang et al., 2016) that skips the computation of the dropped residuals rather than masking the result. This saves memory and compute in proportion approximately equal to the drop rate, thanks to specific fused kernels. With high drop rates ( $d = 40\%$  in this work), this allows a drastic improvement in compute efficiency and memory usage. The implementation consists of randomly shuffling the  $B$  samples over the batch dimension, and slicing the first  $(1 - d) \times B$  samples for the computations in the block.

**Fully-Sharded Data Parallel (FSDP).** Minimizing our objective with the AdamW optimizer requires 4 model replicas in float32 precision – student, teacher, optimizer first moments, optimizer second moments. This sums to 16 GB of memory for a billion-parameter model such as our ViT-g. In order to reduce this memory footprint per GPU, we split the model replicas across GPUs, i.e., sharding 16 GB across GPUs using the PyTorch implementation of FSDP. Consequently, the model size is not bounded by the memory of a single GPU but by the total sum of GPU memory across compute nodes. The Pytorch implementation of FSDP brings a second advantage, which is to save on the cross-GPU communication costs: the weight shards are stored in float32 precision as required by the optimizer, but broadcasting weights and reducing gradients is done in float16 precision for the backbone (MLP heads gradients are reduced in float32 to avoid training instabilities). This leads to approximately 50% reduction in communication costs compared to the float32 gradient all-reduce operation used in DistributedDataParallel (DDP), which is used in other self-supervised pretraining methods (Caron et al., 2021; Zhou et al., 2022a). As a consequence, the training procedure scales more efficiently than DDP with float16 autocast when scaling the number of GPU nodes. Overall, Pytorch-FSDP mixed-precision is superior to DDP with autocast in virtually all cases we encountered.

**Model distillation.** Most of our technical improvements to the training loop aim at improving the training of large models over large quantities of data. For smaller models, we distill them from our largest model, the ViT-g, instead of training them from scratch. Knowledge distillation (Hinton et al., 2014) aims at reproducing the output of a large model with a smaller model by minimizing some distance between both outputs for a set of given inputs. Since our objective function is a form of distillation from the teacher network to the student network, we leverage the same training loop with a few exceptions: we use a larger model as a frozen teacher, keep a spare EMA of the student that we use as our final model, remove the masking and stochastic depth, and, apply the iBOT loss on the two global crops. In our ablations, we observe that this approach achieves better performance than training from scratch, even for a ViT-L. Our distillation method ends up close to the one described by Duval et al. (2023), except we do not modify the loss terms for distillation and evaluate the EMA of the student.

## 6 Ablation Studies

We present a set of ablations to empirically validate different components of our pipeline: the technical modifications described in Sec. 4, the pretraining data and the impact of model distillation. We consider various downstream tasks that are described in Sec. 7.

### 6.1 Improved Training Recipe

Our approach improves over the iBOT method by combining it with several existing components described in Sec. 4. To evaluate their importance, we train multiple models where we successively add components to a baseline iBOT model. We report the Top-1 accuracy on the validation set of ImageNet-1k with a k-NN

<sup>2</sup><https://github.com/facebookresearch/xformers>

	INet-1k k-NN	INet-1k linear
iBOT	72.9	82.3
+ (our reproduction)	74.5 $\uparrow$ 1.6	83.2 $\uparrow$ 0.9
+ LayerScale, Stochastic Depth	75.4 $\uparrow$ 0.9	82.0 $\downarrow$ 1.2
+ 128k prototypes	76.6 $\uparrow$ 1.2	81.9 $\downarrow$ 0.1
+ KoLeo	78.9 $\uparrow$ 2.3	82.5 $\uparrow$ 0.6
+ SwiGLU FFN	78.7 $\downarrow$ 0.2	83.1 $\uparrow$ 0.6
+ Patch size 14	78.9 $\uparrow$ 0.2	83.5 $\uparrow$ 0.4
+ Teacher momentum 0.994	79.4 $\uparrow$ 0.5	83.6 $\uparrow$ 0.1
+ Tweak warmup schedules	80.5 $\uparrow$ 1.1	83.8 $\uparrow$ 0.2
+ Batch size 3k	81.7 $\uparrow$ 1.2	84.7 $\uparrow$ 0.9
+ Sinkhorn-Knopp	81.7 =	84.7 =
+ Untying heads = DINoV2	82.0 $\uparrow$ 0.3	84.5 $\downarrow$ 0.2

Table 1: **Ablation study of the training differences between iBOT and DINoV2.** We optimize for k-NN performance, as in our experience, the linear probe performance is lower-bounded by the k-NN performance. Some modifications, like LayerScale and a high Stochastic Depth (rate=0.4), incur a decrease in linear probe performance, but have the benefits of increasing the stability of training by avoiding NaN loss values during training (Touvron et al., 2022). Overall, these modifications allowed for the next set of improvements to be added. Experiments are run using the ViT-Large architecture on ImageNet-22k.

Training Data	INet-1k	Im-A	ADE-20k	Oxford-M	iNat2018	iNat2021	Places205
INet-22k	<b>85.9</b>	73.5	46.6	62.5	81.1	85.6	67.0
INet-22k \ INet-1k	85.3	70.3	46.2	58.7	80.1	85.1	66.5
Uncurated data	83.3	59.4	48.5	54.3	68.0	76.4	67.2
LVD-142M	85.8	<b>73.9</b>	<b>47.7</b>	<b>64.6</b>	<b>82.3</b>	<b>86.4</b>	<b>67.6</b>

Table 2: **Ablation of the source of pretraining data.** We compare the INet-22k dataset that was used in iBOT to our dataset, LVD-142M. Each model is trained for the same number of iterations, that is smaller than in our final run, without high-resolution adaptation. Pretraining on LVD-142M maintains the performance over INet-1k while leading to models that perform better in other domains.

and a linear probe in Table 1. Generally, we observe that each component improves the performance on either k-NN or linear probing and even both in most cases. Only LayerScale and Stochastic Depth incur a performance drop in linear probing but significantly improve the training stability in our experience.

## 6.2 Pretraining Data Source

The quality of features is directly related to the quality of the pretraining data. In this experiment, we probe the impact of LVD-142M compared to ImageNet-22k, a commonly used pretraining dataset, or using directly raw and uncurated data. For the uncurated dataset, we randomly sample 142 million images from the same data source as LVD-142M. We train a ViT-g/14 on each dataset for the same number of iterations. We also include a variant of ImageNet-22k obtained by removing the synsets of ImageNet-1k (INet-22k \ INet-1k) for completeness. We report the comparisons in Table 2.

The most salient observation is that training on a curated set of images works better on most benchmarks than training on uncurated data. This confirms the benefit of curating data, even in the case of self-supervised pretraining. When compared with models trained on ImageNet-22k, training on LVD-142M is also superior on all the benchmarks but ImageNet-1k. This confirms that training on a more diverse set of images improves the quality of the features in domains that are not covered by ImageNet-22k. We also see that training on our curated data increases the performances on domains that are not used for the curation process (iNaturalist 2018, 2021 and Places205), proving that scale and diversity can benefit unseen domains.

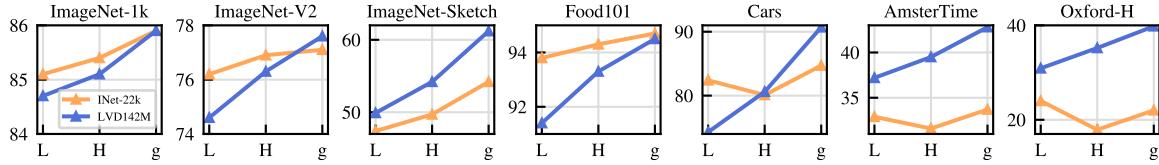


Figure 4: **Model scale versus data scale.** Evolution of performance as a function of model size for two different pretraining datasets: ImageNet-22k (14M images) and LVD-142M (142M images). The ViT-g trained on LVD-142M surpasses the ViT-g trained on ImageNet-22k on most benchmarks.

KoLeo	INet-1k	Im-A	ADE-20k	Oxford-M	MIM	INet-1k	Im-A	ADE-20k	Oxford-M
✗	85.3	70.6	47.2	55.6	✗	85.3	72.0	44.2	64.3
✓	85.8	72.8	47.1	63.9	✓	85.8	72.8	47.1	63.9

(a) Koleo loss

(b) MIM objective in iBOT

Table 3: (a) Effect of the KoLeo loss term. (b) Effect of the iBOT Masked Image Modeling (MIM) loss term. Evaluation performed on ImageNet-{1k,A} (classification with linear probe, accuracy %), ADE-20k (segmentation with linear layer, mIoU) and Oxford-M (image retrieval, mAP). Each model is trained on the same number of iterations, that is smaller than our final run. The KoLeo loss term improves nearest-neighbor search tasks (e.g. retrieval), and the MIM loss improves patch-level tasks (e.g. segmentation).

Overall, the conclusion of this ablation is that our dataset provides a good balance of different types of images that leads to the best performance overall.

### 6.3 Model Size and Data

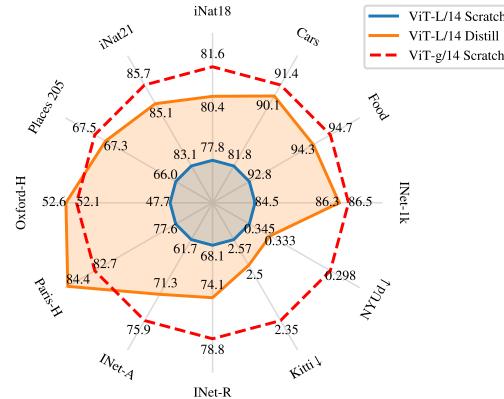
We quantify the importance of scaling data with the model size in Fig. 4. As the size of models grow, training on LVD-142M becomes more beneficial than training on ImageNet-22k. For instance, a ViT-g trained on LVD-142M matches the performance on ImageNet-1k of a model trained on ImageNet-22k while significantly outperforming it on the other benchmarks.

### 6.4 Loss Components

We validated the proposed technical improvements in Sec. 6.1 by adding them incrementally. This section analyzes the performance hit observed if we ablate specific loss terms, starting from our best-performing model. We ablate the importance of the KoLeo loss and the impact of the masked image modeling term. For both, we report performance on ImageNet-1k using a linear classifier, ADE-20k segmentation using a linear classifier, and nearest-neighbor image retrieval on Oxford-M. Table 3a shows the impact of using the KoLeo loss. We see that the instance retrieval performance improves by more than 8%, confirming that this term helps spread features in the output space. At the same time, the other metrics do not suffer from this regularization. In Table 3b, we show the impact of using the masked image modeling term from iBOT. This term is critical for dense prediction tasks, leading to almost 3% performance improvement.

### 6.5 Impact of Knowledge Distillation

For small architectures, we distill larger models instead of training them from scratch. We use the distillation procedure described in Sec. 5. We evaluate the effectiveness of this approach by comparing a ViT-L/14 trained from scratch with one distilled from a ViT-g/14 over 12 benchmarks in Fig. 5. We also report the performance of the ViT-g/14 used for distillation as a topline. The distilled model outperforms the one trained from scratch on all 12 benchmarks, validating our pretraining approach for small models.



(a) Comparison on individual metrics

Arch	Method	INet-1k	Segm.	Depth↓	Classif.
ViT-g/14	Scratch	86.5	73.4	1.00	92.1
ViT-L/14	Scratch	84.5	72.2	1.10	90.2
ViT-L/14	Distill	<b>86.3</b>	<b>73.3</b>	<b>1.08</b>	<b>91.2</b>

Arch	Method	Finegr.	Retriev.	ARSketch	Video
ViT-g/14	Scratch	78.3	75.2	77.0	69.3
ViT-L/14	Scratch	75.8	71.3	69.5	67.3
ViT-L/14	Distill	<b>77.6</b>	<b>76.3</b>	<b>74.5</b>	<b>67.5</b>

(b) Averaged metrics on 8 vision tasks

Figure 5: **Effectiveness of knowledge distillation.** Comparison between a ViT-L trained from scratch or distilled from DINOv2 using ViT-g/14. For reference, we also report the performance of the ViT-g/14 teacher. We show that a ViT-L model distilled from a frozen ViT-g outperforms a the same model trained from scratch on all benchmarks, sometimes even outperforming the distillation target.

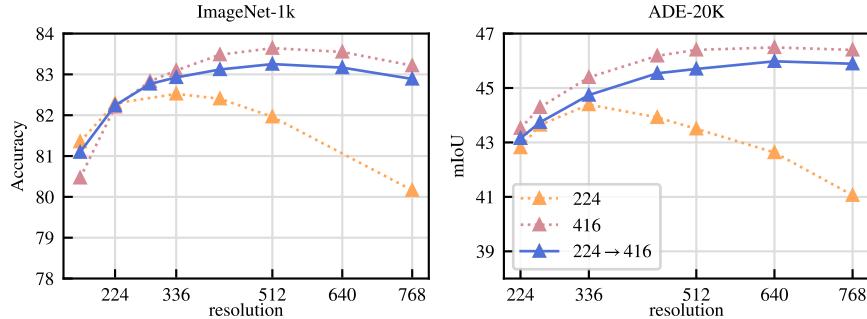


Figure 6: **Role of resolution.** Performance of ViT-L/16 trained on ImageNet-1k at fixed resolution (“224” and “416”) or trained at 224 then 416 for a short duration (“224→416”). We train linear classifiers on top of frozen features at different resolutions and report Top-1 accuracy on ImageNet and mIoU on ADE-20k. We observe that performing SSL training at high resolution for a short duration achieve behavior and results close to training at the same high resolution for the full training, at a fraction of the cost.

## 6.6 Impact of Resolution

We measure the impact of changing the resolution during the pretraining on the performance of image and patch-level features. We consider models trained from scratch using a fixed resolution of either  $224 \times 224$  or  $416 \times 416$ , and a model trained from scratch at  $224 \times 224$ , then resumed for 10k more iterations at  $416 \times 416$ . High-resolution training is compute-intensive, so we conduct this ablation on a small setup: a ViT-L/16 trained on ImageNet1k. In Fig. 6, we report the performance of a linear probe on ImageNet-1k and ADE-20k, evaluated at various resolutions. The model trained on high-resolution images performs best across resolutions, but this comes at a high cost: training at 416 is approximately  $3 \times$  more compute-intensive than training at 224. On the other hand, training at high resolution for only 10k iterations at the end of the training is almost as good and only requiring a fraction of the compute. As a consequence, we include this step at the end of the training rather than training at a high resolution from scratch.

## 7 Results

In this section, we present the empirical evaluation of our models on many image understanding tasks. We evaluate both global and local image representations, on category and instance-level recognition, semantic segmentation, monocular depth prediction, and action recognition. We detail the list of benchmarks in Appendix C. The goal of this evaluation is twofold. First, we show that our self-supervised features outperform the current state of the art by a very large margin. Second, we show that they match, or surpass the performance of weakly-supervised ones on a substantial number of tasks.

**Baselines.** In our comparisons, we use two kinds of models as baselines. We compare to the best performing self-supervised models that are openly available. First, we run our evaluations for MAE (He et al., 2022), DINO (Caron et al., 2021), SEERv2 (Goyal et al., 2022a), MSN (Assran et al., 2022), EsViT (Li et al., 2022a), Mugs (Zhou et al., 2022b) and iBOT (Zhou et al., 2022a). When several architectural variants were proposed for a given method, we report results for the one that leads to best top-1 accuracy on ImageNet-1k. Second, we report performance of open-source weakly-supervised models such as CLIP (Radford et al., 2021), OpenCLIP (Ilharco et al., 2021; Cherti et al., 2023), and SWAG (Singh et al., 2022). When evaluating models on ImageNet-1k, we report the performance for each of the aforementioned methods. For all other evaluations, we report the four best-performing models amongst SSL ones. Also, for reference, we report the best performing OpenCLIP-G for weakly-supervised ones.

### 7.1 ImageNet Classification

As a first evaluation, we probe the quality of the holistic image representation produced by the model on the ImageNet-1k classification dataset. We evaluate the quality of features by training a simple classifier over a frozen backbone, and do not perform finetuning of the backbone weights. Following previous work, we use a linear model for simplicity, ensuring a reproducible evaluation, despite the fact that classes may not be linearly separable. Because most SSL methods were developed using ImageNet-1k validation performance as a debugging signal, we also report the top-1 accuracy on ImageNet-ReAL and ImageNet-V2. In order to report this additional validation performance, for all models, we run the evaluation with our code. We compare our frozen features to the best publicly available SSL features in Table 4, regardless of architecture or pretraining data. We see the components proposed in this work lead to a very significant improvement (+4.2%) over the previous state of the art (iBOT ViT-L/16 trained on ImageNet-22k) on linear evaluation. At the same time, we also see that the performance increase on the alternative test sets is larger for our method, indicating stronger generalization. We describe details of our linear evaluation in Appendix B.3.

**How far are we from weakly-supervised models?** We also want to validate that our features are competitive with state-of-the-art open-source weakly supervised models. To this end, we compare on ImageNet-1k, using the linear evaluation, to three off-the-shelf methods with several architectural variants. For all models, we run the linear evaluation using our code, after making sure that our numbers match those reported in technical reports and papers. We show the result of this evaluation in Table 4. We see that our backbone, surpasses the performance of OpenCLIP with a ViT-G/14 architecture (+0.3%) and EVA-CLIP with a ViT-g/14 (+0.1%). At the same time, we also observe that our performance on the ImageNet-V2 test set is significantly better (+1.1% versus EVA-CLIP), indicating better generalization. For the remainder of this section, we report OpenCLIP-G as a reference for weakly-supervised models.

**Can we finetune the encoders?** We question if the ability of our models to produce high quality frozen features impact their performance when finetuned with supervision on a specific dataset. While this is not core to this paper, this experiment is indicative of whether we have involuntarily specialized our models to the setting of linear evaluations of frozen features. To run this sanity check, we apply the finetuning pipeline from Touvron et al. (2022), without tweaking hyper-parameters. In Table 5, we show that the Top-1 accuracy on the validation set of ImageNet-1k improves by more than +2% when the backbone is finetuned. This is true both when using models at resolution 224 and 448. Further gains can be obtained by tuning the hyper-parameters of the finetuning, but this is beyond the goal of this sanity check. Nonetheless, our best finetuned performance (88.9%) is only a couple of percent below (-2.2%) the absolute state of the

Method	Arch.	Data	Text sup.	kNN		linear	
				val	val	Real	V2
<b>Weakly supervised</b>							
CLIP	ViT-L/14	WIT-400M	✓	79.8	84.3	88.1	75.3
CLIP	ViT-L/14 <sub>336</sub>	WIT-400M	✓	80.5	85.3	88.8	75.8
SWAG	ViT-H/14	IG3.6B	✓	82.6	85.7	88.7	77.6
OpenCLIP	ViT-H/14	LAION-2B	✓	81.7	84.4	88.4	75.5
OpenCLIP	ViT-G/14	LAION-2B	✓	83.2	86.2	89.4	77.2
EVA-CLIP	ViT-g/14	custom*	✓	<b>83.5</b>	86.4	89.3	77.4
<b>Self-supervised</b>							
MAE	ViT-H/14	INet-1k	✗	49.4	76.6	83.3	64.8
DINO	ViT-S/8	INet-1k	✗	78.6	79.2	85.5	68.2
SEERv2	RG10B	IG2B	✗	—	79.8	—	—
MSN	ViT-L/7	INet-1k	✗	79.2	80.7	86.0	69.7
EsViT	Swin-B/W=14	INet-1k	✗	79.4	81.3	87.0	70.4
Mugs	ViT-L/16	INet-1k	✗	80.2	82.1	86.9	70.8
iBOT	ViT-L/16	INet-22k	✗	72.9	82.3	87.5	72.4
DINOv2	ViT-S/14	LVD-142M	✗	79.0	81.1	86.6	70.9
	ViT-B/14	LVD-142M	✗	82.1	84.5	88.3	75.1
	ViT-L/14	LVD-142M	✗	<b>83.5</b>	86.3	89.5	78.0
	ViT-g/14	LVD-142M	✗	<b>83.5</b>	<b>86.5</b>	<b>89.6</b>	<b>78.4</b>

Table 4: **Linear evaluation on ImageNet-1k of frozen pretrained features.** We report Top-1 accuracy on the validation set for publicly available models trained on public or private data, and with or without text supervision (text sup.). For reference, we also report the kNN performance on the validation set. We compare across any possible architectures (Arch.), at resolution  $224 \times 224$  unless stated otherwise. The dataset used for training EVA-CLIP is a custom mixture, see paper for details (Fang et al., 2023).

arts (91.1%), obtained by Chen et al. (2023a). As DINOv2 leads to features that are strong in both the linear and finetuning settings, a strong property of our approach is that *finetuning is optional*.

Arch.	Res.	Linear	Finetuned	$\Delta$
ViT-g/14	224	86.5	88.5	+2.0
	448	86.7	88.9	+2.2

Table 5: **Supervised finetuning on ImageNet-1k.** We use the pipeline of Touvron et al. (2022) to finetune our encoders on ImageNet-1k at resolutions  $224 \times 224$  or  $448 \times 448$ . We compare with the accuracy obtained with linear probing and observe only modest improvements with fine-tuning: this suggests that DINOv2 features already perform well out-of-the-box.

**Robustness analysis.** To complement our study, and probe the generalization of our features, we evaluate our ImageNet-1k models trained with linear classification heads on domain generalization benchmarks. We use the best performing linear classifier as described above and simply run inference on those benchmarks. Please note that most results in the literature are obtained with models that are finetuned end-to-end on ImageNet-1k. We show the result of this experiment in Table 6. When comparing with state-of-the-art SSL methods, our models shows drastically better robustness (+29.6% on A (Hendrycks et al., 2021b), +22.1% on R (Hendrycks et al., 2021a) and +23.0% on Sketch (Wang et al., 2019) compared to iBOT). Our model also improves upon the best weakly-supervised model on ImageNet-A while lagging behind on R and Sketch.

Method	Arch	Data	Im-A	Im-R	Im-C $\downarrow$	Sketch
OpenCLIP	ViT-G/14	LAION-2B	63.8	<b>87.8</b>	45.3	<b>66.4</b>
MAE	ViT-H/14	INet-1k	10.2	34.4	61.4	21.9
DINO	ViT-B/8	INet-1k	23.9	37.0	56.6	25.5
iBOT	ViT-L/16	INet-22k	41.5	51.0	43.9	38.5
DINOv2	ViT-S/14	LVD-142M	33.5	53.7	54.4	41.2
	ViT-B/14	LVD-142M	55.1	63.3	42.7	50.6
	ViT-L/14	LVD-142M	71.3	74.4	31.5	59.3
	ViT-g/14	LVD-142M	<b>75.9</b>	78.8	<b>28.2</b>	62.5

Table 6: **Domain Generalization with a linear probe** on top of frozen features at a resolution of 224. Higher numbers are better for all benchmarks except Im-C.

Feature	Arch	Image classification			Video classification		
		iNat2018	iNat2021	Places205	K400	UCF-101	SSv2
OpenCLIP	ViT-G/14	73.0	76.0	<b>69.8</b>	78.3	90.7	35.8
MAE	ViT-H/14	31.0	32.3	52.4	54.2	70.6	29.2
DINO	ViT-B/8	59.6	68.3	60.4	64.5	85.0	32.6
iBOT	ViT-L/16	66.3	74.6	64.4	72.6	88.6	<b>38.7</b>
DINOv2	ViT-S/14	69.0	74.2	62.9	67.8	87.0	33.1
	ViT-B/14	76.4	81.1	66.2	73.2	89.1	34.4
	ViT-L/14	80.4	85.1	67.3	76.3	90.5	35.6
	ViT-g/14	<b>81.6</b>	<b>85.7</b>	67.5	<b>78.4</b>	<b>91.2</b>	38.3

Table 7: **Linear evaluation on other image and video classification**. The image benchmarks contain a large quantity of fine-grained examples about objects or scenes. The video benchmarks cover action classification and human-object interaction. All the features are frozen with a linear probe on top.

## 7.2 Additional Image and Video classification Benchmarks

In this section we study the generalization of our features on downstream classification benchmarks. We consider two sets of evaluations in that context. On one hand, we use large and finegrained datasets such as iNaturalist and Places205. On the other, we use the 12 image classification tasks originally proposed in SimCLR (Chen et al., 2020). For iNaturalist 2018, iNaturalist 2021, and Places205, we train a linear classifier with data augmentations as in Sec. 7.1 We report top-1 accuracy for those three datasets in Table 7. Interestingly, our model significantly outperforms OpenCLIP ViT-G/14 on both variants of iNaturalist (+8.6% and +9.7% for 2018 and 2021 respectively), and lags slightly behind on Places 205 (-2.3%).

In a second set of evaluations, we measure the performance of our model on video action recognition even though our features were not trained on videos.. We evaluated features on three datasets, namely UCF-101 (Soomro et al., 2012), Kinetics-400 (Kay et al., 2017) and Something-Something v2 (Goyal et al., 2017). For this evaluation, we pick 8 evenly spaced frames in the video and train a linear classifier on the average of the features for UCF and K-400. For SSv2, we opt for concatenation to retain more temporal information than with feature averaging. For each dataset, we measure average accuracy and report the results in Table 7. We see that amongst self-supervised approaches, our model clearly sets a new state of the art. Moreover, our model matches the accuracy of the OpenCLIP features on UCF and Kinetics (+0.1% and +0.5% respectively) and clearly outperforms them on SSv2 (+2.5%). This is particularly interesting, as SSv2 requires a much richer understanding of the video frames.

Finally, in Table 8, we compare selected frozen features on 12 transfer classification benchmarks initially proposed by Chen et al. (2020). This benchmark covers scenes, objects (food, cars, planes), and textures.

Feature	Arch	Food	C10	C100	SUN	Cars	Aircr	VOC	DTD	Pets	Cal101	Flowers	CUB	Avg
OpenCLIP	ViT-G/14	94.5	98.7	91.0	<b>84.0</b>	<b>96.1</b>	80.2	<b>89.3</b>	<b>86.0</b>	95.7	<b>98.1</b>	99.5	89.9	91.9
MAE	ViT-H/14	78.4	96.1	83.9	63.9	56.1	63.4	84.3	75.4	89.4	95.9	92.3	57.2	78.0
DINO	ViT-B/8	85.1	97.2	86.9	70.3	76.6	70.6	86.7	79.6	93.2	95.4	97.6	81.7	85.1
iBOT	ViT-L/16	91.0	99.0	92.8	75.6	71.8	72.4	89.0	80.7	87.7	97.5	99.6	82.1	86.6
DINOv2	ViT-S/14	89.1	97.7	87.5	74.4	81.6	74.0	87.8	80.6	95.1	97.0	99.6	88.1	87.7
	ViT-B/14	92.8	98.7	91.3	77.3	88.2	79.4	88.2	83.3	96.2	96.1	99.6	89.6	90.1
	ViT-L/14	94.3	99.3	93.4	78.7	90.1	81.5	88.3	84.0	96.6	97.5	99.7	90.5	91.2
	ViT-g/14	<b>94.7</b>	<b>99.5</b>	<b>94.4</b>	78.7	91.4	<b>87.2</b>	89.0	84.5	<b>96.7</b>	97.6	<b>99.7</b>	<b>91.6</b>	<b>92.1</b>

Table 8: **Linear evaluation of frozen features on fine-grained benchmarks.** Accuracy on 12 benchmarks covering objects, scenes and textures following the evaluation protocol proposed in Chen et al. (2020).

Feature	Arch	Oxford		Paris		Met			AmsterTime	
		M	H	M	H	GAP	GAP-	ACC	mAP	
OpenCLIP	ViT-G/14	50.7	19.7	79.2	60.2	6.5	23.9	34.4	24.6	
MAE	ViT-H/14	11.7	2.2	19.9	4.7	7.5	23.5	30.5	4.2	
DINO	ViT-B/8	40.1	13.7	65.3	35.3	17.1	37.7	43.9	24.6	
iBOT	ViT-L/16	39.0	12.7	70.7	47.0	25.1	54.8	58.2	26.7	
DINOv2	ViT-S/14	68.8	43.2	84.6	68.5	29.4	54.3	57.7	43.5	
	ViT-B/14	72.9	49.5	90.3	78.6	36.7	63.5	66.1	45.6	
	ViT-L/14	<b>75.1</b>	<b>54.0</b>	<b>92.7</b>	<b>83.5</b>	<b>40.0</b>	68.9	71.6	<b>50.0</b>	
	ViT-g/14	73.6	52.3	92.1	82.6	36.8	<b>73.6</b>	<b>76.5</b>	46.7	

Table 9: **Evaluation of frozen features on instance-level recognition.** We consider 4 different benchmarks and report their main metrics.

We replace the Birdsnap dataset with CUB because the former was not publicly available in its entirety. We follow the experimental protocol as outlined by Chen et al. (2020), namely training a logistic regression on precomputed features. Our model significantly outperforms state-of-the-art SSL models, with most notable differences on Stanford Cars (+14.8% versus DINO ViT-B/8) and FGVC Aircraft (+14.8% versus iBOT ViT-L/16). Even though these benchmarks favor text-guided pretraining, our features are still competitive with OpenCLIP on most classification benchmarks, with the exception of a few datasets, especially SUN (−5.3%) and Cars (−4.7%).

### 7.3 Instance Recognition

In this experiment, we probe our model on the task of instance-level recognition using a non-parametric approach. Images from a database are ranked according to their cosine similarity with a query image. We evaluated our model and compare to baselines on Paris and Oxford, that are landmark recognition benchmarks. We also evaluated on Met, a dataset of artworks from the Metropolitan museum, and AmsterTime, containing street view images matched to archival images of Amsterdam. We measure performance by computing the mean average precision and report our results in Table 9. We see that our features significantly outperform both SSL (+41% mAP on Oxford-Hard), and weakly-supervised (+34% mAP on Oxford-Hard) ones. It is interesting to see that our features perform well across task granularities, both at the category-level and instance-level. This is a desirable property for strong off-the-shelf computer vision features.

Method	Arch.	ADE20k (62.9)		CityScapes (86.9)		Pascal VOC (89.0)	
		lin.	+ms	lin.	+ms	lin.	+ms
OpenCLIP	ViT-G/14	39.3	46.0	60.3	70.3	71.4	79.2
MAE	ViT-H/14	33.3	30.7	58.4	61.0	67.6	63.3
DINO	ViT-B/8	31.8	35.2	56.9	66.2	66.4	75.6
iBOT	ViT-L/16	44.6	47.5	64.8	74.5	82.3	84.3
DINOv2	ViT-S/14	44.3	47.2	66.6	77.1	81.1	82.6
	ViT-B/14	47.3	51.3	69.4	80.0	82.5	84.9
	ViT-L/14	47.7	<b>53.1</b>	70.3	80.9	82.1	86.0
	ViT-g/14	<b>49.0</b>	53.0	<b>71.3</b>	<b>81.0</b>	<b>83.0</b>	<b>86.2</b>

Table 10: **Semantic segmentation on ADE20K, CityScapes and Pascal VOC with frozen features** and a linear classifier (lin.) and with multiscale (+ms). The absolute state of the art – from Wang et al. (2022), Liu et al. (2021) and Chen et al. (2018) respectively – are mentioned at the top of the Table. For reference, using the Mask2Former pipeline (Steiner et al., 2021) with a ViT-Adapter (Chen et al., 2023b) on top of our frozen ViT-g/14 backbone gives 60.2 mIoU on ADE-20k.

#### 7.4 Dense Recognition Tasks

We probe the quality of patch-level features extracted from our network on several dense downstream tasks. We consider semantic image segmentation and monocular depth estimation in several settings and we conduct evaluations on multiple datasets for each.

**Semantic segmentation.** For our semantic segmentation evaluation, we consider two different setups. **Linear:** a linear layer is trained to predict class logits from a patch tokens. It is used to produce a low-resolution logit map (eg 32x32 for a model with patch size 16), which is then upsampled to full resolution (512x512) to obtain a segmentation map. This procedure is extremely simple but cannot easily produce high-resolution segmentations. **+ms:** a boosted version of the linear setup. We concatenate the patch tokens of the 4 last layers, use a larger image resolution of 640, and use multiscale test-time augmentations to improve the predictions. We report the performance of our model variants as well as the baselines on three datasets under the two setups in Table 10.

Our models show very good performance on all datasets and for all setups. Interestingly, our evaluation using **+ms** is on par with fully finetuning MAE with an Upernet decoder (53.0 versus 53.6 mIoU). This is surprising because we use a significantly simpler predictor. Also, our best model, when evaluated using the boosted recipe, almost matches the state of the art on Pascal VOC (86.2 versus 89.0 mIoU).

**Frozen backbone in a SOTA pipeline.** In a final experiment, we freeze our backbone, and plug it into a ViT-Adapter Chen et al. (2023b) with a Mask2former head (Cheng et al., 2022). We tune the weights of the adapter and head, but keep the backbone frozen, meaning 66% of the weights are frozen. This allows for a lighter segmentation training than full end-to-end fine-tuning. With this setup, we reach 60.2 mIoU on ADE20k, close to the competitive state of the art, standing at 62.9 mIoU (Wang et al., 2022). Although our setup for this experiment doesn’t makes use of the optimisations described in Sec. 5, the segmentation training in this experiment took 28 hours on 16 V100 GPUs.

**Depth estimation.** In this experiment, we evaluate our patch-level features on three monocular depth estimation benchmarks: NYUD, KITTI and zero-shot transfer from NYUD to SUN3d. We follow the evaluation protocol of Li et al. (2022b). We consider three different setups for this evaluation. **lin. 1:** we extract the last layer of the frozen transformer and concatenate the [CLS] token to each patch token. Then we bi-linearly upsample the tokens by a factor of 4 to increase the resolution. Finally we train a simple linear layer using a classification loss by dividing the depth prediction range in 256 uniformly distributed bins and

Method	Arch.	NYUd (0.330)			KITTI (2.10)			NYUd → SUN RGB-D (0.421)		
		lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT
OpenCLIP	ViT-G/14	0.541	0.510	0.414	3.57	3.21	2.56	0.537	0.476	0.408
MAE	ViT-H/14	0.517	0.483	0.415	3.66	3.26	2.59	0.545	0.523	0.506
DINO	ViT-B/8	0.555	0.539	0.492	3.81	3.56	2.74	0.553	0.541	0.520
iBOT	ViT-L/16	0.417	0.387	0.358	3.31	3.07	2.55	0.447	0.435	0.426
DINOv2	ViT-S/14	0.449	0.417	0.356	3.10	2.86	2.34	0.477	0.431	0.409
	ViT-B/14	0.399	0.362	0.317	2.90	2.59	2.23	0.448	0.400	0.377
	ViT-L/14	0.384	0.333	0.293	2.78	2.50	2.14	0.429	0.396	0.360
	ViT-g/14	<b>0.344</b>	<b>0.298</b>	<b>0.279</b>	<b>2.62</b>	<b>2.35</b>	<b>2.11</b>	<b>0.402</b>	<b>0.362</b>	<b>0.338</b>

Table 11: **Depth estimation with frozen features.** We report performance when training a linear classifier on top of one (lin. 1) or four (lin. 4) transformer layers, as well, as the DPT decoder (DPT) of Ranftl et al. (2021). We report the RMSE metric on the 3 datasets. Lower is better. For reference, we report state-of-the-art results taken from Li et al. (2022b) on each benchmark on top of the Table.

use a linear normalization following Bhat et al. (2021). **lin. 4**: we use the same protocol that we use with one layer, but concatenate the tokens from layers  $l = \{3, 6, 9, 12\}$  for ViT-S/B,  $l = \{5, 12, 18, 24\}$  for ViT-L, and  $l = \{10, 20, 30, 40\}$  for ViT-g. **DPT**: we use the DPT decoder (Ranftl et al., 2021) on top of our frozen models and setup a regression task. We scale the size of the head following the dimension of the features for each architecture. We show results for all baselines, all datasets and all setups in Table 11.

From this table, we see that our features clearly surpass the best SSL and WSL features available. It is interesting to see that iBOT features extracted from a ViT-L outperform the ones of OpenCLIP with a ViT-G. This observation supports the intuition that caption-based feature learning fails to learn subtle patterns like this one. Also, our model, with the DPT decoder and frozen backbone, matches or exceeds the performance of the recent work of Li et al. (2022b). Finally, the out-of-domain generalization result on SUN-RGBd shows that our features allow very good transfer between domains. A depth prediction module trained on indoor scenes from NYUd generalizes pretty well to the outdoor examples of SUN-RGBd.

## 7.5 Qualitative Results

In this final section of the empirical evaluation of our features, we propose a few qualitative analyses.

**Semantic Segmentation and Depth Estimation.** We show some qualitative results for our dense prediction evaluations: segmentation on ADE20K in Fig. 7 and depth estimation on NYUd, KITTI and SUN RGB-D in Fig. 7. We compare DINOv2 with OpenCLIP with a linear classifier on each dataset. While not perfect, the linear segmentation model using our DINOv2 backbone produces good results and behaves much better than the OpenCLIP one under this evaluation setup. Indeed, the segmentation mask produced by OpenCLIP-G shows many artifacts and disconnected components. The qualitative results on depth estimation clearly illustrate the quantitative gap between OpenCLIP and DINOv2. These results highlight that our features, as well as the features extracted from OpenCLIP, are able to linearly separate complex information such as depth, even though neither was trained with this type of information. However, our features lead to a much smoother depth estimation, with less artifacts. Some objects such as the chair on the SUN RGB-D image are completely ignored by OpenCLIP and correctly positioned using our features.

**Out-of-distribution generalization.** We show a few examples of applying the depth prediction and segmentation linear classifiers to out-of-distribution examples in Fig. 8. The qualitative results support our claim that our features transfer between domains. The quality of the depth and segmentation predicted for pictures of animals, or paintings is very good, even though the domains are very different.

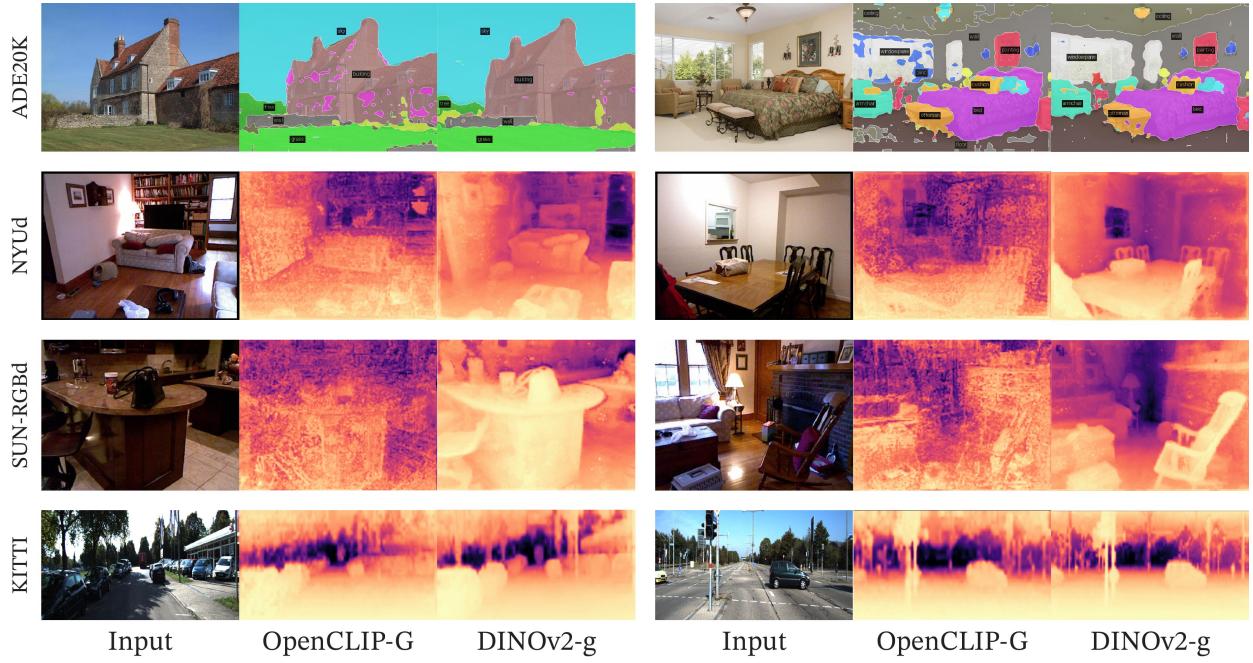


Figure 7: **Segmentation and depth estimation with linear classifiers.** Examples from ADE20K, NYUD, SUN RGB-D and KITTI with a linear probe on frozen OpenCLIP-G and DINOv2-g features.



Figure 8: **Examples of out-of-distribution examples** with frozen DINOv2-g features and a linear probe.

**PCA of patch features.** We show the results of the principal component analysis (PCA) performed on the patch features extracted by our model. We keep only patches with a positive value after we threshold the first component. This procedure turns out to separate the image’s main object from the background. We compute a second PCA on the remaining patches across three images depicting the same category. We color the three first components with three different colors and present the results in Fig. 1 and 9. There are two interesting observations: first, our unsupervised foreground / background detector, based on detecting the highest variance direction, performs very well and is capable of delineating the boundary of the main object in the picture. Second, the other components correspond to “parts” of objects and match well for images of the same category. This is an emerging property – our model was not trained to parse parts of objects.

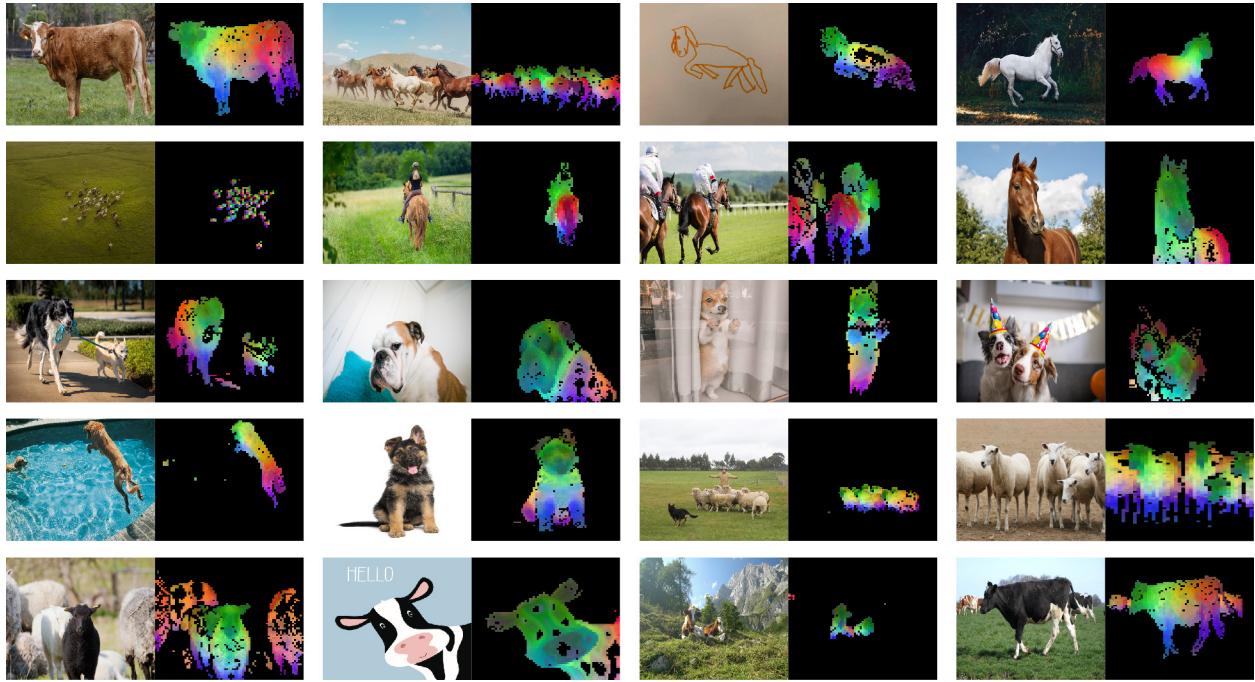


Figure 9: **More visualization of the first PCA components.** We compute the PCA between the patches from all of the images and show their first 3 components. Each component corresponds to a specific color channel. Same parts are matched between related images despite changes of pose, style or even objects. Background is removed by removing patches with a negative score of the first PCA component.

**Patch matching.** Finally, we explore what type of information our patch-level features contain by matching them across images. We start by detecting the foreground object using the procedure described above. Then, we compute the euclidean distance between patch features extracted from two images and map them by solving an assignment problem. In order to reduce the number of matches, we then apply a non-maximum suppression to keep only the salient ones. In Fig. 10, we show some examples of such matchings.

We observe that the features seem to capture information about semantic regions that serve similar purpose in different objects or animals. For instance, the wing of a plane matches the wing of a bird. We also observe that the model is robust to style (image versus drawing), and to large variation of poses (see the elephant).

## 8 Fairness and Bias Analysis

We conduct two fairness evaluations of our models. We probe for geographical fairness and potential harmful label associations. For both evaluations, we experiment with our largest ViT-g model.

### 8.1 Geographical Fairness

We evaluate geographical fairness on the Dollar Street dataset introduced in De Vries et al. (2019) using the evaluation protocol of Goyal et al. (2022b). This benchmark compares performance across countries and income levels. It contains 16,073 images from 289 households across 54 countries. The task is to recognize 94 concepts that vary visually between households based on income or location. In Table 12, we compare our model with SEERv2 (Goyal et al., 2022a), a model trained on a geographically diverse set of images. Our model is slightly fairer across regions and incomes than the SEERv2 model and significantly better than the supervised baseline reported by Goyal et al. (2022a). However, we still observe a significant difference between regions, particularly in Africa, where our model performance drops by 25.7% compared to Europe. This shows that our model is still biased toward Western countries. Similarly, our model performs

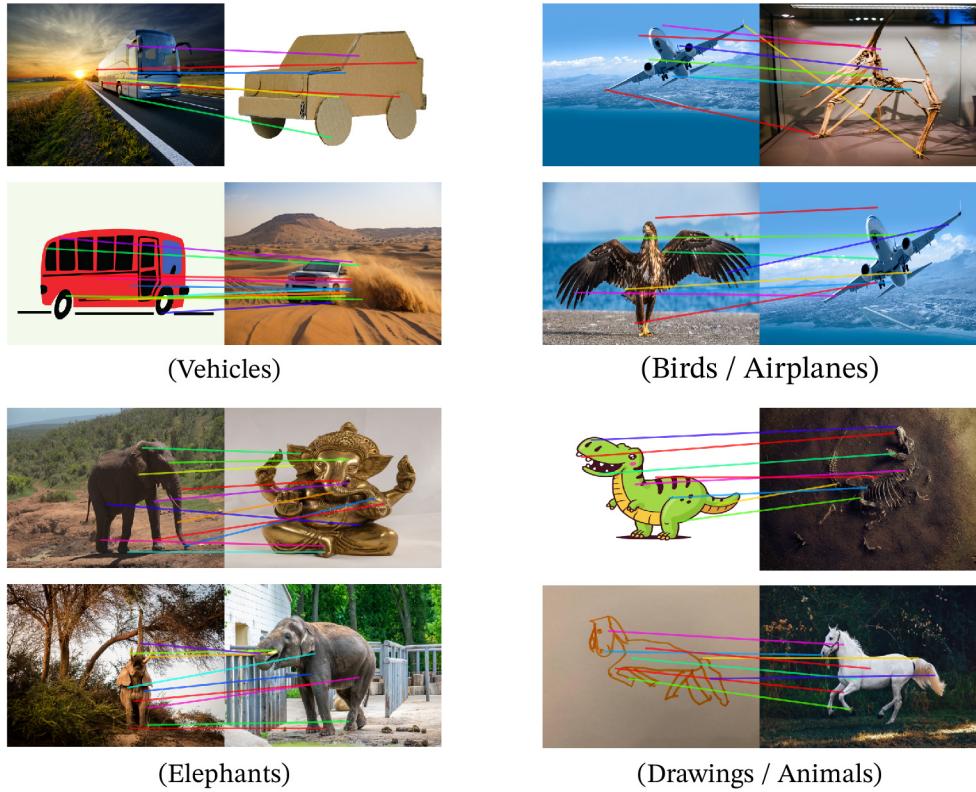


Figure 10: **Matching across images.** We match patch-level features between images from different domains, poses and even objects that share similar semantic information. This exhibits the ability of our model to transfer across domains and understand relations between similar parts of different objects.

Method	Arch.	Data	Income buckets			Regions			
			low	medium	high	Africa	Asia	Americas	Europe
SEERv2	RG-10B	IG-1B	59.7	78.5	86.6	65.9	76.3	81.1	85.6
DINOv2	ViT-g/14	LVD-142M	67.4	83.3	90.5	74.0	81.6	86.2	89.7

Table 12: **Geographical fairness and diversity analysis across income buckets and regions.**

significantly better on high-income households than low-income ones, with a difference of 31.7%. Despite improvements, we observe significant biases in our models toward wealthy households from Western countries.

## 8.2 Gender, Skintones and Age

In a second set of evaluations, we question how our model classifies images of people of different gender, skin tone, and age (all self-reported). We follow the protocol of Goyal et al. (2022b), where we train a multiclass classifier on a subset of 619 classes of ImageNet-22k. We group the 619 classes into four broader categories: Human, Possibly Human, Non-Human, or Crime. Non-Human and Crime are considered harmful. Using this classifier, we run inference on 2955 images from the Casual Conversations dataset (Hazirbas et al., 2021) and keep all labels in the top-5 that are assigned a probability of 0.1 or more. Because of that, we can assign multiple classes to each image. We make one modification to the original evaluation protocol: we do not backpropagate gradients to the backbone and keep it frozen. We compare our model to SEERv2 in Table 13.

Model	Assoc.	Gender Skintone				Age Groups			
		female darker	female lighter	male darker	male lighter	18-30	30-45	45-70	70+
SEER	Non-Human	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RG-10B	Crime	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Human	94.9	95.8	86.6	79.0	90.5	88.3	91.9	82.3
	Possibly-Human	13.6	6.7	65.0	60.2	32.8	37.2	29.4	6.5
DINOv2	Non-Human	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ViT-g/14	Crime	0.0	0.0	0.2	0.0	0.0	0.1	0.0	0.0
	Human	97.3	97.7	86.1	84.0	91.2	90.2	93.2	88.7
	Possibly-Human	15.8	17.2	52.2	48.1	35.3	37.3	23.0	9.7

Table 13: **Label association fairness evaluation across gender, skintones and age groups.** We follow the protocol proposed by Goyal et al. (2022b) with a slight modification. Instead of finetuning the backbone, we simply learn a linear classifier on the subset of 619 classes of ImageNet-22k.

Model to Reproduce	GPU Type	GPU Power consumption	GPU-hours	PUE	Total power consumption	Carbon emitted (tCO <sub>2</sub> eq)
DINOv2-g	A100-40GB	400W	22,016	1.1	9.7 MWh	3.7

Table 14: **Carbon footprint of reproducing DINOv2.** We report the potential carbon emission of reproducing DINOv2-g when assuming a power consumption for the A100-40GB of 400W, a PUE of 1.1 and carbon intensity factor of 0.385 kg CO<sub>2</sub>e per KWh.

Our model often classifies images of all groups as Human without large deviations across skin tones. Neither SEERv2 nor DINOv2 predict harmful labels from the Non-Human or Crime meta-categories (except for two instances where the background contains bars visually similar to prison bars). We see that our model triggers the Possibly-Human classes often. This class is constructed from objects in ImageNet-22k that are often related to Humans, such as Scarf, Glasses, or Beard. Our model often predicts the Possibly-Human class for men because of the prevalence of the Beard class. No clear pattern indicates a bias against a particular group in this study. While this is encouraging, we also acknowledge that a more thorough evaluation of biases may reveal flaws in our model.

## 9 Estimating the Environmental Impact of Training our Models

Training foundation models consumes a significant amount of energy, resulting in carbon dioxide emissions. Patterson et al. (2021) propose a methodology to report an estimation of the carbon emitted during the training of a model based on the specifics of the data center and its power grid. This computation informs the design of the data center used for the training of models and the choice of location for data centers. This methodology requires to know the specifics of the data center used for training, which can be complex when multiple data centers are involved over time. Additionally, these specifics are most often not in the control of the AI practitioner, and hence, this methodology is less helpful when practitioners make technical decisions about future trainings. Instead, in this section, we follow an alternative that reports the potential carbon emission of retraining a similar model in an average data center located in the US. This methodology was used in previous work in natural language processing (Strubell et al., 2019; Touvron et al., 2023) to establish an apple-to-apple comparison between pretraining schemes. More precisely, we fix the value of all exogenous variables, i.e., the Power Usage Effectiveness (PUE) and carbon intensity factor of a power grid to the same values as in Touvron et al. (2023), that is, a PUE of 1.1 and the carbon intensity factor to the US average of 0.385 kg CO<sub>2</sub>eq/KWh. We use the same formula as in Patterson et al. (2021) to estimate the potential energy consumption and the carbon emission. For the power consumption of an A100-80GB, we take the thermal design power for NVLink systems, which is 400W. We report the potential carbon emission

of retraining a DINOv2 ViT-g in Table 14. For comparison, retraining an OpenCLIP ViT-L or OpenCLIP ViT-G would require 22.4 MWh and 118.9 MWh, respectively, if run in the same data center. This is  $10\times$  more carbon emission. Note that this comparison is not fair to them, since they also train a text encoder in parallel, and we thus do not report them in the table. However, it gives a reasonable guideline for those who are interested in training only visual features: in this context, training a self-supervised model is preferable in terms of carbon emission. Training a text-guided model still makes sense when planning to reuse the text encoder.

**Carbon footprint of the whole project.** Additionally, we estimate the footprint of the whole project to be between 0.5k and 1k tCO<sub>2</sub>eq using the same grid as presented above <sup>3</sup>. This carbon footprint represents in the order of 200k GPU-days. The primary sources of emissions are the self-supervised pre-trainings of the models. For example, a single pre-training of a ViT-g model (22k GPU-hours) emits 3.7 tons of CO<sub>2</sub>eq, while a finetuning on ImageNet-1k (1k GPU-hours) emits 0.2 tons. This estimate only considers the GPUs’ electricity consumption and ignores other emissions, such as their manufacturing and disposal.

## 10 Future work and Discussion

In this work, we present DINOv2, a new series of image encoders pretrained on large curated data with no supervision. This is the first SSL work on image data that leads to visual features that close the performance gap with (weakly) supervised alternatives across a wide range of benchmarks and without the need for finetuning. We can attribute the strong performance of the DINOv2 family of models to several factors: **i**) an improved training recipe with better hyperparameters and regularization (Table 1), **ii**) a larger model scale with improved results regardless of the data used for training (Fig. 4), **iii**) a larger dataset (Fig. 4) and **iv**) the distillation process that makes smaller models benefit from the performance of the strongest ViT-g model (Fig. 5). A few properties emerge from these models, such as an understanding of object parts and scene geometry regardless of the image domains. We expect that more of these properties will emerge at larger scales of models and data, akin to instruction emergence in large language models, and plan to continue scaling along these axes. This paper also demonstrates that these visual features are compatible with classifiers as simple as linear layers - meaning the underlying information is *readily available*. In future work, we plan to leverage this ability to train a language-enabled AI system that can process visual features as if they were word tokens, and extract the required information to ground the system.

### Acknowledgments.

We thank Mathilde Caron for initial discussions that led to this work. Julien Mairal was supported by the ERC grant number 101087696 (APHELAIA project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003). We thank Olivia Joulin for the horse drawing used in Fig. 10. We thank Madeleine and Léon for posing for Fig. 8. We also thank the rest of FAIR and Meta AI for feedback on this work through the entire project.

## References

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. In *ECCV workshop on "What is Motion For?",* 2022.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022.

<sup>3</sup>For context, a full Boeing 777 return flight between London and New York corresponds to approximately 560 tCO<sub>2</sub>eq.

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 *CVPR*, 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 *ICML*, 2022.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021.

Jan Beirlant, Edward J Dudewicz, László Györfi, Edward C Van der Meulen, et al. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

Maxim Berman, Hervé Jégou, Vedaldi Andrea, Iasonas Kokkinos, and Matthijs Douze. MultiGrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *CVPR*, 2023.

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *CVPR*, 2021.

Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

Zhe Chen, Yuchen Duan, Wenhui Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *ICLR*, 2023b.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*, 2022.

Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. Does object recognition work for everyone? In *CVPR workshops*, 2019.

Sylvain Delattre and Nicolas Fournier. On the kozachenko–leonenko entropy estimator. *Journal of Statistical Planning and Inference*, 185:69–93, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 2016.

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 *ICLR*, 2021.

Matthijs Douze, Hervé Jegou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*, 2009.

Quentin Duval, Ishan Misra, and Nicolas Ballas. A simple recipe for competitive low-compute self supervised vision models. *arXiv preprint arXiv:2301.09451*, 2023.

Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR*, 2004.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. In *CVPR*, 2023.

Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019.

Priya Goyal, Mathilde Caron, Benjamin Lefauze, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *preprint arXiv:2103.01988*, 2021.

Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022a.

Priya Goyal, Adriana Romero Soriano, Caner Hazirbas, Levent Sagun, and Nicolas Usunier. Fairness indicators for systematic assessments of visual feature extractors. In *FAcct*, 2022b.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022.

Caner Hazirbas, Joanna Bitton, Brian Dolhansky, Jacqueline Pan, Albert Gordo, and Cristian Canton Ferrer. Towards measuring fairness in ai: the casual conversations dataset. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *PMLR*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2014.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. 2021.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *TPAMI*, 2010.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *3DRR*, 2013.

Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Benjamin Lefauze, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. In *ICLR*, 2022a.

Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022b.

Tatiana Likhomanenko, Qiantong Xu, Gabriel Synnaeve, Ronan Collobert, and Alex Rogozhnikov. Cape: Encoding relative positions with continuous augmented positional embeddings. In *NeurIPS*, 2021.

Huajun Liu, Fuqiang Liu, Xinyi Fan, and Dong Huang. Polarized self-attention: towards high-quality pixel-wise regression. *arXiv preprint arXiv:2107.00782*, 2021.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *CVPR*, 2023.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*, 2022.

Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018a.

Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *TPAMI*, 2018b.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019.

Yangjun Ruan, Saurabh Singh, Warren Morningstar, Alexander A Alemi, Sergey Ioffe, Ian Fischer, and Joshua V Dillon. Weighted ensemble self-supervised learning. In *ICLR*, 2023.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *ICLR*, 2019.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPS Data Centric AI Workshop*, 2021.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting Weakly Supervised Pre-Training of Visual Perception Models. In *CVPR*, 2022.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *TMLR*, 2021.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *ACL*, 2019.

Yonglong Tian, Olivier J Henaff, and Aäron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *ICCV*, 2021.

Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, 2022.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *CVPR*, 2021.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2022.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

Frederik Warburg, Soren Hauberg, Manuel Lopez-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *CVPR*, 2020.

Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning super-features for image retrieval. In *ICLR*, 2021.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, 2010.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *LREC*, 2020.

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, Christoph Feichtenhofer, et al. Masked autoencoders that listen. *arXiv preprint arXiv:2207.06405*, 2022.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

Burak Yildiz, Seyran Khademi, Ronald Maria Siebes, and Jan van Gemert. Amstertime: A visual place recognition benchmark dataset for severe domain shift. In *ICPR*, 2022.

Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, 2014.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022a.

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*, 2022b.

## A Data Processing

### A.1 Data selection

Our selection of datasets for building LVD-142M is detailed in Tab. 15. This collection is intended to provide images covering well various downstream vision tasks both for image-level and dense recognition.

### A.2 Image similarity

We employ cosine similarity to compare image features (whether ours or feature generated for deduplication) with the following similarity function  $m$ :

$$m(s, r) = \text{cosine-similarity}(f(s), f(r)) = \frac{f(s) \cdot f(r)}{\|f(s)\|_2 \|f(r)\|_2}$$

where  $s$  and  $r$  are a pair of images to compare and  $f$  is the model generating features.

### A.3 Deduplication

**Self-deduplication.** To deduplicate our uncurated data source of 1.3B images, we compute and use the embeddings generated by Pizzi et al. (2022) and retrieve the  $k = 64$  nearest neighbors of each image (using cosine similarity). Considering only neighbors with a similarity  $>0.6$ , we extract the connected components of the associated  $k$ -NN graph thanks to a scalable disjoint set data structure implementation. We then only keep one representative for each component of duplicate images. This results in a self-deduplicated data source of 1.1B images.

**Relative deduplication** To reduce redundancy and also properly evaluate the performance of our features, we discard remaining images of our self-deduplicated data source that are too similar to train and test splits of our evaluation datasets. To achieve this, we apply a similar procedure as for self-deduplication, with a stricter similarity  $>0.45$ , this time identifying the duplicate components (if any) to which each reference image belong and discarding it entirely. This results in a self- and relatively-deduplicated data source of 744M images.

### A.4 Retrieval

We employ two approaches to augment dataset via retrieval: sample-based and cluster-based. The first one, sample-based, applies to datasets larger than 1M images and consists in collecting a fixed number  $k$  of nearest images for each sample image of the dataset to retrieve, effectively trying to multiply by  $k$  the size of the dataset. We use  $k = 4$  for Google Landmarks v2 and ImageNet-22k but a larger  $k = 32$  to make this specific retrieval a core part of our LVD-142M dataset. For smaller datasets, the second approach, cluster-based, consists in first clustering our uncurated data source into 100,000 separate clusters thanks to a distributed  $k$ -means implementation. Each cluster should capture different types of image concept and contents. We then pick 10,000 images from each cluster associated with more than 3 images of the retrieved dataset. As this can result in a very large number of retrieved images for some dataset, we restrict such retrievals to a maximum of 1M images to maintain the balance between the different datasets within LVD-142M.

## B Implementation Details

### B.1 Unsupervised pre-training

For unsupervised pre-training we build on the DINO and iBOT codebases. We use hyperparameters shown in Table 16, ViT architectures described in Table 17.

**KoLeo regularization.** We apply the KoLeo regularizer with a weight of 0.1 between the class tokens of the first global crop, for all samples within a GPU without cross-communication for this step.

Task	Dataset / Split	Images	Retrieval	Retrieved	Final
classification	ImageNet-22k / –	14,197,086	as is	–	14,197,086
classification	ImageNet-22k / –	14,197,086	sample	56,788,344	56,788,344
classification	ImageNet-1k / train	1,281,167	sample	40,997,344	40,997,344
fine-grained classif.	Caltech 101 / train	3,030	cluster	2,630,000	1,000,000
fine-grained classif.	CUB-200-2011 / train	5,994	cluster	1,300,000	1,000,000
fine-grained classif.	DTD / train1	1,880	cluster	1,580,000	1,000,000
fine-grained classif.	FGVC-Aircraft / train	3,334	cluster	1,170,000	1,000,000
fine-grained classif.	Flowers-102 / train	1,020	cluster	1,060,000	1,000,000
fine-grained classif.	Food-101 / train	75,750	cluster	21,670,000	1,000,000
fine-grained classif.	Oxford-IIIT Pet / trainval	3,680	cluster	2,750,000	1,000,000
fine-grained classif.	Stanford Cars / train	8,144	cluster	7,220,000	1,000,000
fine-grained classif.	SUN397 / train1	19,850	cluster	18,950,000	1,000,000
fine-grained classif.	Pascal VOC 2007 / train	2,501	cluster	1,010,000	1,000,000
segmentation	ADE20K / train	20,210	cluster	20,720,000	1,000,000
segmentation	Cityscapes / train	2,975	cluster	1,390,000	1,000,000
segmentation	Pascal VOC 2012 (seg.) / trainaug	1,464	cluster	10,140,000	1,000,000
depth estimation	Mapillary SLS / train	1,434,262	as is	–	1,434,262
depth estimation	KITTI / train (Eigen)	23,158	cluster	3,700,000	1,000,000
depth estimation	NYU Depth V2 / train	24,231	cluster	10,850,000	1,000,000
depth estimation	SUN RGB-D / train	4,829	cluster	4,870,000	1,000,000
retrieval	Google Landmarks v2 / train (clean)	1,580,470	as is	–	1,580,470
retrieval	Google Landmarks v2 / train (clean)	1,580,470	sample	6,321,880	6,321,880
retrieval	AmsterTime / new	1,231	cluster	960,000	960,000
retrieval	AmsterTime / old	1,231	cluster	830,000	830,000
retrieval	Met / train	397,121	cluster	62,860,000	1,000,000
retrieval	Revisiting Oxford / base	4,993	cluster	3,680,000	1,000,000
retrieval	Revisiting Paris / base	6,322	cluster	3,660,000	1,000,000
					142,109,386

Table 15: **Composition of our LVD-142M dataset.** We report the list of datasets and associated splits used to build the dataset, how they were included (as is without retrieval or via sample-based or cluster-based retrieval). For retrievals, we indicate the actual number of retrieved images and the final number included in the dataset. We chose to include as many datasets as possible in the pretraining data in order to cover as many domains as possible. We kept a few datasets aside in order to evaluate performance outside of the pretraining domain. More details about dataset usages can be found in Table 18.

	Arch.	Drop-rate	LR	Batch size
DINOv2-S (distilled)	ViT-S/14	0	1e-3	2048
DINOv2-B (distilled)	ViT-B/14	0	1e-3	2048
DINOv2-L (distilled)	ViT-L/14	0	1e-3	2048
DINOv2-L (from scratch)	ViT-L/14	0.4	3.5e-4	3072
DINOv2-g (from scratch)	ViT-g/14	0.4	3.5e-4	3072

Table 16: **Training hyperparameters for DINOv2-S, DINOv2-B, DINOv2-L and DINOv2-g.** All models run for 625k iterations with optimizer AdamW, an initial LayerScale value of 1e-5, a weight decay cosine schedule from 0.04 to 0.2, a learning rate warmup of 100k iterations, a teacher momentum cosine schedule from 0.994 to 1, and we train in float16 precision in all cases (except for the DINO heads where we reduce the gradients in float32).

Arch.	Embed dim	Heads	Blocks	FFN layer
ViT-S/14 (distilled)	384	6	12	MLP
ViT-B/14 (distilled)	768	12	18	MLP
ViT-L/14 (distilled)	1024	16	24	MLP
ViT-L/14 (from scratch)	1024	16	24	SwiGLU
ViT-g/14 (from scratch)	1536	24	40	SwiGLU

Table 17: **Architecture details of the ViT-S/B/L/g networks used in this work.** We use MLP feed-forward networks for distilled models, and SwiGLU (Shazeer, 2020) when training from scratch.

**EMA update for the teacher.** The teacher is initialized with the same state as the student, and is an exponential moving average of the student network, with a momentum value in [0.994, 1.0] following a cosine schedule. It is updated at the end of every training step.

## B.2 High-Resolution adaptation

We initialise the model with the pretrained weights then train it for 10k iterations with the same procedure as the original pretraining. All the schedules are kept the same as in the original training, but compressed to fit in 10k iterations. All the hyperparameters are kept the same as in the first pretraining, except the base learning rate which is reduced.

## B.3 Linear probing evaluation

For linear probing we define 3 evaluation parameters: the learning rate, how many output layers we use, whether we concatenate the average-pooled patch token features with the class token (or use only the class token). We train our linear layer with SGD for 12500 iterations, using random-resized-crop data augmentation, and perform the following grid search:

- learning rate in  $\{0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$
- output layers in  $\{1, 4\}$
- concatenate average-pooled tokens in  $\{yes, no\}$

We then report the highest accuracy value obtained on the validation set as is common practice. Note that this grid search is not expensive, because at each iteration we perform inference on the backbone only once, then feed the output to all linear classifiers (each performing a single matrix multiplication).

## C List of Datasets used

We show in Table 18 the list of benchmarks and datasets used and their purposes.

Dataset	Pretraining (as is)	Retrieving pretraining data	Eval.	Task	Citation
ImageNet-1k	✗	✓	✓	Classif.	(Russakovsky et al., 2015)
ImageNet-22k	✓	✓	✗		(Deng et al., 2009)
ImageNet-V2	✗	✗	✓	Classif.	(Recht et al., 2019)
ImageNet-ReAL	✗	✗	✓	Classif.	(Beyer et al., 2020)
ImageNet-A	✗	✗	✓	Classif.	(Hendrycks et al., 2021b)
ImageNet-C	✗	✗	✓	Classif.	(Hendrycks & Dietterich, 2019)
ImageNet-R	✗	✗	✓	Classif.	(Hendrycks et al., 2021a)
ImageNet-Sk.	✗	✗	✓	Classif.	(Wang et al., 2019)
Food-101	✗	✓	✓	Classif.	(Bossard et al., 2014)
CIFAR-10	✗	✓	✓	Classif.	(Krizhevsky et al., 2009)
CIFAR-100	✗	✓	✓	Classif.	(Krizhevsky et al., 2009)
SUN397	✗	✓	✓	Classif.	(Xiao et al., 2010)
StanfordCars	✗	✓	✓	Classif.	(Krause et al., 2013)
FGVC-Aircraft	✗	✓	✓	Classif.	(Maji et al., 2013)
VOC 2007	✗	✓	✓	Classif.	(Everingham et al., 2010)
DTD	✗	✓	✓	Classif.	(Cimpoi et al., 2014)
Oxford Pets	✗	✓	✓	Classif.	(Parkhi et al., 2012)
Caltech101	✗	✓	✓	Classif.	(Fei-Fei et al., 2004)
Flowers	✗	✓	✓	Classif.	(Nilsback & Zisserman, 2008)
CUB200	✗	✓	✓	Classif.	(Welinder et al., 2010)
iNaturalist 2018	✗	✗	✓	Classif.	(Van Horn et al., 2018)
iNaturalist 2021	✗	✗	✓	Classif.	(Van Horn et al., 2021)
Places-205	✗	✗	✓	Classif.	(Zhou et al., 2014)
UCF101	✗	✗	✓	Video	(Soomro et al., 2012)
Kinetics-400	✗	✗	✓	Video	(Kay et al., 2017)
SSv2	✗	✗	✓	Video	(Goyal et al., 2017)
GLD v2	✓	✓	✗		(Weyand et al., 2020)
R-Paris	✗	✓	✓	Retrieval	(Radenović et al., 2018a)
R-Oxford	✗	✓	✓	Retrieval	(Radenović et al., 2018a)
Met	✗	✓	✓	Retrieval	(Ypsilantis et al., 2021)
Amstertime	✗	✓	✓	Retrieval	(Yıldız et al., 2022)
ADE20k	✗	✓	✓	Seg.	(Zhou et al., 2017)
Cityscapes	✗	✓	✓	Seg.	(Cordts et al., 2016)
VOC 2012	✗	✓	✓	Seg.	(Everingham et al., 2010)
Mapillary SLS	✓	✗	✗		(Warburg et al., 2020)
NYU-Depth V2	✗	✓	✓	Depth	(Silberman et al., 2012)
KITTI	✗	✓	✓	Depth	(Geiger et al., 2013)
SUN-RGBD	✗	✓	✓	Depth	(Song et al., 2015)
DollarStreet	✗	✗	✓	Fairness	(De Vries et al., 2019)
Casual Conv.	✗	✗	✓	Fairness	(Hazirbas et al., 2021)

Table 18: List of datasets used.

# 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 Santana Claire Roberts  
 Andrea Vedaldi Jamie Tolan John Brandt<sup>1</sup> Camille Couprie  
 Julien Mairal<sup>2</sup> Hervé Jégou Patrick Labatut Piotr Bojanowski  
*Meta AI Research* <sup>1</sup>*WRI* <sup>2</sup>*Inria*

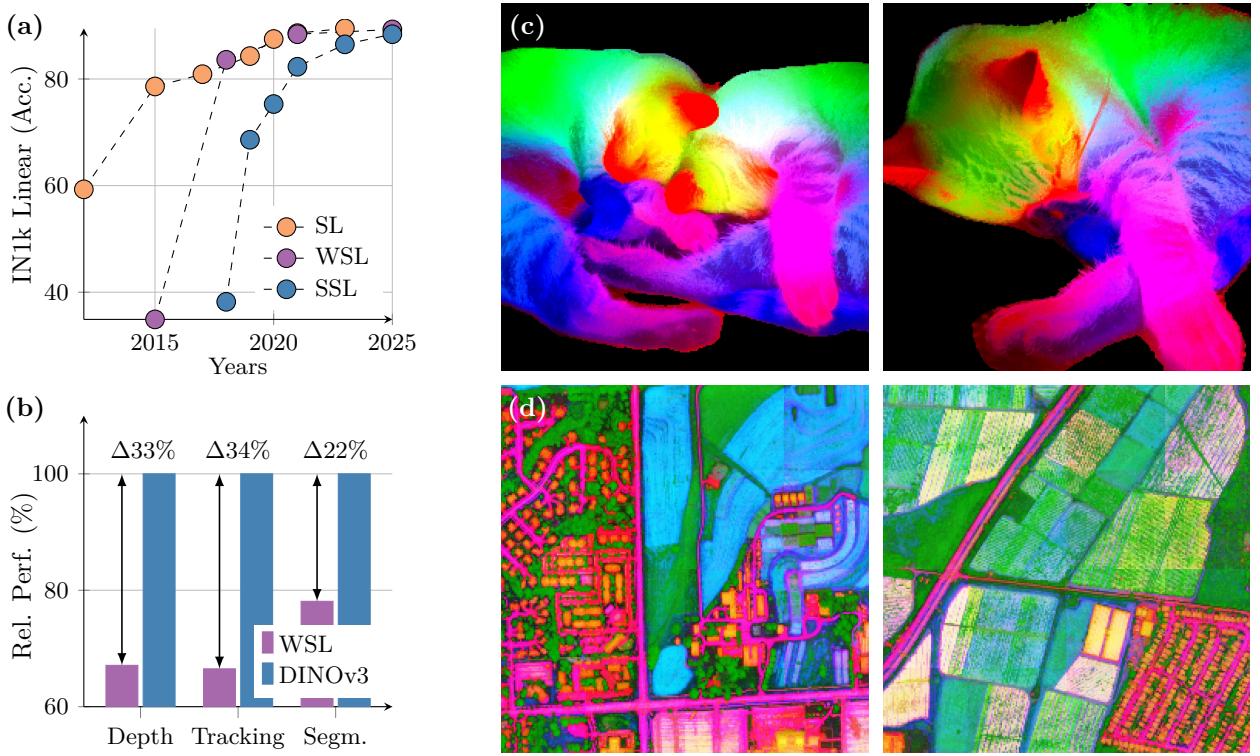
\*corresponding authors: *{osimeoni, huyvvo, seitzer, baldassarre, gas}@meta.com*

## Abstract

Self-supervised learning holds the promise of eliminating the need for manual data annotation, enabling models to scale effortlessly to massive datasets and larger architectures. By not being tailored to specific tasks or domains, this training paradigm has the potential to learn visual representations from diverse sources, ranging from natural to aerial images—using a single algorithm. This technical report introduces DINOv3, a major milestone toward realizing this vision by leveraging simple yet effective strategies. First, we leverage the benefit of scaling both dataset and model size by careful data preparation, design, and optimization. Second, we introduce a new method called Gram anchoring, which effectively addresses the known yet unsolved issue of dense feature maps degrading during long training schedules. Finally, we apply post-hoc strategies that further enhance our models’ flexibility with respect to resolution, model size, and alignment with text. As a result, we present a versatile vision foundation model that outperforms the specialized state of the art across a broad range of settings, without fine-tuning. DINOv3 produces high-quality dense features that achieve outstanding performance on various vision tasks, significantly surpassing previous self- and weakly-supervised foundation models. We also share the DINOv3 suite of vision models, designed to advance the state of the art on a wide spectrum of tasks and data by providing scalable solutions for diverse resource constraints and deployment scenarios.

## 1 Introduction

Foundation models have become a central building block in modern computer vision, enabling broad generalization across tasks and domains through a single, reusable model. Self-supervised learning (SSL) is a powerful approach for training such models, by learning directly from raw pixel data and leveraging the natural co-occurrences of patterns in images. Unlike weakly and fully supervised pretraining methods (Radford et al., 2021; Dehghani et al., 2023; Bolya et al., 2025) which require images paired with high-quality metadata, SSL unlocks training on massive, raw image collections. This is particularly effective for training large-scale visual encoders thanks to the availability of virtually unlimited training data. DINOv2 (Oquab et al., 2024) exemplifies these strengths, achieving impressive results in image understanding tasks (Wang et al., 2025) and enabling pre-training for complex domains such as histopathology (Chen et al., 2024). Models trained with SSL exhibit additional desirable properties: they are robust to input distribution shifts, provide strong global and local features, and generate rich embeddings that facilitate physical scene understanding. Since SSL models are not trained for any specific downstream task, they produce versatile and robust generalist features. For instance, DINOv2 models deliver strong performance across diverse tasks and domains without requiring task-specific finetuning, allowing a single frozen backbone to serve multiple purposes. Importantly, self-supervised learning is especially suitable to train on the vast amount of available observational data in

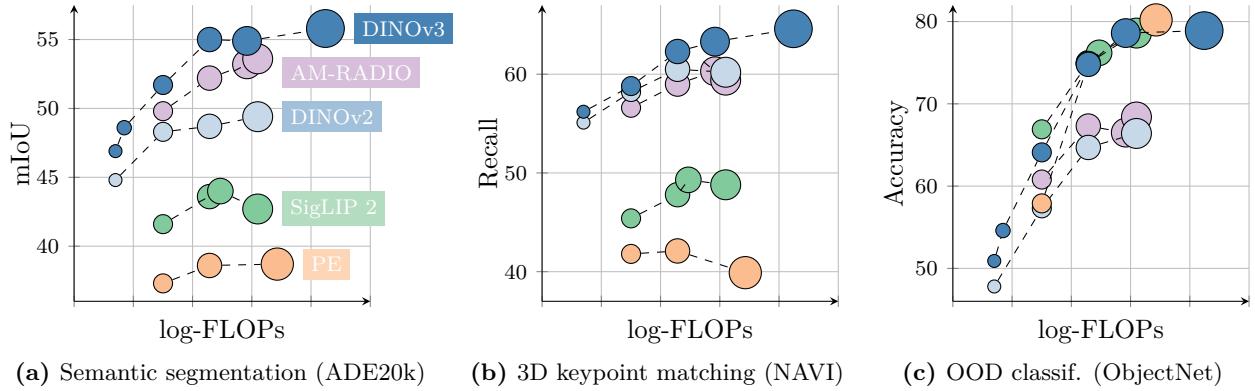


**Figure 1:** (a) Evolution of linear probing results on ImageNet1k (IN1k) over the years, comparing fully-supervised (SL), weakly-supervised (WSL) and self-supervised learning (SSL) methods. Despite coming into the picture later, SSL has quickly progressed and now reached the Imagenet accuracy plateau 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).

domains like histopathology (Vorontsov et al., 2024), biology (Kim et al., 2025), medical imaging (Pérez-García et al., 2025), remote sensing (Cong et al., 2022; Tolan et al., 2024), astronomy (Parker et al., 2024), or high-energy particle physics (Dillon et al., 2022). These domain often lack metadata and have already been shown to benefit from foundation models like DINOv2. Finally, SSL, requiring no human intervention, is well-suited for lifelong learning amid the growing volume of web data.

In practice, the promise of SSL, namely producing arbitrarily large and powerful models by leveraging large amounts of unconstrained data, remains challenging at scale. While model instabilities and collapse are mitigated by the heuristics proposed by Oquab et al. (2024), more problems emerge from scaling further. First, it is unclear how to collect useful data from unlabeled collections. Second, in usual training practice, employing cosine schedules implies knowing the optimization horizon a priori, which is difficult when training on large image corpora. Third, the performance of the features gradually decreases after early training, confirmed by visual inspection of the patch similarity maps. This phenomenon appears in longer training runs with models above ViT-Large size (300M parameters), reducing the usefulness of scaling DINOv2.

Addressing the problems above leads to this work, *DINOv3*, which advances SSL training at scale. We demonstrate that 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. Our research is guided by the following objectives: (1) training a foundational model versatile across tasks and domains, (2) improving the shortcomings of existing SSL models on dense features, (3) disseminating a family of models that can be used off-the-shelf. We discuss the three aims in the following.

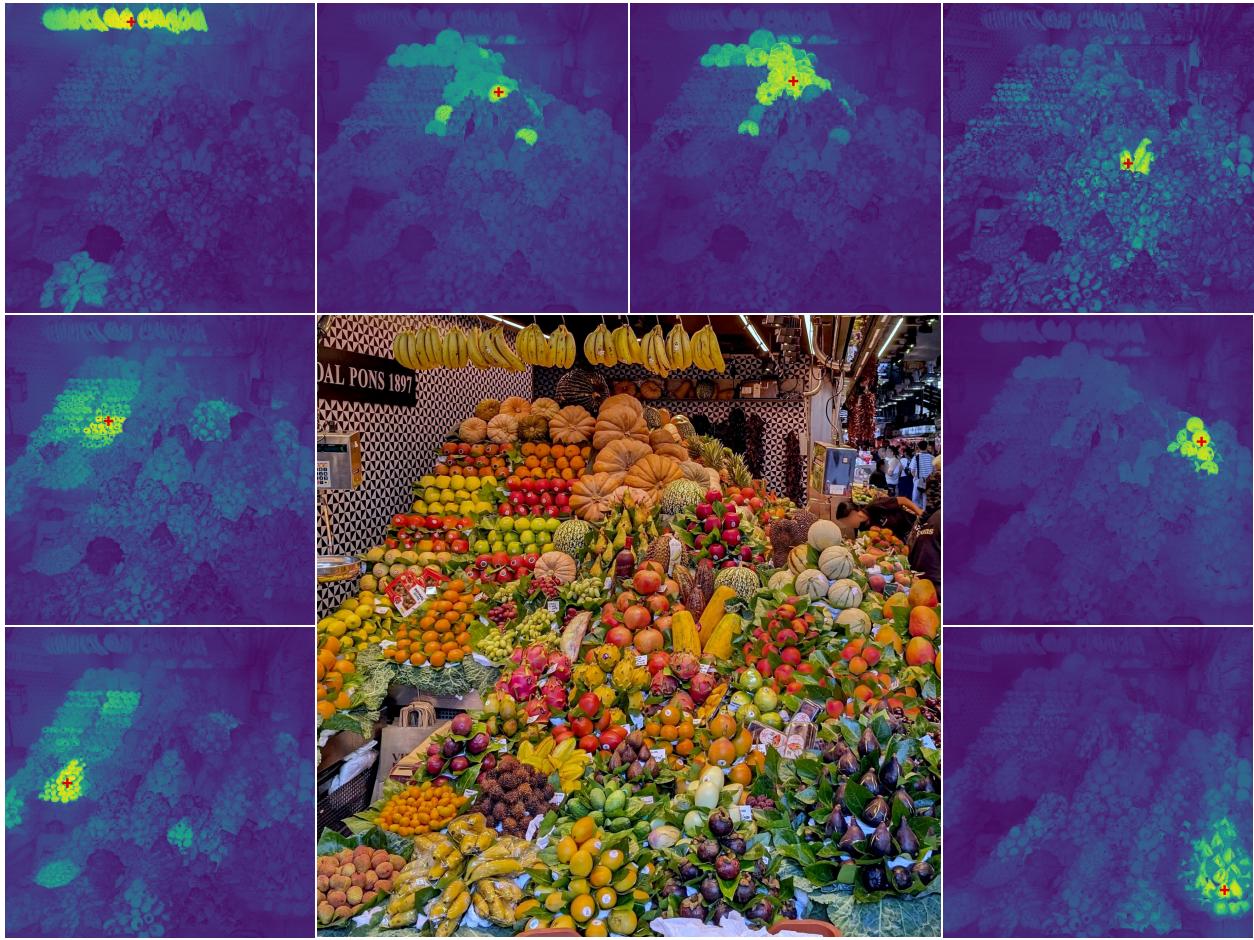


**Figure 2:** Performance of the DINOv3 family of models, compared to other families of self- or weakly-supervised models, on different benchmarks. DINOv3 significantly surpasses others on dense benchmarks, including models that leverage mask annotation priors such as AM-RADIO (Heinrich et al., 2025).

**Strong & Versatile Foundational Models** DINOv3 aims to offer a high level of versatility along two axes, which is enabled by the scaling of the model size and training data. First, a key desirable property for SSL models is to achieve excellent performance while being kept frozen, ideally reaching similar state-of-the-art results as specialized models. In that case, a single forward pass can deliver cutting-edge results across multiple tasks, leading to substantial computational savings—an essential advantage for practical applications, particularly on edge devices. We show the wide breadth of tasks that DINOv3 can successfully be applied to in Sec. 6. Second, a scalable SSL training pipeline that does not depend on metadata unlocks numerous scientific applications. By pre-training on a diverse set of images, whether web images or observational data, SSL models generalize across a large set of domains and tasks. As illustrated in Fig. 1(d), the PCA of DINOv3 features extracted from a high-resolution aerial image clearly allows to separates roads, houses, and greenery, highlighting the model’s feature quality.

**Superior Feature Maps Through Gram Anchoring** Another key feature of DINOv3 is a significant improvement of its dense feature maps. The DINOv3 SSL training strategy aims at producing models excelling at high-level semantic tasks while producing excellent feature maps amenable to solving geometric tasks such as depth estimation, or 3D matching. In particular, the models should produce dense features that can be used off-the-shelf or with little post-processing. The compromise between dense and global representation is especially difficult to optimize when training with vast amounts of images, since the objective of high-level understanding can conflict with the quality of the dense feature maps. These contradictory objectives lead to a collapse of dense features with large models and long training schedules. Our new Gram anchoring strategy effectively mitigates this collapse (see Sec. 4). As a result, DINOv3 obtains significantly better dense feature maps than DINOv2, staying clean even at high resolutions (see Fig. 3).

**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.



**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  $4096 \times 4096$ . *Please zoom in, do you agree with DINOv3?*

**Overview of Contributions** In this work, we introduce multiple contributions to address the challenge of scaling SSL towards a large frontier model. We build upon recent advances in automatic data curation (Vo et al., 2024) to obtain a large “background” training dataset that we carefully mix with a bit of specialized data (ImageNet-1k). This allows leveraging large amounts of unconstrained data to improve the model performance. This contribution (i) around data scaling will be described in Sec. 3.1.

We increase our main model size to 7B parameters by defining a custom variant of the ViT architecture. We include modern position embeddings (axial RoPE) and develop a regularization technique to avoid positional artifacts. Departing from the multiple cosine schedules in DINOv2, we train with constant hyperparameter schedules for 1M iterations. This allows producing models with stronger performance. This contribution (ii) on model architecture and training will be described in Sec. 3.2.

With the above techniques, we are able to train a model following the DINOv2 algorithm at scale. However, as mentioned previously, scale leads to a degradation of dense features. To address this, we propose a core improvement of the pipeline with a Gram anchoring training phase. This cleans the noise in the feature maps, leading to impressive similarity maps, and drastically improving the performance on both parametric and non-parametric dense tasks. This contribution (iii) on Gram training will be described in Sec. 4.

Following previous practice, the last steps of our pipeline consist of a high-resolution post-training phase and distillation into a series of high-performance models of various sizes. For the latter, we develop a novel and

---

efficient single-teacher multiple-students distillation procedure. This contribution **(iv)** transfers the power of our 7B frontier model to a family of smaller practical models for common usage, that we describe in [Sec. 5.2](#).

As measured in our thorough benchmarking, results in [Sec. 6](#) show that our approach defines a new standard in dense tasks and performs comparably to CLIP derivatives on global tasks. In particular, *with a frozen vision backbone*, we achieve state-of-the-art performance on longstanding computer vision problems such as object detection (COCO detection, mAP 66.1) and image segmentation (ADE20k, mIoU 63.0), outperforming specialized fine-tuned pipelines. Moreover, we provide evidence of the generality of our approach across domains by applying the DINOv3 algorithm to satellite imagery, in [Sec. 8](#), surpassing all prior approaches.

## 2 Related Work

**Self-Supervised Learning** Learning without annotations requires an artificial learning task that provides supervision in lieu for training. The art and challenge of SSL lies in carefully designing these so-called pre-text tasks in order to learn powerful representations for downstream tasks. The language domain, by its discrete nature, offers straightforward ways to set up such tasks, which led to many successful unsupervised pre-training approaches for text data. Examples include word embeddings ([Mikolov et al., 2013](#); [Bojanowski et al., 2017](#)), sentence representations ([Devlin et al., 2018](#); [Liu et al., 2019](#)), and plain language models ([Mikolov et al., 2010](#); [Zaremba et al., 2014](#)). In contrast, computer vision presents greater challenges due to the continuous nature of the signal. Early attempts mimicking language approaches extracted supervisory signals from parts of an image to predict other parts, *e.g.* by predicting relative patch position ([Doersch et al., 2015](#)), patch re-ordering ([Noroozi and Favaro, 2016](#); [Misra and Maaten, 2020](#)), or inpainting ([Pathak et al., 2016](#)). Other tasks involve re-colorizing images ([Zhang et al., 2016](#)) or predicting image transformations ([Gidaris et al., 2018](#)).

Among these tasks, *inpainting-based* approaches have gathered significant interest thanks to the flexibility of the patch-based ViT architecture ([He et al., 2021](#); [Bao et al., 2021](#); [El-Nouby et al., 2021](#)). The objective is to reconstruct corrupted regions of an image, which can be viewed as a form of denoising auto-encoding and is conceptually related to the masked token prediction task in BERT pretraining ([Devlin et al., 2018](#)). Notably, [He et al. \(2021\)](#) demonstrated that pixel-based masked auto-encoders (MAE) can be used as strong initializations for finetuning on downstream tasks. In the following, [Baevski et al. \(2022; 2023\)](#); [Assran et al. \(2023\)](#) showed that predicting a *learned latent space* instead of the pixel space leads to more powerful, higher-level features—a learning paradigm called JEPAs: “Joint-Embedding Predictive Architecture” ([LeCun, 2022](#)). Recently, JEPAs have also been extended to video training ([Bardes et al., 2024](#); [Assran et al., 2025](#)).

A second line of work, closer to ours, leverages *discriminative signals between images* to learn visual representations. This family of methods traces its origins to early deep learning research ([Hadsell et al., 2006](#)), but gained popularity with the introduction of instance classification techniques ([Dosovitskiy et al., 2016](#); [Bojanowski and Joulin, 2017](#); [Wu et al., 2018](#)). Subsequent advancements introduced contrastive objectives and information-theoretic criteria ([Hénaff et al., 2019](#); [He et al., 2020](#); [Chen and He, 2020](#); [Chen et al., 2020a](#); [Grill et al., 2020](#); [Bardes et al., 2021](#)), as well as self clustering-based strategies ([Caron et al., 2018](#); [Asano et al., 2020](#); [Caron et al., 2020; 2021](#)). More recent approaches, such as iBOT ([Zhou et al., 2021](#)), combine these discriminative losses with masked reconstruction objectives. All of these methods show the ability to learn strong features and achieve high performance on standard benchmarks like ImageNet ([Russakovsky et al., 2015](#)). However, most face challenges scaling to larger model sizes ([Chen et al., 2021](#)).

**Vision Foundation Models** The deep learning revolution began with the AlexNet breakthrough ([Krizhevsky et al., 2012](#)), a deep convolutional neural network that outperformed all previous methods on the ImageNet challenge ([Deng et al., 2009](#); [Russakovsky et al., 2015](#)). Already early on, features learned end-to-end on the large manually-labeled ImageNet dataset were found to be highly effective for a wide range of transfer learning tasks ([Oquab et al., 2014](#)). Early work on vision *foundation models* then focused on architecture development, including VGG ([Simonyan and Zisserman, 2015](#)), GoogleNet ([Szegedy et al., 2015](#)), and ResNets ([He et al., 2016](#)).

Given the effectiveness of *scaling*, subsequent works explored training larger models on big datasets. [Sun et al. \(2017\)](#) expanded supervised training data with the proprietary JFT dataset containing 300 million

---

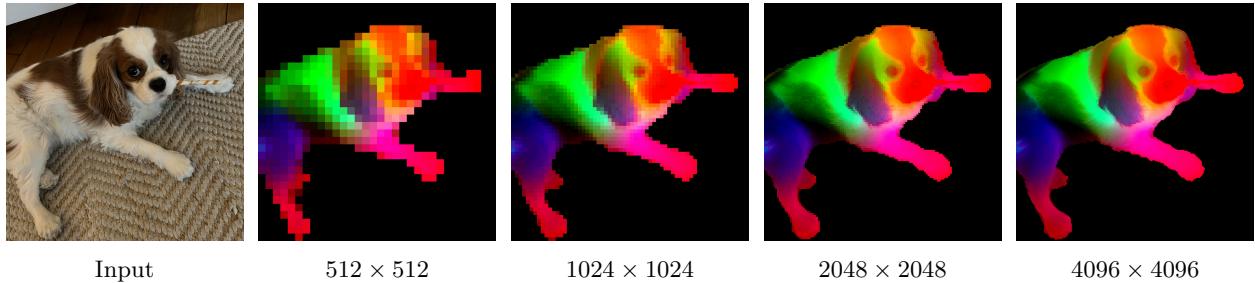
labeled images, showing impressive results. JFT also enabled significant performance gains for [Kolesnikov et al. \(2020\)](#). In parallel, scaling was explored using a combination of supervised and unsupervised data. For instance, an ImageNet-supervised model can be used to produce pseudo-labels for unsupervised data, which then serve to train larger networks ([Yalniz et al., 2019](#)). Subsequently, the availability of large supervised datasets such as JFT also facilitated the adaptation of the transformer architecture to computer vision ([Dosovitskiy et al., 2020](#)). In particular, achieving performance comparable to that of the original vision transformer (ViT) without access to JFT requires substantial effort ([Touvron et al., 2020; 2022](#)). Due to the learning capacity of ViTs, scaling efforts were further extended by [Zhai et al. \(2022a\)](#), culminating in the very large ViT-22B encoder ([Dehghani et al., 2023](#)).

Given the complexity of manually labeling large datasets, *weakly-supervised training*—where annotations are derived from metadata associated with images—provides an effective alternative to supervised training. Early on, [Joulin et al. \(2016\)](#) demonstrated that a network can be pre-trained by simply predicting all words in the image caption as targets. This initial approach was further refined by leveraging sentence structures ([Li et al., 2017](#)), incorporating other types of metadata and involve curation ([Mahajan et al., 2018](#)), and scaling ([Singh et al., 2022](#)). However, weakly-supervised algorithms only reached their full potential with the introduction of contrastive losses and the joint-training of caption representations, as exemplified by Align ([Jia et al., 2021](#)) and CLIP ([Radford et al., 2021](#)).

This highly successful approach inspired numerous *open-source reproductions and scaling efforts*. Open-CLIP ([Cherti et al., 2023](#)) was the first open-source effort to replicate CLIP by training on the LAION dataset ([Schuhmann et al., 2021](#)); following works leverage pre-trained backbones by fine-tuning them in a CLIP-style manner ([Sun et al., 2023; 2024](#)). Recognizing that data collection is a critical factor in the success of CLIP training, MetaCLIP ([Xu et al., 2024](#)) precisely follows the original CLIP procedure to reproduce its results, whereas [Fang et al. \(2024a\)](#) use supervised datasets to curate pretraining data. Other works focus on improving the training loss, *e.g.* using a sigmoid loss in SigLIP ([Zhai et al., 2023](#)), or leveraging a pre-trained image encoder ([Zhai et al., 2022b](#)). Ultimately though, the most critical components for obtaining cutting-edge foundation models are abundant high-quality data and substantial compute resources. In this vein, SigLIP 2 ([Tschanen et al., 2025](#)) and Perception Encoder (PE) ([Bolya et al., 2025](#)) achieve impressive results after training on more than 40B image-text pairs. The largest PE model is trained on 86B billion samples with a global batch size of 131K. Finally, a range of more complex and natively multimodal approaches have been proposed; these include contrastive captioning ([Yu et al., 2022](#)), masked modeling in the latent space ([Bao et al., 2021; Wang et al., 2022b; Fang et al., 2023; Wang et al., 2023a](#)), and auto-regressive training ([Fini et al., 2024](#)).

In contrast, relatively little work has focused on *scaling unsupervised image pretraining*. Early efforts include [Caron et al. \(2019\)](#) and [Goyal et al. \(2019\)](#) utilizing the YFCC dataset ([Thomee et al., 2016](#)). Further progress has been achieved by focusing on larger datasets and models ([Goyal et al., 2021; 2022a](#)), as well as initial attempts at data curation for SSL ([Tian et al., 2021](#)). Careful tuning of the training algorithms, larger architectures, and more extensive training data lead to the impressive results of DINOv2 ([Oquab et al., 2024](#)); for the first time, an SSL model matched or surpassed open-source CLIP variants on a range of tasks. This direction has recently been further pushed by [Fan et al. \(2025\)](#) by scaling to larger models without data curation, or by [Venkataramanan et al. \(2025\)](#) using open datasets and improved training recipes.

**Dense Transformer Features** A broad range of modern vision applications consume *dense features* of pre-trained transformers, including multi-modal models ([Liu et al., 2023; Beyer et al., 2024](#)), generative models ([Yu et al., 2025; Yao et al., 2025](#)), 3D understanding ([Wang et al., 2025](#)), video understanding ([Lin et al., 2023a; Wang et al., 2024b](#)), and robotics ([Driess et al., 2023; Kim et al., 2024](#)). On top of that, traditional vision tasks such as detection, segmentation, or depth estimation require accurate local descriptors. To enhance the quality of SSL-trained local descriptors, a substantial body of work focuses on developing *local SSL losses*. Examples include leveraging spatio-temporal consistency in videos, *e.g.* using point track loops as training signal ([Jabri et al., 2020](#)), exploiting the spatial alignment between different crops of the same image ([Pinheiro et al., 2020; Bardes et al., 2022](#)), or enforcing consistency between neighboring patches ([Yun et al., 2022](#)). [Dariset et al. \(2025\)](#) show that predicting clustered local patches leads to improved dense representations. DetCon ([Hénaff et al., 2021](#)) and ORL ([Xie et al., 2021](#)) perform contrastive learning on



**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.

region proposals but assume that such proposals exist *a priori*; this assumption is relaxed by approaches such as ODIN (Hénaff et al., 2022) and SlotCon (Wen et al., 2022). Without changing the training objective, Dariset et al. (2024) show that adding register tokens to the input sequence greatly improves dense feature maps, and recent works find this can be done without model training (Jiang et al., 2025; Chen et al., 2025).

A recent trend are distillation-based, “*agglomerative*” methods that combine information from multiple image encoders with varying in global and local feature quality, trained using different levels of supervision (Ranzinger et al., 2024; Bolya et al., 2025): AM-RADIO (Ranzinger et al., 2024) combines the strengths of the fully-supervised SAM (Kirillov et al., 2023), the weakly-supervised CLIP, and the self-supervised DINOv2 into a unified backbone. The Perception Encoder (Bolya et al., 2025) similarly distills SAM(v2) into a specialized dense variant called PEspatial. They use an objective enforcing cosine similarity between student and teacher patches to be high, where their teacher is trained with mask annotations. Similar losses were shown to be effective in the context of style transfer, by reducing the inconsistency between the Gram matrices of feature dimensions (Gatys et al., 2016; Johnson et al., 2016; Yoo et al., 2024). In this work, we adopt a Gram objective to regularize cosine similarity between student and teacher patches, favoring them being close. In our case, we use earlier iterations of the SSL model itself as the teacher, demonstrating that early-stage SSL models effectively guides SSL training for both global and dense tasks.

Other works focus on post-hoc improvements to the local features of SSL-trained models. For example, Ziegler and Asano (2022) fine-tune a pre-trained model with a dense clustering objective; similarly, Salehi et al. (2023) fine-tune by aligning patch features temporally, in both cases enhance the quality of local features. Closer to us, Pariza et al. (2025) propose a patch-sorting based objective to encourage the student and teacher to produce features with consistent neighbor ordering. Without finetuning, STEGO (Hamilton et al., 2022) learns a non-linear projection on top of frozen SSL features to form compact clusters and amplify correlation patterns. Alternatively, Simoncini et al. (2024) augment self-supervised features by concatenating gradients from different self-supervised objectives to frozen SSL features. Recently, Wysoczańska et al. (2024) show that noisy feature maps are significantly improved through a weighted average of patches.

Related, but not specific to SSL, some recent works generate high-resolution feature maps from ViT feature maps (Fu et al., 2024), which are often low-resolution due to patchification of images. In contrast with this body of work, our models natively deliver high-quality dense feature maps that remain stable and consistent across resolutions, as shown in Fig. 4.

### 3 Training at Scale Without Supervision

DINOv3 is a next-generation model designed to produce the most robust and flexible visual representations to date by pushing the boundaries of self-supervised learning. We draw inspiration from the success of large language models (LLMs), for which scaling-up the model capacity leads to outstanding *emerging properties*. By leveraging models and training datasets that are an order of magnitude larger, we seek to unlock the full potential of SSL and drive a similar paradigm shift for computer vision, unencumbered by the limitations

**Table 1:** Influence of training data on features quality shown via performance on downstream tasks. We compare datasets curated with *clustering* (Vo et al., 2024) and *retrieval* (Oquab et al., 2024) to *raw* data and to our data mixture. This ablation study is run for a shorter schedule of 200k iterations.

Dataset	IN1k k-NN	IN1k Linear	ObjectNet	iNaturalist 2021	Paris Retrieval
Raw	80.1	84.8	70.3	70.1	63.3
Clustering	79.4	85.4	72.3	81.3	85.2
Retrieval	84.0	86.7	70.7	86.0	82.7
LVD-1689M (ours)	<b>84.6</b>	<b>87.2</b>	<b>72.8</b>	<b>87.0</b>	<b>85.9</b>

inherent to traditional supervised or task-specific approaches. In particular, SSL produces rich, high-quality visual features that are not biased toward any specific supervision or task, thereby providing a versatile foundation for a wide range of downstream applications. While previous attempts at scaling SSL models have been hindered by issues of instability, this section describes how we harness the benefits of scaling with careful data preparation, design, and optimization. We first describe the dataset creation procedure (Sec. 3.1), then present the self-supervised SSL recipe used for this first training phase of DINOv3 (Sec. 3.2). This includes the choice of architecture, loss functions, and optimization techniques. The second training phase, focusing on dense features, will be described in Sec. 4.

### 3.1 Data Preparation

Data scaling is one of the driving factors behind the success of large foundation models (Touvron et al., 2023; Radford et al., 2021; Xu et al., 2024; Oquab et al., 2024). However, increasing naively the size of the training data does not necessarily translate into higher model quality and better performance on downstream benchmarks (Goyal et al., 2021; Oquab et al., 2024; Vo et al., 2024): Successful data scaling efforts typically involve careful data curation pipelines. These algorithms may have different objectives: either focusing on improving data *diversity* and *balance*, or data *usefulness*—its relevance to common practical applications. For the development of DINOv3, we combine two complementary approaches to improve both the generalizability and performance of the model, striking a balance between the two objectives.

**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).

**Data Sampling** During pre-training, we use a sampler to mix different data parts together. There are several different options for mixing the above data components. One is to train with *homogeneous* batches of data that come from a single, randomly selected component in each iteration. Alternatively, we can optimize the model on *heterogeneous* batches that are assembled by data from all components, selected using certain ratios. Inspired by Charton and Kempe (2024), who observed that it is beneficial to have homogeneous batches consisting of very high quality data from a small dataset, we randomly sample in each iteration

**Table 2:** Comparison of the teacher architectures used in DINOv2 and DINOv3 models. We keep the model 40 blocks deep, and increase the embedding dimension to 4096. Importantly, we use a patch size of 16 pixels, changing the effective sequence length for a given resolution.

Teacher model	DINOv2	DINOv3
Backbone	ViT-giant	ViT-7B
#Params	1.1B	6.7B
#Blocks	40	40
Patch Size	14	16
Pos. Embeddings	Learnable	RoPE
Registers	4	4
Embed. Dim.	1536	4096
FFN Type	SwiGLU	SwiGLU
FFN Hidden Dim.	4096	8192
Attn. Heads	24	32
Attn. Heads Dim.	64	128
DINO Head MLP	4096-4096-256	8192-8192-512
DINO Prototypes	128k	256k
iBOT Head MLP	4096-4096-256	8192-8192-384
iBOT Prototypes	128k	96k

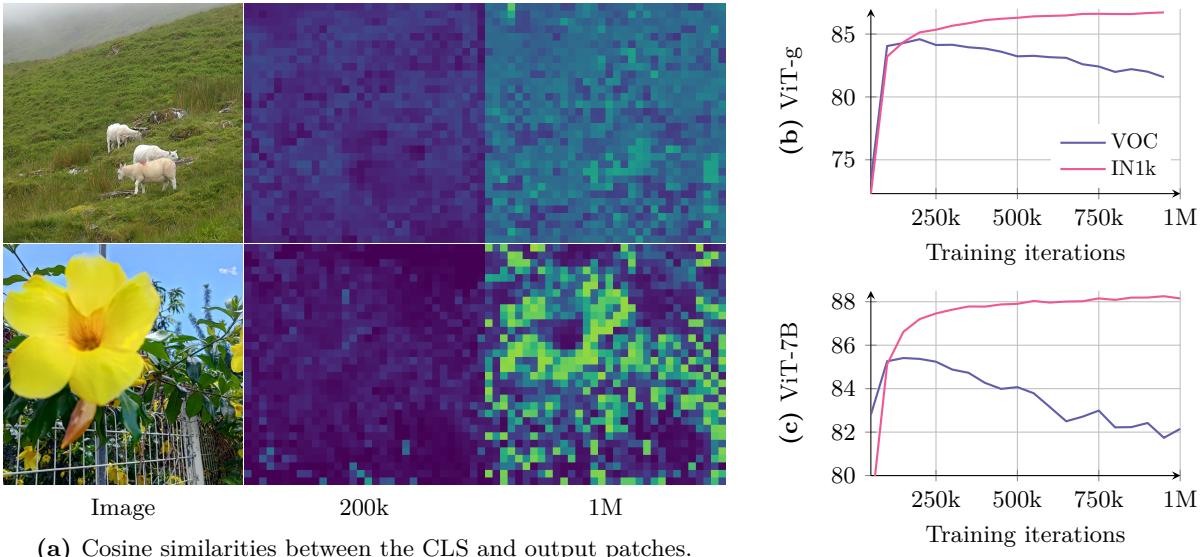
either a homogeneous batch from ImageNet1k alone or a heterogeneous batch mixing data from all other components. In our training, homogeneous batches from ImageNet1k account for 10% of training.

**Data Ablation** To assess the impact of our data curation technique, we perform an ablation study to compare our data mix against datasets curated with clustering or retrieval-based methods alone, and the raw data pool. To this end, we train a model on each dataset and compare their performance on standard downstream tasks. For efficiency, we use a shorter schedule of 200k iterations instead of 1M iterations. In Tab. 1, it can be seen that no single curation technique works best across all benchmarks, and that our full pipeline allows us to obtain the best of both worlds.

### 3.2 Large-Scale Training with Self-Supervision

While models trained with SSL have demonstrated interesting properties (Chen et al., 2020b; Caron et al., 2021), most SSL algorithms have not been scaled-up to larger models sizes. This is either due to issues with training stability (Dariset et al., 2025), or overly simplistic solutions that fail to capture the full complexity of the visual world. When trained at scale (Goyal et al., 2022a), models trained with SSL do not necessarily show impressive performance. One notable exception is DINOv2, a model with 1.1 billion parameters trained on curated data, matching the performance of weakly-supervised models like CLIP (Radford et al., 2021). A recent effort to scale DINOv2 to 7 billion parameters (Fan et al., 2025) demonstrates promising results on global tasks, but with disappointing results on dense prediction. Here, we aim to scale up the model and data, and obtain even more powerful visual representations with both improved global and local properties.

**Learning Objective** We train the model with a discriminative self-supervised strategy which is a mix of several self-supervised objectives with both global and local loss terms. Following DINOv2 (Oquab et al., 2024), we use an image-level objective (Caron et al., 2021)  $\mathcal{L}_{\text{DINO}}$ , and balance it with a patch-level latent reconstruction objective (Zhou et al., 2021)  $\mathcal{L}_{\text{iBOT}}$ . We also replace the centering from DINO with the Sinkhorn-Knopp from SwAV (Caron et al., 2020) in both objectives. Each objective is computed using the output of a dedicated head on top of the backbone network, allowing for some specialization of features before the computation of the losses. Additionally, we use a dedicated layer normalization applied to the backbone outputs of the local and global crops. Empirically, we found this change to stabilize ImageNet kNN-classification late in training (+0.2 accuracy) and improve dense performance (e.g. +1 mIoU on ADE20k segmentation, -0.02 RMSE on NYUv2 depth estimation). In addition, a Koleo regularizer  $\mathcal{L}_{\text{Koleo}}$  is added to encourage the features within a batch to spread uniformly in the space (Sablayrolles et al., 2018). We use



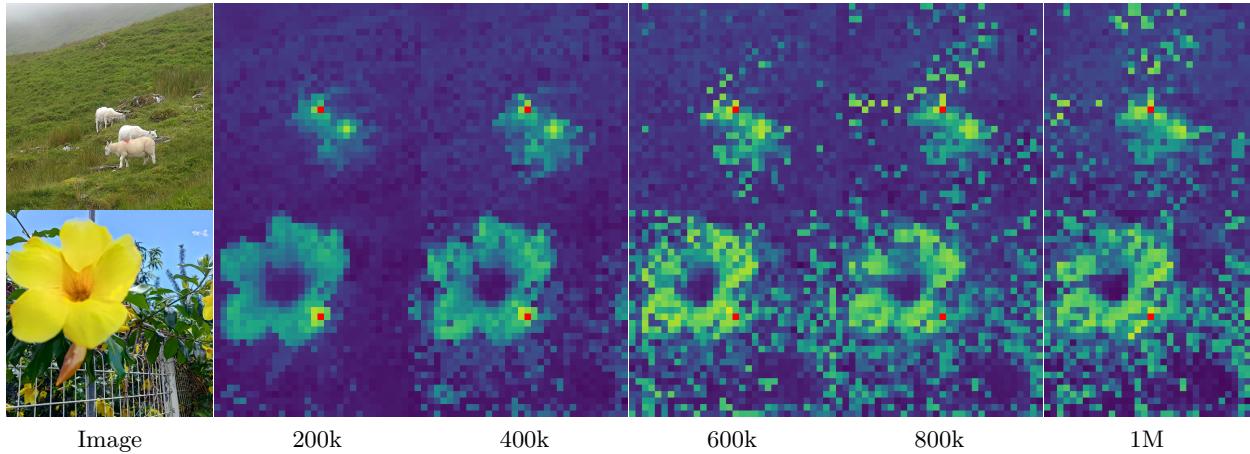
**Figure 5:** Evolution of the cosine similarities (a) and of the accuracy on ImageNet1k linear (IN1k) and segmentation on VOC for ViT-g (b) and ViT-7B (c). We observe that the segmentation performance is maximal when the cosine similarities between the patch tokens and the class tokens are low. As training progresses, these similarities increase and the performance on dense tasks decreases.

a distributed implementation of Koleo in which the loss is applied in small batches of 16 samples—possibly across GPUs. Our initial training phase is carried by optimizing the following loss:

$$\mathcal{L}_{\text{Pre}} = \mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + 0.1 * \mathcal{L}_{\text{DKoleo}}. \quad (1)$$

**Updated Model Architecture** For the model scaling aspect of this work, we increase the size of the model to 7B parameters, and provide in Tab. 2 a comparison of the corresponding hyperparameters with the 1.1B parameter model trained in the DINOv2 work. We also employ a custom variant of RoPE: our base implementation assigns coordinates in a normalized  $[-1, 1]$  box to each patch, then applies a bias in the multi-head attention operation depending on the relative position of two patches. In order to improve the robustness of the model to resolutions, scales and aspect ratios, we employ *RoPE-box jittering*. The coordinate box  $[-1, 1]$  is randomly scaled to  $[-s, s]$ , where  $s \in [0.5, 2]$ . Together, these changes enable DINOv3 to better learn detailed and robust visual features, improving its performance and scalability.

**Optimization** Training large models on very large datasets represents a complicated experimental workflow. Because the interplay between model capacity and training data complexity is hard to assess *a priori*, it is impossible to guess the right optimization horizon. To overcome this, we get rid of all parameter scheduling, and train with constant learning rate, weight decay, and teacher EMA momentum. This has two main benefits. First, we can continue training as long as downstream performance continues to improve. Second, the number of optimization hyperparameters is reduced, making it easier to choose them properly. For the training to start properly, we still use a linear warmup for learning rate and teacher temperature. Following common practices, we use AdamW (Loshchilov and Hutter, 2017), and set the total batch size to 4096 images split across 256 GPUs. We train our models using the multi-crop strategy (Caron et al., 2020), taking 2 global crops and 8 local crops per image. We use square images with a side length of 256/112 pixels for global/local crops, which, along with the change in patch size, results in the same effective sequence length per image as in DINOv2 and a total sequence length of 3.7M tokens per batch. Additional hyperparameters can be found in App. C and in the code release.



**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.

## 4 Gram Anchoring: A Regularization for Dense Features

To fully leverage the benefits of large-scale training, we aim to train the 7B model for an extended duration, with the notion that it could potentially train indefinitely. As expected, prolonged training leads to improvements on global benchmarks. However, as training progresses, the performance degrades on dense tasks (Figs. 5b and 5c). This phenomenon, which is due to the emergence of patch-level inconsistencies in feature representations, undermines the interest behind extended training.<sup>1</sup> In this section, we first analyze the loss of patch-level consistency, then propose a new objective to mitigate it, called *Gram anchoring*. We finally discuss the impact of our approach on both training stability and model performance.

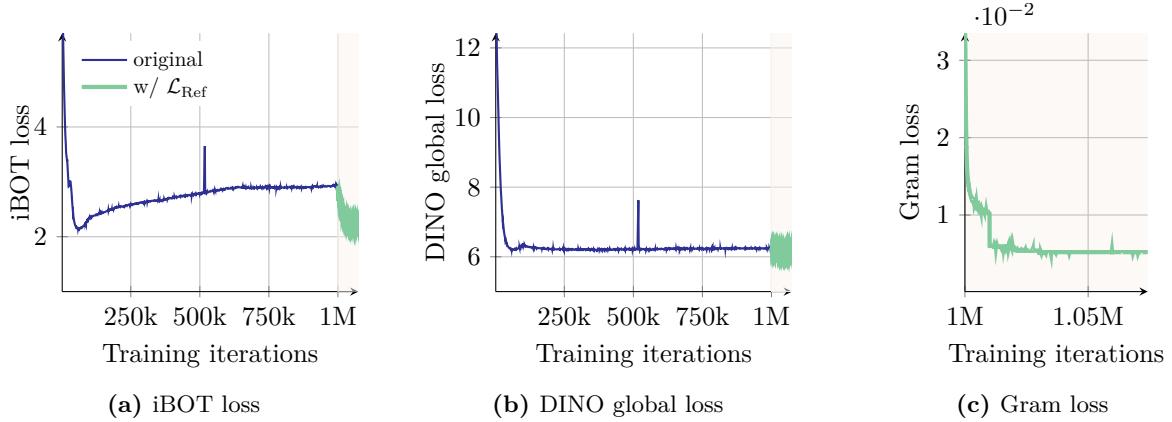
### 4.1 Loss of Patch-Level Consistency Over Training

During extended training, we observe consistent improvements in global metrics but a notable decline in performance on dense prediction tasks. This behavior was previously observed, to a lesser extent, during the training of DINOv2, and also discussed in the scaling effort of Fan et al. (2025). However, to the best of our knowledge, it remains unresolved to date. We illustrate the phenomenon in Figs. 5b and 5c, which present the performance of the model across iterations on both image classification and segmentation tasks. For classification, we train a linear classifier on ImageNet-1k using the CLS token and report top-1 accuracy. For segmentation, we train a linear layer on patch features extracted from Pascal VOC and report mean Intersection over Union (mIoU). We observe that both for the ViT-g and the ViT-7B, the classification accuracy monotonically improves throughout training. However, segmentation performance declines in both cases after approximately 200k iterations, falling below its early levels in the case of the ViT-7B.

To better understand this degradation, we analyze the quality of patch features by visualizing cosine similarities between patches. Fig. 6 shows the cosine similarity maps between the backbone’s output patch features and a reference patch (highlighted in red). At 200k iterations, the similarity maps are smooth and well-localized, indicating consistent patch-level representations. However, by 600k iterations and beyond, the maps degrade substantially, with an increasing number of irrelevant patches with high similarity to the reference patch. This loss of patch-level consistency correlates with the drop in dense task performance.

These patch-level irregularities differ from the high-norm patch outliers described in Dariset et al. (2024). Specifically, with the integration of register tokens, patch norms remain stable throughout training. However, we notice that the cosine similarity between the CLS token and the patch outputs gradually increases during training. This is expected, yet it means that the locality of the patch features diminishes. We visualize this phenomenon in Fig. 5a, which depicts the cosine maps at 200k and 1M iterations. In order to mitigate the

<sup>1</sup>We also observed different types of outliers appearing with continued training; we provide a discussion in App. A.



**Figure 7:** Evolution through the training iterations of the patch-level iBOT loss, the global loss DINO (applied to the global crops) and the newly introduced Gram loss. We highlight the iterations of the refinement step  $\mathcal{L}_{\text{Ref}}$  which uses the Gram objective.

drop on dense tasks, we propose a new objective specifically designed to regularize the patch features and ensure a good patch-level consistency, while preserving high global performance.

#### 4.2 Gram Anchoring Objective

Throughout our experiments, we have identified a relative independence between learning strong discriminative features and maintaining local consistency, as observed in the lack of correlation between global and dense performance. While combining the global DINO loss with the local iBOT loss has begun to address this issue, we observe that the balance is unstable, with global representation dominating as training progresses. Building on this insight, we propose a novel solution that explicitly leverages this independence.

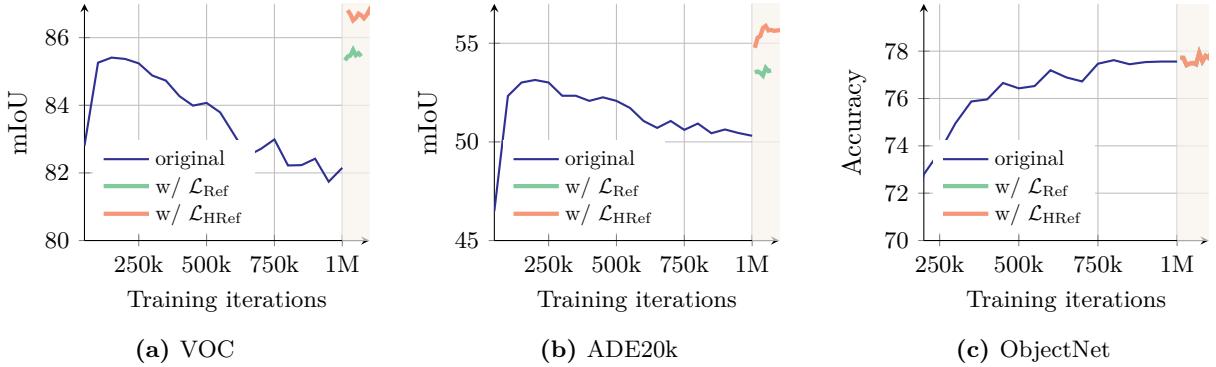
We introduce a new objective which mitigates the degradation of patch-level consistency by enforcing the quality of the patch-level consistency, without impacting the features themselves. This new loss function operates on the Gram matrix: the matrix of all pairwise dot products of patch features in 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:

$$\mathcal{L}_{\text{Gram}} = \|\mathbf{X}_S \cdot \mathbf{X}_S^\top - \mathbf{X}_G \cdot \mathbf{X}_G^\top\|_F^2. \quad (2)$$

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. We call this second step of training the *refinement step*, which optimizes the objective  $\mathcal{L}_{\text{Ref}}$ , with

$$\mathcal{L}_{\text{Ref}} = w_D \mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + w_{\text{DK}} \mathcal{L}_{\text{DKoleo}} + w_{\text{Gram}} \mathcal{L}_{\text{Gram}}. \quad (3)$$

We visualize the evolution of different losses in Fig. 7 and observe that applying the Gram objective significantly influences the iBOT loss, causing it to decrease more rapidly. This suggests that the stability introduced by the stable Gram teacher positively impacts the iBOT objective. In contrast, the Gram objective does not have a significant effect on the DINO losses. This observation implies that the Gram and iBOT objectives impact the features in a similar way, whereas the DINO losses affect them differently.



**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.

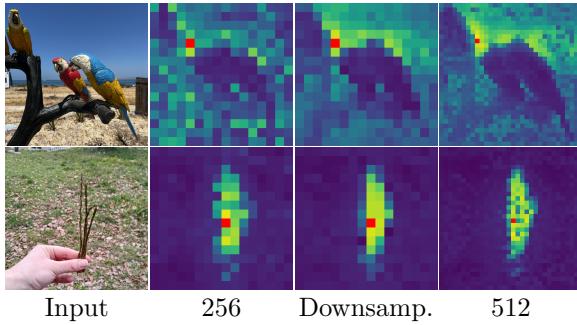
Regarding performance, we observe the impact of the new loss is almost immediate. As shown in Fig. 8, incorporating Gram anchoring leads to significant improvements on dense tasks within the first 10k iterations. We also see notable gains on the ADE20k benchmark following the Gram teacher updates. Additionally, longer training further benefits performance on the ObjectNet benchmark and other global benchmarks show mild impact from the new loss.

### 4.3 Leveraging Higher-Resolution Features

Recent work shows that a weighted average of patch features can yield stronger local representations by smoothing outlier patches and enhancing patch-level consistency (Wysoczańska et al., 2024). On the other hand, feeding higher-resolution images into the backbone produces finer and more detailed feature maps. We leverage the benefits of both observations to compute high-quality features for Gram teacher. Specifically, we first input images at twice the normal resolution into the Gram teacher, then  $2\times$  down-sample the resulting feature maps with the bicubic interpolation to achieve the desired smooth feature maps that match the size of the student output. Fig. 9a visualizes the Gram matrices of patch features obtained with images at resolutions 256 and 512, as well as those obtained after  $2\times$  down-sampling features from the 512-resolution (denoted as ‘downsamp.’). We observe that the superior patch-level consistency in the higher-resolution features is preserved through down-sampling, resulting in smoother and more coherent patch-level representations. As a side note, our model can seamlessly process images at varying resolutions without requiring adaptation, thanks to the adoption of Rotary Positional Embeddings (RoPE) introduced by Su et al. (2024).

We compute the Gram matrix of the down-sampled features and use it to replace  $\mathbf{X}_G$  in the objective  $\mathcal{L}_{\text{Gram}}$ . We note the new resulting refinement objective as  $\mathcal{L}_{\text{HRef}}$ . This approach enables the Gram objective to effectively distill the improved patch consistency of smoothed high-resolution features into the student model. As shown in Fig. 8 and Fig. 9b, this distillation translates into better predictions on dense tasks, yielding additional gains on top of the benefit brought by  $\mathcal{L}_{\text{Ref}}$  (+2 mIoU on ADE20k). We also ablate the choice of Gram teacher in Fig. 9b. Interestingly, choosing the Gram teacher from 100k or 200k does not significantly impact the results, but using a much later Gram teacher (1M iterations) is detrimental because the patch-level consistency of such a teacher is inferior.

Finally, we qualitatively illustrate the effect of Gram anchoring to patch-level consistency in Fig. 10 which visualizes the Gram matrices patch features obtained with the initial training and high-resolution Gram anchoring refinement. We observe great improvements in feature correlations that our high-resolution refinement procedure brings about.

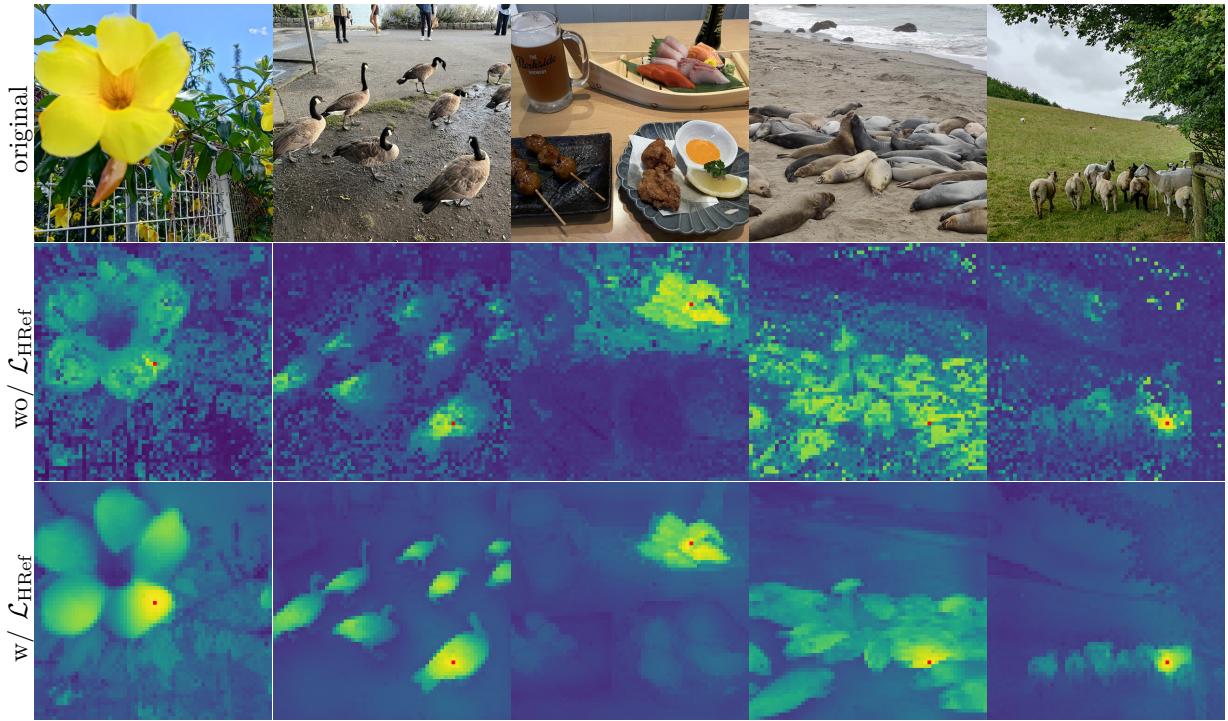


(a) Gram matrices at different input resolutions.

Method	Teacher Iteration	Res.	IN1k Linear	ADE mIoU	NYU RMSE
Baseline	—	—	<b>88.2</b>	50.3	0.307
GRAM	200k	$\times 1$	88.0	53.6	0.285
GRAM	200k	$\times 2$	88.0	<b>55.7</b>	<b>0.281</b>
GRAM	100k	$\times 2$	87.9	<b>55.7</b>	0.284
GRAM	1M	$\times 2$	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.



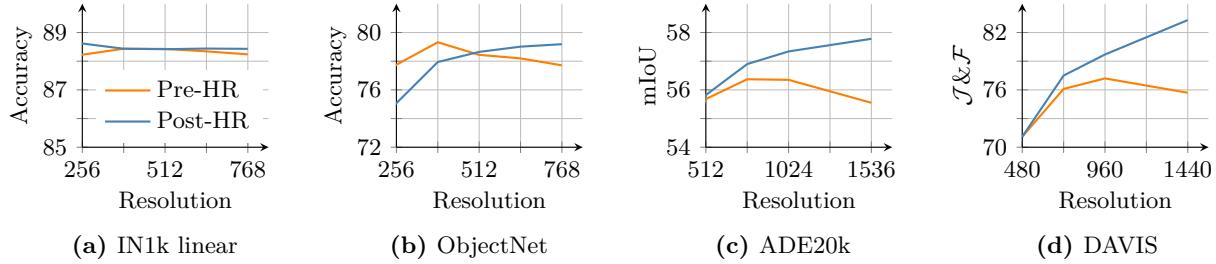
**Figure 10:** Qualitative effect of Gram anchoring. We visualize cosine maps before and after using the refinement objective  $\mathcal{L}_{HRef}$ . The input resolution of the images is  $1024 \times 1024$  pixels.

## 5 Post-Training

This section presents *post-training* stages. This includes a high-resolution adaptation phase enabling effective inference at different input resolutions (Sec. 5.1), model distillation producing quality and efficient smaller-sized models (Sec. 5.2), and text alignment adding zero-shot capabilities to DINOv3 (Sec. 5.3).

### 5.1 Resolution Scaling

We train our model at a relatively small resolution of 256, which gives us a good trade-off between speed and effectiveness. For a patch size of 16, this setup leads to the same input sequence length as DINOv2, which was trained with resolution 224 and patch size 14. However, many contemporary computer vision



**Figure 11:** Effect of high resolution adaptation. Results before (‘Pre-HR’) and after (‘Post-HR’) resolution scaling (Sec. 5.1) on (a) linear classification on ImageNet, (b) applied OOD to ObjectNet, (c) linear semantic segmentation on ADE20k, and (d) segmentation tracking on DAVIS at different evaluation resolutions.

applications require processing images at significantly higher resolutions, often  $512 \times 512$  pixels or greater, to capture intricate spatial information. The inference image resolution is also not fixed in practice and varies depending on specific use cases. To address this, we extend our training regime with a high-resolution adaptation step (Touvron et al., 2019). To ensure high performance across a range of resolutions, we utilize *mixed resolutions*, sampling differently-sized pairs of global and local crops per mini-batch. Specifically, we consider global crop sizes from  $\{512, 768\}$  and local crop sizes from  $\{112, 168, 224, 336\}$  and train the model for 10k additional iterations.

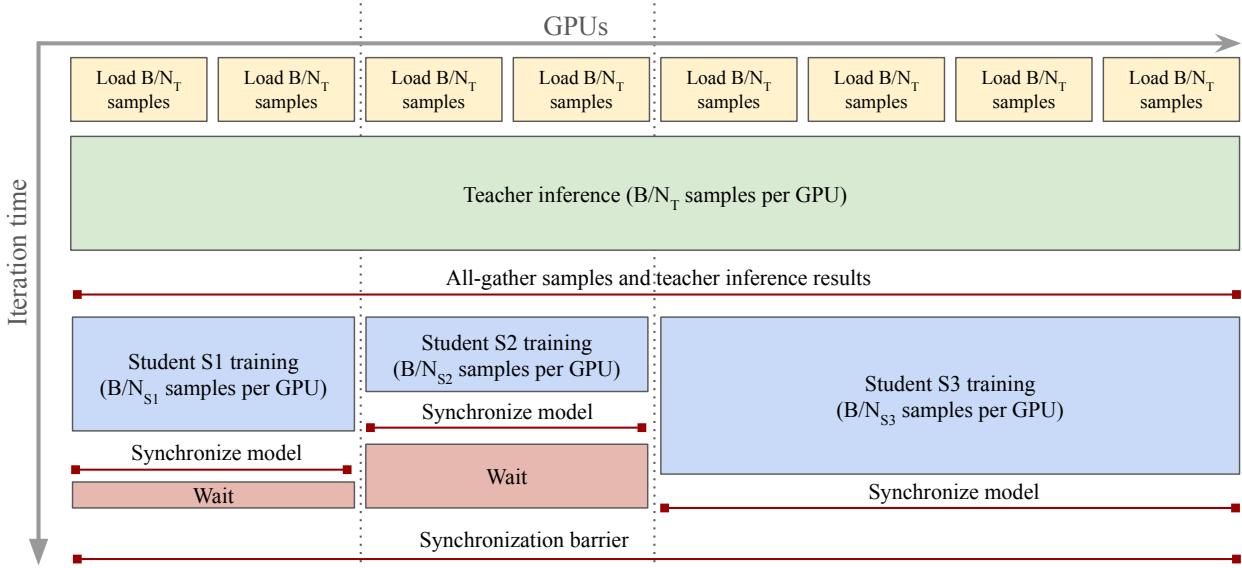
Similar to the main training, a key component of this high-resolution adaptation phase is the addition of Gram anchoring, using the 7B teacher as Gram teacher. We found this component to be essential: without it, the model performance on dense prediction tasks degrades significantly. The Gram anchoring encourages the model to maintain consistent and robust feature correlations across spatial locations, which is crucial when dealing with the increased complexity of high-resolution inputs.

Empirically, we observe that this relatively brief but targeted high-resolution step substantially enhances the overall model’s quality and allows it to generalize across a wide range of input sizes, as shown visually in Fig. 4. In Fig. 11, we compare our 7B model before and after adaptation. We find that resolution scaling leads to a small gain on ImageNet classification (a) with relatively stable performance w.r.t. resolution. However, in ObjectNet OOD transfer (b), we observe that the performance tends to degrade slightly for lower resolutions, while improving for higher resolutions. This is largely compensated by the improvement in the quality of local features at high resolution, shown by the positive trend in segmentation on ADE20k (c) and tracking on DAVIS (d). Adaptation leads to local features that *improve with image size*, leveraging the richer spatial information available at larger resolutions and effectively enabling high-resolution inference. Interestingly, the adapted model supports resolutions way beyond the maximum training resolution of 768—we visually observe stable feature maps at resolutions above 4k (c.f. Fig. 4).

## 5.2 Model Distillation

**A Family of Models for Multiple Use-Cases** We perform knowledge distillation of the ViT-7B model into smaller Vision Transformer variants (ViT-S, ViT-B, and ViT-L), which are highly valued by the community for their improved manageability and efficiency. Our distillation approach uses the same training objective as in the first training phase, ensuring consistency in learning signals. However, instead of relying on an exponential moving average (EMA) of model weights, we use the 7B model directly as the teacher to guide the smaller student models. In this case, the teacher model is fixed. We do not observe patch-level consistency issues and therefore do not apply the Gram anchoring technique. This strategy enables the distilled models to inherit the rich representational power of the large teacher while being more practical for deployment and experimentation.

Our ViT-7B model is distilled into a series of ViT models with sizes covering a broad range of compute budgets, and allowing proper comparison with concurrent models. They include the standard ViT-S (21M params), B (86M), L (0.3B), along with a custom ViT-S+ (29M) and a custom ViT-H+ (0.8B) model to close the performance gap with the self-distilled 7B teacher model. Indeed, we observe in DINOv2 that smaller



**Figure 12:** Multi-student distillation procedure. In this diagram, we distill 3 students in parallel: we first share teacher inference across all  $T$  nodes to save compute, and gather inputs and results on all GPUs. Then, smaller groups perform student training. We adjust the size of these groups such that the training step has the same duration across all students  $S_i$ , minimizing idle time waiting at the synchronization barrier.

student models can reach a performance on par with their teacher as the distillation. As a result, the distilled models deliver frontier-level performance for a fraction of the inference compute as we see in [Tab. 14](#). We train the models for 1M iterations then perform 250k iterations of learning-rate cooldown following a cosine schedule before applying the high-resolution phase described in [Sec. 5.1](#) above without Gram anchoring.

**Efficient Multi-Student Distillation** As the inference cost for a large teacher can be orders of magnitude higher than for students (see [Fig. 16a](#)), we design a parallel distillation pipeline that allows training multiple students at the same time and sharing the teacher inference across all nodes involved in the training (see [Fig. 12](#) for a diagram). Let  $C_T$  and  $C_S$  be respectively the cost of running the teacher inference and the student training on a single sample, in single-teacher/single-student distillation with batch-size  $B$  where each of the  $N$  GPUs processes a  $B/N$  slice of the data, the teacher inference costs  $B/N \times C_T$  and the student training costs  $B/N \times C_S$  per GPU. In multi-student distillation, we proceed as follows. Each student  $S_i$  is assigned a set of  $N_{S_i}$  GPUs for training, and all  $N_T = \sum N_{S_i}$  GPUs are part of the global inference group. At each iteration, we first run the teacher inference on the global group for a  $B/N_T \times C_T$  compute cost per GPU. We then run an *all-gather* collective operation to share the input data and inference result with all compute nodes. Finally, each student group separately performs student training for a  $B/N_{S_i} \times C_{S_i}$  cost.

The above calculations shows that adding an additional student to the distillation pipeline will (1) reduce the per-GPU compute at each iteration, thus globally improving distillation speed, and (2) increase the overall compute only by the training cost of the new student, since the total teacher inference cost is now fixed. The implementation only requires setting up GPU process groups carefully, adapting data-loaders and teacher inference to ensure inputs and outputs are synchronized across groups using NCCL collectives. As the groups are synchronized at each iteration, in order to maximize speed, we adapt the number of GPUs for each student such that their iteration times are roughly the same. With this procedure, we seamlessly train multiple students, and produce a whole family of distilled models from our flagship 7B model.

### 5.3 Aligning DINOv3 with Text

Open-vocabulary image-text alignment has received significant interest and enthusiasm from the research community, thanks to its potential to enable flexible and scalable multimodal understanding. A large body

---

of work has focused on improving the quality of CLIP (Radford et al., 2021), which originally learned only a global alignment between image and text representations. While CLIP has demonstrated impressive zero-shot capabilities, its focus on global features limits its ability to capture fine-grained, localized correspondences. More recent works (Zhai et al., 2022b) have shown that effective image-text alignment can be achieved with pre-trained self-supervised visual backbones. This makes it possible to leverage these powerful models in multi-modal settings, facilitating richer and more precise text-to-image associations that extend beyond global semantics while also reducing computational costs, since the visual encoding is already learned.

We align a text encoder with our DINOv3 model by adopting the training strategy previously proposed in Jose et al. (2025). This approach follows the LiT training paradigm (Zhai et al., 2022b), training a text representation from scratch to match images to their captions with a contrastive objective, while keeping the vision encoder frozen. To allow for some flexibility on the vision side, two transformer layers are introduced on top of the frozen visual backbone. A key enhancement of this method is the concatenation of the mean-pooled patch embeddings with the output CLS token before matching to the text embeddings. This enables aligning both global and local visual features to text, leading to improved performance on dense prediction tasks without requiring additional heuristics or tricks. Furthermore, we use the same data curation protocol as established in Jose et al. (2025) to ensure consistency and comparability.

## 6 Results

In this section, we evaluate our flagship DINOv3 7B model on a variety of computer vision tasks. Throughout our experiments, unless otherwise specified, *we keep DINOv3 frozen* and solely use its representations. We demonstrate that with DINOv3, finetuning is not necessary to obtain strong performance. This section is organized as follows. We first probe the quality of DINOv3’s dense (Sec. 6.1) and global (Sec. 6.2) image representations using lightweight evaluation protocols and compare it to the strongest available vision encoders. We show that DINOv3 learns exceptional dense features while offering robust and versatile global image representations. Then, we consider DINOv3 as a basis for developing more complex computer vision systems (Sec. 6.3). We show with little effort on top of DINOv3, we are able to achieve results competitive with or exceeding the state of the art in tasks as diverse as object detection, semantic segmentation, 3D view estimation, or relative monocular depth estimation.

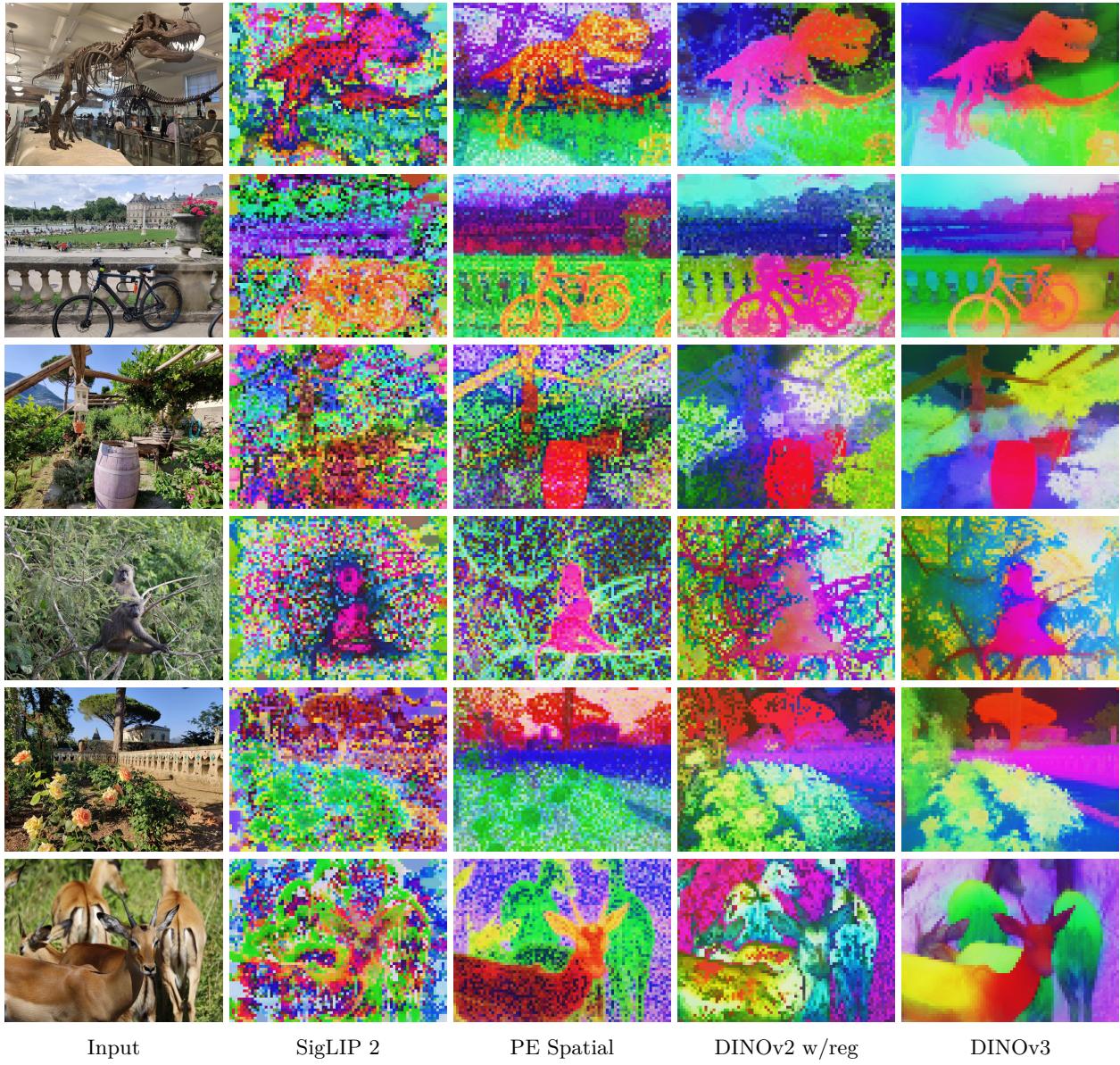
### 6.1 DINOv3 provides Exceptional Dense Features

We first investigate the raw quality of DINOv3’s dense representations using a diverse set of lightweight evaluations. In all cases, we utilize the frozen patch features of the last layer, and evaluate them using (1) qualitative visualizations (Sec. 6.1.1), (2) dense linear probing (Sec. 6.1.2: segmentation, depth estimation), (3) non-parametric approaches (Sec. 6.1.3: 3D correspondence estimation, Sec. 6.1.4: object discovery, Sec. 6.1.5: tracking), and (4) lightweight attentive probing (Sec. 6.1.6: video classification).

**Baselines** We compare the dense features of DINOv3 with those of the strongest publicly available image encoders, both weakly- and self-supervised ones. We consider the weakly-supervised encoders Perception Encoder (PE) Core (Bolya et al., 2025) and SigLIP 2 (Tschannen et al., 2025), which use CLIP-style image-text contrastive learning. We also compare to the strongest self-supervised methods: DINOv3’s predecessor DINOv2 (Oquab et al., 2024) with registers (Darisetty et al., 2024), Web-DINO (Fan et al., 2025), a recent scaling effort of DINO, and Franca (Venkataramanan et al., 2025), the best open-data SSL model. Finally, we compare to the agglomerative models AM-RADIOv2.5 (Heinrich et al., 2025), distilled from DINOv2, CLIP (Radford et al., 2021), DFN (Fang et al., 2024a), and Segment Anything (SAM) (Kirillov et al., 2023), and to PEspatial, distilling SAM 2 (Ravi et al., 2025) into PEcore. For each baseline, we report the performance of the strongest model available and specify the architecture in the tables.

#### 6.1.1 Qualitative Analysis

We start by analyzing DINOv3’s dense feature maps qualitatively. To this end, we project the dense feature space into 3 dimensions using principal component analysis (PCA), and map the resulting 3D space into RGB. Because of the sign ambiguity in PCA (eight variants) and the arbitrary mapping between principal



**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.

components and colors (six variants), we explore all combinations and report the visually most compelling one. The resulting visualization is shown in Fig. 13. Compared to other vision backbones, it can be seen that the features of DINOv3 are sharper, containing much less noise, and showing superior semantical coherence.

### 6.1.2 Dense Linear Probing

We perform linear probing on top of the dense features for two tasks: semantic segmentation and monocular depth estimation. In both cases, we train a linear transform on top of the frozen patch outputs of DINOv3. For semantic segmentation, we evaluate on the ADE20k (Zhou et al., 2017), Cityscapes (Cordts et al., 2016), and PASCAL VOC 2012 (Everingham et al., 2012) datasets and report the mean intersection-over-union

**Table 3:** Dense linear probing results on semantic segmentation and monocular depth estimation with frozen backbones. We report the mean Intersection-over-Union (mIoU) metric for the segmentation benchmarks ADE20k, Cityscapes, and VOC. We report the Root Mean Squared Error (RMSE) metric for the depth benchmarks NYUv2 and KITTI. For segmentation, all models are evaluated with input resolution adapted to 1024 patch tokens (*i.e.*  $448 \times 448$  for patch size 14,  $512 \times 512$  for patch size 16).

Method	ViT	Segmentation			Depth	
		ADE20k	Citysc.	VOC	NYUv2 ↓	KITTI ↓
<i>Agglomerative backbones</i>						
AM-RADIOv2.5	g/14	53.0	78.4	85.4	0.340	2.918
PEspatial	G/14	49.3	73.2	82.7	0.362	3.082
<i>Weakly-supervised backbones</i>						
SigLIP 2	g/16	42.7	64.8	72.7	0.494	3.273
PEcore	G/14	38.9	61.1	69.2	0.590	4.119
<i>Self-supervised backbones</i>						
Franca	g/14	46.3	68.7	82.9	0.445	3.140
DINOv2	g/14	49.5	75.6	83.1	0.372	2.624
Web-DINO	7B/14	42.7	68.3	76.1	0.466	3.158
DINOv3	7B/16	<b>55.9</b>	<b>81.1</b>	<b>86.6</b>	<b>0.309</b>	<b>2.346</b>

(mIoU) metric. For depth estimation, we use the NYUv2 (Silberman et al., 2012) and KITTI (Geiger et al., 2013) datasets and report the root mean squared error (RMSE).

**Results (Tab. 3)** The segmentation results demonstrate the superior quality of our dense features. On the general ADE20k dataset, DINOv3 outperforms the self-supervised baselines by more than 6 mIoU points, and the weakly supervised baselines by more than 13 points. Furthermore, DINOv3 surpasses PEspatial by more than 6 points, and AM-RADIOv2.5 by nearly 3 points. These results are remarkable as both are strong baselines, being distilled from the heavily supervised segmentation model SAM (Kirillov et al., 2023). Similar results are observed on the self-driving benchmark Cityscapes, with DINOv3 achieving the best mIoU of 81.1, surpassing AM-RADIOv2.5 by 2.5 points, and all other backbones by at least 5.5 points.

On monocular depth estimation, DINOv3 again outperforms all other models by significant margins: the weakly-supervised models PEcore and SigLIP 2 are still lagging, with DINOv2 and the more advanced models derived from SAM are the closest competitors. Interestingly, while PEspatial and AM-RADIO show strong performance on NYU, their performance is lower than DINOv2’s on KITTI. Even there, DINOv3 outperforms its predecessor DINOv2 by 0.278 RMSE.

Both sets of evaluations show the outstanding representation power of the dense features of DINOv3 and reflect the visual results from Fig. 13. With only a linear predictor, DINOv3 allows robust prediction of object categories and masks, as well as physical measurements of the scene such as relative depth. These results show that the features are not only visually sharp and properly localized, they also represent many important properties of the underlying observations in a linearly separable way. Finally, the absolute performance obtained with a linear classifier on ADE20k (55.9 mIoU) is itself impressive, as it is not far from the absolute the state-of-the-art (63.0 mIoU) on this dataset.

### 6.1.3 3D Correspondence Estimation

Understanding the 3D world has always been an important goal of computer vision. Image foundation models have recently fueled research in 3D understanding by offering *3D-aware features*. In this section, we evaluate the *multi-view consistency* of DINOv3—that is, whether patch features of the same keypoint in different views of an object are similar—following the protocol defined in Probe3D (Banani et al., 2024). We distinguish between *geometric* and *semantic* correspondence estimation. The former refers to matching keypoints for the *same object instance* while the latter refers to matching keypoints for different instances of the *same object class*. We evaluate geometric correspondence on the NAVI dataset (Jampani et al., 2023) and semantic

**Table 4:** Evaluation of 3D consistency of dense representations. We estimate 3D keypoint correspondences across views following the evaluation protocol of Probe3D (Banani et al., 2024). To measure performance, we report the correspondence recall, *i.e.* the percentage of correspondences falling into a specified distance.

Method	ViT	Geometric		Semantic	
		NAVI	SPair	NAVI	SPair
<i>Agglomerative backbones</i>					
AM-RADIOv2.5	g/14	59.4	56.8		
PEspatial	G/14	53.8	49.6		
<i>Weakly-supervised backbones</i>					
SigLIP 2	g/16	49.4	42.6		
PEcore	G/14	39.9	23.1		
<i>Self-supervised backbones</i>					
Franca	g/14	54.6	51.0		
DINOv2	g/14	60.1	56.1		
Web-DINO	7B/14	55.0	32.2		
DINOv3	7B/16	<b>64.4</b>	<b>58.7</b>		

correspondence on the SPair dataset (Min et al., 2019), and measure performance with correspondence recall in both cases. Please refer to App. D.3 for more experimental details.

**Results (Tab. 4)** For geometric correspondences, DINOv3 outperforms all other models and improves over the second best model (DINOv2) by 4.3% recall. Other SSL scaling endeavors (Franca and WebSSL) lag behind DINOv2, showing that it is still a strong baseline. Weakly-supervised models (PEcore and SigLIP 2) do not fare well on this task, indicating a lack of 3D awareness. For models with SAM distillation, AM-RADIO nearly reaches the performance of DINOv2, but PEspatial still lags behind it (−11.6% recall), and even falls behind Franca (−0.8% recall). This suggests that self-supervised learning is a key component for strong performance on this task. For semantic correspondences, the same conclusions apply. DINOv3 performs best, outperforming both its predecessor (+2.6% recall) and AM-RADIO (+1.9% recall). Overall, these impressive performance on keypoint matching are very promising signals for downstream use of DINOv3 in other 3D-heavy applications.

#### 6.1.4 Unsupervised Object Discovery

Powerful self-supervised features facilitate discovering object instances in images without requiring *any* annotations (Vo et al., 2021; Siméoni et al., 2021; Seitzer et al., 2023; Wang et al., 2023c; Siméoni et al., 2025). We test this capability for different vision encoders via the task of unsupervised object discovery, which requires class-agnostic segmentation of objects in images (Russell et al., 2006; Tuytelaars et al., 2010; Cho et al., 2015; Vo et al., 2019). In particular, we use the non-parametric graph-based TokenCut algorithm (Wang et al., 2023c), which has shown strong performance on a variety of backbones. We run it on three widely used datasets: VOC 2007, VOC 2012 (Everingham et al., 2015), and COCO-20k (Lin et al., 2014; Vo et al., 2020). We follow the evaluation protocol defined by Siméoni et al. (2021) and report the CorLoc metric. To properly compare backbones with different feature distributions, we perform a search over the main TokenCut hyperparameter, namely the cosine similarity threshold applied when constructing the patch graph used for partitioning. Originally, the best object discovery results were obtained with DINO (Caron et al., 2021) using the keys of the last attention layer. However, this hand-crafted choice does not consistently generalize to other backbones. For simplicity, we always employ the output features for all models.

**Results (Fig. 14)** The original DINO has set a very high bar for this task. Interestingly, while DINOv2 has shown very strong performance for pixel-wise dense tasks, it fails at object discovery. This can in part be attributed to the artifacts present in the dense features (*c.f.* Fig. 13). DINOv3, with its clean and precise output feature maps outperforms both its predecessors, with a 5.9 CorLoc improvement on VOC 2007, and all other backbones, whether self-, weakly-supervised or agglomerative. This evaluation confirms that

Method	ViT	VOC07	VOC12	COCO
<i>Agglomerative backbones</i>				
AM-RADIOv2.5	g/14	55.0	59.7	45.9
PEspatial	G/14	51.2	56.0	43.9
<i>Weakly-supervised backbones</i>				
SigLIPv2	g/16	20.5	24.7	18.6
PEcore	G/14	14.2	18.2	13.5
<i>Self-supervised backbones</i>				
DINO	S/16	61.1	66.0	48.7
DINO	B/16	60.1	64.4	50.5
DINOv2	g/14	55.6	60.4	45.4
Web-DINO	7B/14	26.1	29.7	20.9
DINOv3	7B/16	<b>66.1</b>	<b>69.5</b>	<b>55.1</b>



**Figure 14:** Unsupervised object discovery. We apply TokenCut (Wang et al., 2022c) on the output patch features of different backbones and report CorLoc metric. We also visualize predicted masks obtained with DINOv3 (red overlay on input images at res. 1024), obtained *with no annotation and no post-processing*.

DINOv3’s dense features are both semantically strong and well localized. We believe that this will pave the way for more class-agnostic object detection approaches, especially in scenarios where annotations are costly or unavailable, and where the set of relevant classes is not confined to a predefined subset.

### 6.1.5 Video Segmentation Tracking

Beyond static images, an important property of visual representations is their *temporal consistency*, *i.e.* whether the features evolve in a stable manner through time. To test for this property, we evaluate DINOv3 on the task of video segmentation tracking: given ground-truth instance segmentation masks in the first frame of a video, the goal is to propagate these masks to subsequent frames. We use the DAVIS 2017 (Pont-Tuset et al., 2017), YouTube-VOS (Xu et al., 2018), and MOSE (Ding et al., 2023) datasets. We evaluate performance using the standard  $\mathcal{J}$ & $\mathcal{F}$ -mean metric, which combines region similarity ( $\mathcal{J}$ ) and contour accuracy ( $\mathcal{F}$ ) (Perazzi et al., 2016). Following Jabri et al. (2020), we use a non-parametric label propagation algorithm that considers the similarity between patch features across frames. We evaluate at three input resolutions, using a short side length of 420/480 (S), 840/960 (M), and 1260/1440 (L) pixels for models with patch size 14/16 (matching the number of patch tokens). The  $\mathcal{J}$ & $\mathcal{F}$  score is always computed at the native resolution of the videos. See App. D.5 for more detailed experimental settings.

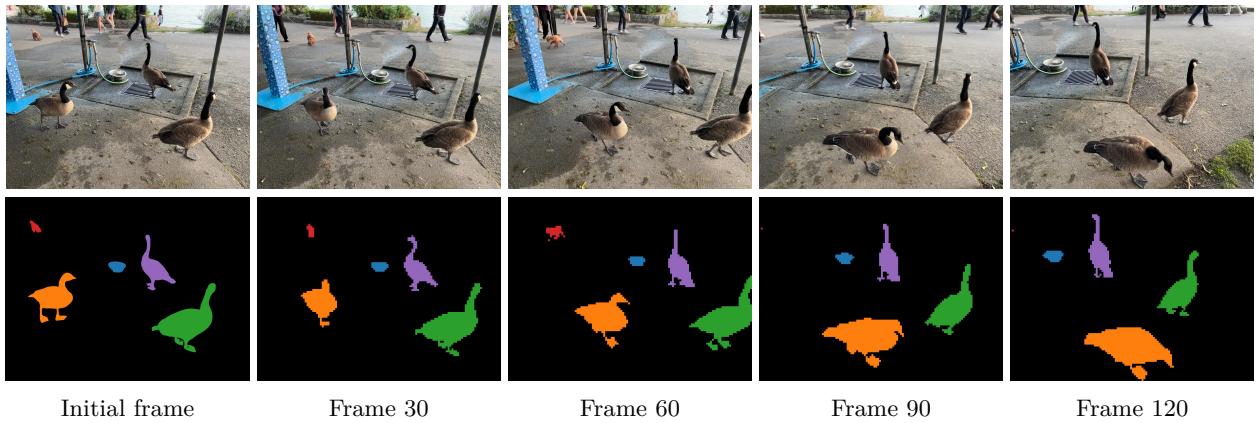
**Results (Tab. 5)** Aligned with all previous results, weakly-supervised backbones do not deliver convincing performance. PEspatial, distilled from the video model SAMv2, provides satisfactory performance, surpassing DINOv2 on smaller resolutions, but falling short on larger ones. Across resolutions, DINOv3 outperforms all competitors, with a staggering 83.3  $\mathcal{J}$ & $\mathcal{F}$  on DAVIS-L, 6.7 points above DINOv2. Furthermore, performance as a function of resolution follows a healthy trend, confirming that our model is able to make use of more input pixels to output precise, high-resolution feature maps (*c.f.* Figs. 3 and 4). In contrast, performance at higher resolutions stays almost flat for SigLIP 2 and PEcore, and degrades for PEspatial. Interestingly, our image model, without any tuning on video, allows to properly track objects in time (see Fig. 15). This makes it a great candidate to embed videos, allowing to build strong video models on top.

### 6.1.6 Video Classification

The previous results have shown the low-level temporal consistency of DINOv3’s representations, allowing to accurately track objects in time. Going beyond, we evaluate in this section the suitability of its dense features for high-level video classification. Similar to the setup of V-JEPA 2 (Assran et al., 2025), we train an *attentive probe*—a shallow 4-layer transformer-based classifier—on top of patch features extracted from each frame. This enables reasoning over temporal and spatial dimensions as the features are extracted independently per

**Table 5:** Video segmentation tracking evaluation. We report the  $\mathcal{J}$ & $\mathcal{F}$ -mean on DAVIS, YouTube-VOS, and MOSE at multiple resolutions. For models with patch size 14/16, the small, medium and large resolutions correspond to a video short side of 420/480, 840/960, 1260/1140 pixels.

Method	ViT	DAVIS			YouTube-VOS			MOSE		
		S	M	L	S	M	L	S	M	L
<i>Agglomerative backbones</i>										
AM-RADIOv2.5	g/14	66.5	77.3	81.4	70.1	78.1	79.2	44.0	52.6	54.3
PEspatial	G/14	68.4	74.5	70.5	68.5	67.5	55.6	39.3	40.2	34.0
<i>Weakly-supervised backbones</i>										
SigLIP 2	g/16	56.1	62.3	62.9	52.0	57.3	55.1	28.0	30.3	29.2
PEcore	G/14	48.2	53.1	49.8	34.7	33.0	25.3	17.8	19.0	15.4
<i>Self-supervised backbones</i>										
Franca	g/14	61.8	66.9	66.5	67.3	70.5	67.9	40.3	42.6	41.9
DINOv2	g/14	63.9	73.6	76.6	65.6	73.5	74.6	40.4	47.6	48.5
Web-DINO	7B/14	57.2	65.8	69.5	43.9	49.6	50.9	24.9	29.9	31.1
DINOv3	7B/16	<b>71.1</b>	<b>79.7</b>	<b>83.3</b>	<b>74.1</b>	<b>80.2</b>	<b>80.7</b>	<b>46.0</b>	<b>53.9</b>	<b>55.6</b>



**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.

frame. During evaluation, we either take a single clip per video, or use test-time augmentation (TTA) by averaging the predictions of 3 spatial and 2 temporal crops per video. See [App. D.6](#) for experimental details. We run this evaluation on three datasets: UCF101 ([Soomro et al., 2012](#)), Something-Something V2 ([Goyal et al., 2017](#)), and Kinetics-400 ([Kay et al., 2017](#)), and report top-1 accuracy. As an additional baseline, we report the performance of V-JEPA v2, a state-of-the-art SSL model for video understanding.

**Results (Tab. 6)** In line with the conclusion of the previous experiment, we find that DINOv3 can be successfully used for extracting strong video features. As this evaluation involves training several layers of self-attention, the differences between models are less visible. However, DINOv3 lands in the same range as PEcore and SigLIP 2, and clearly outperforms other models (DINOv2, AM-RADIO) across datasets. UCF101 and K400 are appearance-focused, where strong category-level understanding of objects gives most of the performance. SSv2 on the other hand, requires better understanding of motion—the dedicated video model V-JEPA v2 shines on this dataset. Interestingly, the gap between DINOv3 and the weakly-supervised models is slightly bigger on this dataset. This again confirms the suitability of DINOv3 to video tasks.

**Table 6:** Video classification evaluation using attentive probes. We report top-1 accuracy on UCF101, Something-Something V2 (SSv2), and Kinetics-400 (K400). For each model, we report performance for evaluating a single clip per video, or applying test-time augmentation (TTA) by averaging the predicted probabilities from multiple clips.

Method	ViT	UCF101		SSv2		K400	
		Single	TTA	Single	TTA	Single	TTA
<i>Agglomerative backbones</i>							
AM-RADIOv2.5	g/14	92.8	92.5	69.1	70.0	84.8	85.2
PEspatial	G/14	92.7	92.8	66.4	68.4	83.5	84.8
<i>Weakly-supervised backbones</i>							
SigLIP 2	g/16	93.6	<b>94.2</b>	68.8	70.2	86.9	87.7
PEcore	G/14	93.1	93.3	69.0	70.4	<b>87.9</b>	<b>88.8</b>
<i>Self-supervised backbones</i>							
DINOv2	g/14	93.5	93.8	67.4	68.4	84.4	85.6
V-JEPA 2	g/16	<b>94.0</b>	93.8	<b>73.8</b>	<b>75.4</b>	83.3	84.3
Web-DINO	7B/14	93.9	94.1	67.3	68.1	86.8	87.2
DINOv3	7B/16	93.5	93.5	70.1	70.8	87.8	88.2

## 6.2 DINOv3 has Robust and Versatile Global Image Descriptors

In this section, we evaluate DINOv3’s ability to capture global image statistics. To this end, we consider classic classification benchmarks using linear probes (Sec. 6.2.1) and instance retrieval benchmarks (Sec. 6.2.2). Again, we compare to the strongest publicly available image encoders. In addition to the models from the previous section, we evaluate the two weakly supervised models AIMv2 (Fini et al., 2024), trained using joint auto-regressive pixel and text prediction, and the massive EVA-CLIP-18B (Sun et al., 2024).

### 6.2.1 Image Classification with Linear Probing

We train a linear classifier on top of DINOv3’s output CLS token to evaluate the model on classification benchmarks. We consider the ImageNet1k (Deng et al., 2009) dataset and its variants to evaluate out-of-distribution robustness, and a suite of datasets from different domains to understand DINOv3’s ability to distinguish fine-grained classes. See App. D.7 for evaluation details.

**Domain Generalization from ImageNet (Tab. 7)** In this experiment, we train on ImageNet-*train*, use ImageNet-*val* as a *validation set* to select hyperparameters, and transfer the best found classifier to different test datasets: ImageNet-**V2** (Recht et al., 2019) and **ReaL** (Beyer et al., 2020) are alternative sets of images and labels for ImageNet, used to test overfitting on the ImageNet validation set; **Rendition** (Hendrycks et al., 2021a) and **Sketch** (Wang et al., 2019) show stylized and artificial versions of the ImageNet classes; **Adversarial** (Hendrycks et al., 2021b) and **ObjectNet** (Barbu et al., 2019) contain deliberately-chosen difficult examples; **Corruptions** (Hendrycks and Dietterich, 2019) measures robustness to common image corruptions. For reference, we also list linear probing results from Dehghani et al. (2023) for ViTs trained using supervised classification on the massive JFT dataset (3B–4B images). Note that these results follow a slightly different evaluation protocol and are not directly comparable to our results.

DINOv3 significantly surpasses all previous self-supervised backbones, with gains of +10% on ImageNet-R, +6% on -Sketch, +13% on ObjectNet over the previously strongest SSL model DINOv2. We note that the strongest weakly-supervised models, SigLIP 2 and PE, are now better than the strongest supervised ones (ViT-22B) on hard OOD tasks like ImageNet-A and ObjectNet. DINOv3 reaches comparable results on ImageNet-R and -Sketch, and, on the hard tasks ImageNet-A and ObjectNet, is closely behind PE, while exceeding SigLIPv2. On ImageNet, while validation scores are 0.7–0.9 points behind SigLIPv2 and PE, the performance on the “cleaner” test sets -V2 and -ReaL is virtually the same. Notably, DINOv3 achieves the best robustness to corruptions (ImageNet-C). All in all, *this is the first time that a SSL model has reached*

**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.*  $448 \times 448$  for patch size 14,  $512 \times 512$  for patch size 16). For reference, we also list results from Dehghani et al. (2023) using a different evaluation protocol (marked with \*).

Method	ViT	ImageNet			Rendition		Hard		
		Val	V2	ReaL	R	S	A	$C \downarrow$	Obj.
<i>Supervised backbones</i>									
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
<i>Agglomerative backbones</i>									
AM-RADIOv2.5	g/14	88.0	80.2	90.3	83.8	67.1	81.3	27.1	68.4
<i>Weakly-supervised backbones</i>									
PEcore	G/14	<b>89.3</b>	<b>81.6</b>	90.4	<b>92.2</b>	<b>71.9</b>	<b>89.0</b>	22.7	<b>80.2</b>
SigLIP 2	g/16	89.1	81.6	<b>90.5</b>	<b>92.2</b>	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
<i>Self-supervised backbones</i>									
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	<b>19.6</b>	79.0

**Table 8:** Finegrained classification benchmarks. Fine-S averages over 12 datasets, see Tab. 22 for full results.

Method	ViT	Fine-S	Places	iNat18	iNat21
<i>Agglomerative backbones</i>					
AM-RADIOv2.5	g/14	93.9	70.2	79.0	83.7
<i>Weakly-supervised backbones</i>					
SigLIP 2	g/16	93.7	70.5	80.7	82.7
PEcore	G/14	<b>94.5</b>	<b>71.3</b>	<b>86.6</b>	87.0
AIMv2	3B/14	92.9	70.7	80.8	83.2
EVA CLIP	18B/14	92.9	71.1	80.7	83.5
<i>Self-supervised backbones</i>					
Franca	g/14	87.7	64.6	61.4	70.6
DINOv2	g/14	92.6	68.2	80.7	86.1
Web-DINO	7B/14	90.2	69.6	65.3	74.1
DINOv3	7B/16	93.0	70.0	85.6	<b>89.8</b>

**Table 9:** Instance recognition benchmarks. See Tab. 23 for additional metrics.

Oxford-H	Paris-H	Met (GAP)	AmsterTime
47.5	85.7	30.5	23.1
25.1	60.9	13.9	15.5
32.7	68.9	10.6	23.1
28.8	71.4	29.5	14.6
27.1	65.6	0.5	18.9
14.3	51.6	27.2	21.1
58.2	84.6	44.6	48.9
31.2	80.3	35.2	30.6
<b>60.7</b>	<b>87.1</b>	<b>55.4</b>	<b>56.5</b>

comparable results to weakly- and supervised models on image classification—a domain which used to be the strong point of (weakly-)supervised training approaches. This is a remarkable result, given that models like ViT-22B, SigLIP 2, and PE are trained using massive human-annotated datasets. In contrast, DINOv3 learns purely from images, which makes it feasible to further scale/improve the approach in the future.

**Finegrained Classification (Tab. 8)** We also measure DINOv3’s performance when training linear probes on several datasets for fine-grained classification. In particular, we report the accuracy on 3 large datasets, namely Places205 (Zhou et al., 2014) for scene recognition, and iNaturalist 2018 (Van Horn et al., 2018) and iNaturalist 2021 (Van Horn et al., 2021)) for detailed plant and animal-species recognition, as well as the average over 12 smaller datasets covering scenes, objects, and textures (as in Oquab et al. (2024), here termed Fine-S). See also Tab. 22 for individual results on those datasets.

---

We find that, again, DINOv3 surpasses all previous SSL methods. It also shows competitive results compared to the weakly-supervised methods, indicating its robustness and generalization capability across diverse finegrained classification tasks. Notably, DINOv3 attains the highest accuracy on the difficult iNaturalist21 dataset at 89.8%, outperforming even the best weakly-supervised model PEcore with 87.0%.

### 6.2.2 Instance Recognition

To evaluate the instance-level recognition capabilities of our model, we adopted a non-parametric retrieval approach. Here, database images are ranked by their cosine similarity to a given query image, using the output CLS token. We benchmark performance across several datasets: the Oxford and Paris datasets for landmark recognition (Radenović et al., 2018), the Met dataset featuring artworks from the Metropolitan Museum (Ypsilantis et al., 2021), and AmsterTime, which consists of modern street view images matched to historical archival images of Amsterdam (Yıldız et al., 2022). Retrieval effectiveness is quantified using mean average precision for Oxford, Paris, and AmsterTime, and global average precision for Met. See App. D.8 for more evaluation details.

**Results (Tabs. 9 and 23)** Across all evaluated benchmarks, DINOv3 achieves the strongest performance by large margins, *e.g.* improving over the second best model DINOv2 by +10.8 points on Met and +7.6 points on AmsterTime. On this benchmark, weakly-supervised models are lagging far behind DINOv3, with the exception of AM-RADIO, which is distilled from DINOv2 features. These findings highlight the robustness and versatility of DINOv3 for instance-level retrieval tasks, spanning both traditional landmark datasets and more challenging domains such as art and historical image retrieval.

## 6.3 DINOv3 is a Foundation for Complex Computer Vision Systems

The previous two sections already provided solid signal for the quality of DINOv3 in both dense and global tasks. However, these results were obtained under “model probing” experimental protocols, using lightweight linear adapters or even non-parametric algorithms to assess the quality of features. While such simple evaluations allowed to remove confounding factors from involved experimental protocols, they are not enough to evaluate the full potential of DINOv3 as a foundational component in a larger computer vision system. Thus, in this section, we depart from the lightweight protocols, and instead train more involved downstream decoders and consider stronger, task-specific baselines. In particular, we use DINOv3 as a basis for (1) object detection with Plain-DETR (Sec. 6.3.1), (2) semantic segmentation with Mask2Former (Sec. 6.3.2), (3) monocular depth estimation with Depth Anything (Sec. 6.3.3), and (4) 3D understanding with the Visual Geometry Grounded Transformer (Sec. 6.3.4). These tasks are only intended as explorations for what is possible with DINOv3. Still, we find that building on DINOv3 unlocks competitive or even state-of-the-art results with little effort.

### 6.3.1 Object Detection

As a first task, we tackle the long-standing computer vision problem of object detection. Given an image, the goal is to provide bounding boxes for all instances of objects of pre-defined categories. This task requires both precise localization and good recognition, as boxes need to match the object boundaries and correspond to the correct category. While performance on standard benchmarks like COCO (Lin et al., 2014) is mostly saturated, we propose to tackle this task with a *frozen* backbone, only training a small decoder on top.

**Datasets and Metrics** We evaluate DINOv3 on object detection capabilities with the COCO dataset (Lin et al., 2014), reporting results on the COCO-VAL2017 split. Additionally, we evaluate out-of-distribution performance on the COCO-O evaluation dataset (Mao et al., 2023). This dataset contains the same classes but provides input images under six distribution shift settings. For both datasets, we report mean Average Precision (mAP) with IoU thresholds in  $[0.5 : 0.05 : 0.95]$ . For COCO-O, we additionally report the effective robustness (ER). Since COCO is a small dataset, comprising only 118k training images, we leverage the larger Objects365 dataset (Shao et al., 2019) for pre-training the decoder, as is common practice.

**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.

Model	Detector	FT	Parameters			COCO		COCO-O	
			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	<b>65.6</b>	<b>66.1</b>	<b>66.4</b>	<b>36.8</b>

**Implementation** We build upon the Plain-DETR (Lin et al., 2023b), but make the following modification: We do not fuse the transformer encoder into the backbone, but keep it as a separate module, similar to the original DETR (Carion et al., 2020), which allows us to keep the DINOv3 backbone completely frozen during training and inference. To the best of our knowledge, this makes it *the first competitive detection model to use a frozen backbone*. We train the Plain-DETR detector on Objects365 for 22 epochs at resolution 1536, then one epoch at resolution 2048, followed by 12 epochs on COCO at resolution 2048. At inference time, we run at resolution 2048. Optionally, we also apply test-time augmentation (TTA) by forwarding the image at multiple resolutions (from 1536 to 2880). See App. D.9 for full experimental details.

**Results (Tab. 10)** We compare our system with four models: EVA-02 with a Cascade detector (Fang et al., 2024b), EVA-02 with Co-DETR (Zong et al., 2023), InternImage-G with DINO (Wang et al., 2023b), and PEspatial with DETA (Bolya et al., 2025). We find that our lightweight detector (100M parameters) trained on top of a frozen DINOv3 backbone manages to reach state-of-the-art performance. For COCO-O, the gap is pronounced, showing that the detection model can effectively leverage the robustness of the DINOv3. Interestingly, our model outperforms all previous models with much fewer trained parameters, with the smallest comparison point still using more than 300M trainable parameters. We argue that achieving such strong performance without specializing the backbone is an enabler for various practical applications: A single backbone forward can provide features that support multiple tasks, reducing compute requirements.

### 6.3.2 Semantic Segmentation

Following the previous experiment, we now evaluate on semantic segmentation, another long-standing computer vision problem. This task also requires strong, well localized representations, and expects a dense per-pixel prediction. However, opposed to object detection, the model does not need to differentiate instances of the same object. Similar to detection, we train a decoder on top of a *frozen* DINOv3 model.

**Datasets and Metrics** We focus our evaluation on the ADE20k dataset (Zhou et al., 2017), which contains 150 semantic categories across 20k training images and 2k validation images. We measure performance using the mean Intersection over Union (mIoU). To train the segmentation model, we additionally use the COCO-Stuff (Caesar et al., 2018) and Hypersim (Roberts et al., 2021) datasets. Those contain 164k images with 171 semantic categories, and 77k images with 40 categories respectively.

**Implementation** To build a decoder that maps DINOv3 features to semantic categories, we combine ViT-Adapter (Chen et al., 2022) and Mask2Former (Cheng et al., 2022), similar to prior work (Wang et al., 2022b; 2023b;a). However, in our case, the DINOv3 backbone remains frozen during training. In order to avoid altering the backbone features, we further modify the original ViT-Adapter architecture by removing the injector component. Compared to baselines, we also increase the embedding dimensions from 1024 to 2048, to support processing the 4096-dimensional output of the DINOv3 backbone. We start by pre-training the

**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

Model	FT	Parameters			mIoU	
		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	<b>63.0</b>
DINOv3	✳️	7B	927M	927M	<b>62.6</b>	<b>63.0</b>

segmentation decoder on COCO-Stuff for 80k iterations, followed by 10k iterations on Hypersim (Roberts et al., 2021). Finally, we train for 20k iterations on the training split of ADE20k and report results on the validation split. All training is done at an input resolution of 896. At inference time we consider two setups: single-scale, *i.e.* we forward images at training resolution, or multi-scale, *i.e.* we average predictions at multiple image ratios between  $\times 0.9$  and  $1.1$  the original training resolution. We refer to App. D.10 for more experimental details.

**Results (Tab. 11)** We compare our model’s performance with several state-of-the-art baselines, including BEIT-3 (Wang et al., 2022b), InternImage-H (Wang et al., 2023b) and ONE-PEACE (Wang et al., 2023a), and report results on additional datasets in Tab. 24. Our segmentation model based on the frozen DINOv3 backbone reaches state-of-the-art performance, equaling that of ONE-PEACE (63.0 mIoU). It also improves over all prior models on the COCO-Stuff (Caesar et al., 2018) and VOC 2012 (Everingham et al., 2012) datasets. As semantic segmentation requires accurate per-pixel predictions, vision transformer backbones pose a fundamental problem. Indeed, the 16 pixel-wide input patches make the granularity of the prediction relatively coarse—encouraging solutions like ViT-Adapter. On the other hand, we have shown that we can obtain high-quality feature maps, even at very high resolutions up to 4096 (*c.f.* Figs. 3 and 4); this corresponds to dense feature maps 512-tokens wide. We hope that future work will be able to leverage these high-resolution features to reach state-of-the-art performance without having to rely on heavy decoders like ViT-Adapter with Mask2Former.

### 6.3.3 Monocular Depth Estimation

We now consider building a system for monocular depth estimation. To do so, we follow the setup of Depth Anything V2 (DAv2) (Yang et al., 2024b), a recent state-of-the-art method. The key innovation of DAv2 is to use a large collection of synthetically generated images with ground truth depth annotations. Critically, this relies on DINOv2 as a feature extractor that is able to bridge the *sim-to-real* gap, a capability that other vision backbones like SAM (Kirillov et al., 2023) do not show (Yang et al., 2024b). Thus, we swap DINOv2 with DINOv3 in the DAv2 pipeline to see if we can achieve similar results.

**Implementation** Like DAv2, we use a Dense Prediction Transformer (DPT) (Ranftl et al., 2021) to predict a pixelwise depth field, using features from four equally spaced layers of DINOv3 as input. We train the model using the set of losses from DAv2 on DAv2’s synthetic dataset, increasing the training resolution to  $1024 \times 768$  to make use of DINOv3’s high resolution capabilities. In contrast to DAv2, we *keep the backbone frozen* instead of finetuning it, testing the out-of-the-box capabilities of DINOv3. We also found it beneficial to scale up the DPT head to obtain the full potential DINOv3 7B’s larger features. See App. D.11 for details.

**Datasets and Metrics** We evaluate our model on 5 real-world datasets (NYUv2 (Silberman et al., 2012), KITTI (Geiger et al., 2013), ETH3D (Schöps et al., 2017), ScanNet (from Ke et al. (2025)) and DIODE (Vasiljevic et al., 2019)) in the zero-shot scale-invariant depth setup, similar to Ranftl et al. (2020); Ke et al. (2025);

**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.

Method	FT	NYUv2		KITTI		ETH3D		ScanNet		DIODE	
		ARel $\downarrow$	$\delta_1 \uparrow$								
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	<b>18.2</b>	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	✳️	<b>4.3</b>	<b>98.0</b>	<b>7.3</b>	<b>96.7</b>	<b>5.4</b>	<b>97.5</b>	<b>4.4</b>	<b>98.1</b>	25.6	<b>82.2</b>

Yang et al. (2024b). We report the standard metrics absolute relative error (ARel) (lower is better) and  $\delta_1$  (higher is better). We refer to Yang et al. (2024a) for a description of those metrics.

**Results (Tab. 12)** We compare to the state of the art for relative depth estimation: MiDaS (Ranftl et al., 2020), LeReS (Yin et al., 2021), Omnidata (Eftekhar et al., 2021), DPT (Ranftl et al., 2021), Marigold in the ensemble version (Ke et al., 2025) and DAv2. Our depth estimation model reaches a new state-of-the-art on all datasets, only lacking behind in ARel on DIODE compared to DPT. Remarkably, this is possible using a *frozen backbone*, whereas all other baselines need to finetune the backbone for depth estimation. In addition, this validates that DINOv3 inherits DINOv2’s *strong sim-to-real capabilities*, a desirable property that opens up the possibility for downstream tasks to use synthetically generated training data.

### 6.3.4 Visual Geometry Grounded Transformer with DINOv3

Finally, we consider 3D understanding with the recent Visual Geometry Grounded Transformer (VGGT) (Wang et al., 2025). Trained on a large set of 3D-annotated data, VGGT learns to estimate all important 3D attributes of a scene, such as camera intrinsics and extrinsics, point maps, or depth maps, in a single forward pass. Using a simple, unified pipeline, it reaches state-of-the-art results on many 3D tasks while being more efficient than specialized methods—constituting a major advance in 3D understanding.

**Implementation** VGGT uses a DINOv2-pretrained backbone to obtain representations for different views of a scene, before fusing them with a transformer. Here, we simply swap the DINOv2 backbone with DINOv3, using our ViT-L variant (see Sec. 7) to match DINOv2 ViT-L/14 in the original work. We run the same training pipeline as VGGT, including finetuning of the image backbone. We switch the image resolution from  $518 \times 518$  to  $592 \times 592$  to accommodate DINOv3’s patch size 16 and keep the the results comparable to VGGT. We additionally adopt a small number of hyperparameter changes detailed in App. D.12.

**Datasets and Metrics** Following Wang et al. (2025), we evaluate on camera pose estimation on the Re10K (Zhou et al., 2018) and CO3Dv2 (Reizenstein et al., 2021) datasets, dense multi-view estimation on DTU (Jensen et al., 2014), and two-view matching on ScanNet-1500 (Dai et al., 2017). For camera pose estimation and two-view matching, we report the standard area-under-curve (AUC) metric. For multi-view estimation, we report the smallest L2-distance between prediction to ground truth as “Accuracy”, the smallest L2-distance from ground truth to prediction as “Completeness” and their average as ‘Overall’. We refer to Wang et al. (2025) for details about method and evaluation.

**Results (Tab. 13)** We find that VGGT equipped with DINOv3 *further improves over the previous state-of-the-art* set by VGGT on all three considered tasks—using DINOv3 leads to clear and consistent gains. This is encouraging, given that we only applied minimal tuning for DINOv3. These tasks span different levels of visual understanding: high-level abstraction of scene content (camera pose estimation), dense geometric prediction (multi-view depth estimation), and fine-grained pixel-level correspondence (view matching). To-

**Table 13:** 3D understanding using Visual Geometry Grounded Transformer (VG GT) (Wang et al., 2025). Simply by swapping DINOv2 for DINOv3 ViT-L as the image feature extractor in the VG GT 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.		(b) Multi-view estimation on DTU.			(c) View matching on ScanNet-1500.				
Method	Re10K	CO3Dv2	Method	Acc. $\downarrow$	Comp. $\downarrow$	Overall $\downarrow$	Method	AUC@5	AUC@10
DUS3R	67.7	76.7	Gipuma*	0.283	0.873	0.578	SuperGlue	16.2	33.8
MAS3R	76.4	81.8	CIDER*	0.417	0.437	0.427	LoFTR	22.1	40.8
VG GSFM v2	78.9	83.4	MAS3R*	0.403	0.344	0.374	DKM	29.4	50.7
CUT3R	75.3	82.8	GeoMVSNet*	0.331	0.259	0.295	CasMTR	27.1	47.0
FLARE	78.8	83.3	DUS3R	2.677	0.805	1.741	Roma	31.8	53.4
VG GT	85.3	88.2	VG GT	0.389	0.374	0.382	VG GT	33.9	55.2
DINOv3	<b>86.3</b>	<b>89.6</b>	DINOv3	<b>0.375</b>	<b>0.361</b>	<b>0.368</b>	DINOv3	<b>35.2</b>	<b>56.1</b>

gether with the previous results on correspondence estimation (Sec. 6.1.3) and depth estimation (Sec. 6.3.3), we take this as further empirical evidence for the strong suitability of DINOv3 as a basis for 3D tasks. Additionally, we anticipate further improvements from using the larger DINOv3 7B model.

## 7 Evaluating the Full Family of DINOv3 Models

In this section, we provide quantitative evaluations on the family of models distilled from our 7B-parameters model (See Sec. 5.2). This family includes variants based on the Vision Transformer (ViT) and the ConvNeXt (CNX) architectures. We provide the detailed parameter counts and inference FLOPs for all models in Fig. 16a. These models cover a wide range of computational budgets to accommodate a broad spectrum of users and deployment scenarios. We conduct a thorough evaluation of all ViT (Sec. 7.1) and ConvNeXt variants to assess their performance across tasks.

Figure 2 provides an overview comparison of the DINOv3 family versus other model collections. The DINOv3 family significantly outperforms all others on dense prediction tasks. This includes specialized models distilled from supervised backbones like AM-RADIO and PEspatial. At the same time, our models achieve similar results on classification tasks, making them the optimal choice across compute budgets.

In Sec. 7.1 detail our ViT models and compare them to other open-source alternatives. Then, in Sec. 7.2, we discuss the ConvNeXt models. Finally, following Sec. 5.3, we trained a text encoder aligned with the output of our ViT-L model. We present multi-modal alignment results for this model in Sec. 7.3.

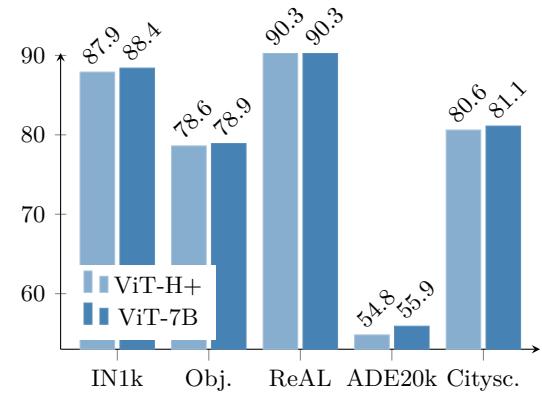
### 7.1 A Vision Transformer for Every Use Case

Our ViT family spans architectures from the compact ViT-S to the larger 840 million parameter ViT-H+ models. The former is designed to run efficiently on resource-constrained devices such as laptops, the latter delivers state-of-the-art performance for more demanding applications. We compare our ViT models to the best open-source image encoders of corresponding size, namely DINOv2 (Oquab et al., 2024), SigLIP 2 (Tschannen et al., 2025) and Perception Encoder (Bolya et al., 2025). For a fair comparison, we ensure that the input sequence length is equivalent across models. Specifically, for model with a patch size of 16 we input images of size  $512 \times 512$  versus  $448 \times 448$  when models are using patch size 14.

Our empirical study clearly demonstrates that DINOv3 models consistently outperform their counterparts on dense prediction tasks. Most notably, on the ADE20k benchmark, the DINOv3 ViT-L model achieves an improvement of over 6 mIoU points compared to the best competitor DINOv2. The ViT-B variant shows a gain of approximately 3 mIoU points against the next best competitor. These substantial improvements highlight the effectiveness of DINOv3’s local features in capturing fine-grained spatial details. Furthermore, evaluations on depth estimation tasks also reveal consistent performance gains over competing approaches.

Model	#Params	Inference GFLOPs	
		Res. 256	Res. 512
CNX-Tiny	29M	5	20
CNX-Small	50M	11	46
CNX-Base	89M	20	81
CNX-Large	198M	38	152
ViT-S	21M	12	63
ViT-S+	29M	16	79
ViT-B	86M	47	216
ViT-L	300M	163	721
ViT-H+	840M	450	1903
ViT-7B	6716M	3550	14515

(a) DINOV3 family of models.



(b) ViT-H+ v.s. ViT-7B.

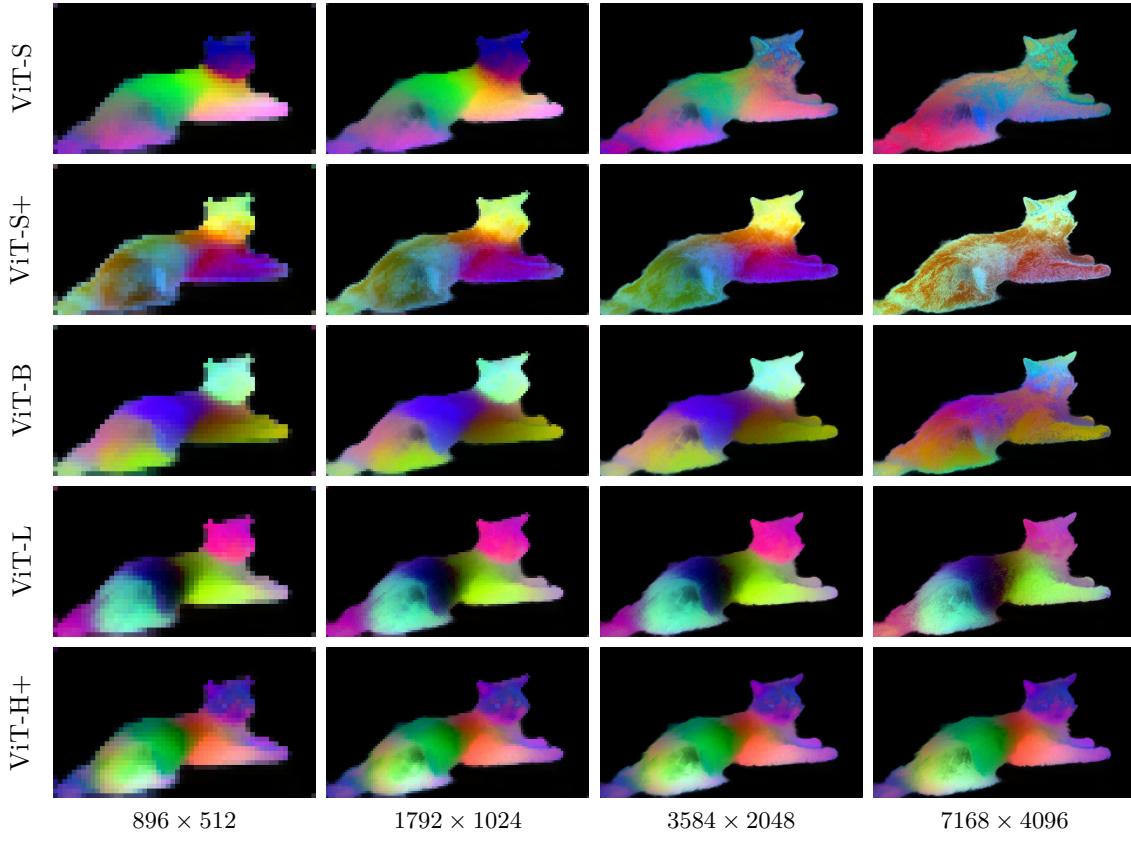
**Figure 16:** (a) Presentation of the distilled models’ characteristics. CNX stands for ConvNeXT. We present per model the number of parameters and the GFLOPs estimated on images of size  $256 \times 256$  and  $512 \times 512$ . (b) We compare DINOV3 ViT-H+ to its 7B-sized teacher; despite having almost  $10 \times$  less parameters, the ViT-H+ is close to DINOV3 7B in performance.

**Table 14:** Comparison of our family of models against open-source alternatives of comparable size. We showcase our ViT- $\{S, S+, B, L, H+\}$  models on a representative set of global and dense benchmarks: classification (IN-ReAL, IN-R, ObjectNet), retrieval (Oxford-H), segmentation (ADE20k), depth (NYU), tracking (DAVIS at 960px), and keypoint matching (NAVI, SPair). We match the number of patch tokens for a fair comparison across models of different patch size.

Size	Model	Global Tasks				Dense Tasks				
		IN-ReAL	IN-R	Obj.	Ox-H	ADE20k	NYU $\downarrow$	DAVIS	NAVI	SPair
S	DINOv2	87.3	54.0	47.8	39.5	45.5	0.446	73.6	53.4	51.6
S	DINOv3	87.0	60.4	50.9	49.5	47.0	0.403	72.7	56.3	50.4
S+	DINOv3	88.0	68.8	54.6	50.0	48.8	0.399	75.5	57.1	55.2
B	PEcore	87.5	68.4	57.9	20.2	37.4	0.641	44.5	41.8	13.7
B	SigLIP 2	89.3	80.6	66.9	20.2	41.6	0.512	63.2	45.4	32.8
B	DINOv2	89.0	68.4	57.3	51.0	48.4	0.416	72.9	56.9	57.1
B	DINOv3	89.3	76.7	64.1	58.5	51.8	0.373	77.2	58.8	57.2
L	PEcore	90.1	87.7	74.9	25.6	39.7	0.650	48.2	42.1	19.2
L	SigLIP 2	90.1	89.2	75.0	21.4	43.6	0.484	66.3	47.8	41.9
L	DINOv2	89.7	79.1	64.7	55.7	48.8	0.394	73.4	59.9	57.0
L	DINOv3	90.2	88.1	74.8	63.1	54.9	0.352	79.9	62.3	61.2
SO400m	SigLIP 2	90.3	90.4	76.2	23.0	44.0	0.402	64.8	48.8	38.7
H+	DINOv3	90.3	90.0	78.6	64.5	54.8	0.352	79.3	63.3	56.3

This underscores the versatility of the DINOV3 family across different dense vision problems. Importantly, our models achieve competitive results on global recognition benchmarks such as ObjectNet and ImageNet-1k. This indicates that the enhanced dense task performance does not come at the expense of global task accuracy. This balance confirms that DINOV3 models provide a robust and well-rounded solution, excelling across both dense and global vision tasks without compromise.

On another note, we want to also validate if the largest models that we distill capture all the information from the teacher. To this end, we run a comparison of our largest ViT-H+ with the 7B teacher. As shown in Fig. 16b, the largest student achieves performance that is on par with the 8 times larger ViT-7B model.



**Figure 17:** Stability of the features at multiple resolutions for the DINOv3 ViT family of models. Top-to-bottom: ViT-S, S+, B, L, H+. We run inference on an image at multiple resolutions, then perform principal component analysis on the features computed on a  $1792 \times 1024$  image ( $112 \times 64$  image tokens). We then project features at all resolutions onto the principal components 5–7 that we map to the RGB space for visualization. While the models are functional at all resolutions, we observe that the features remain consistent across a large range of resolutions before drifting: for example, ViT-S+ features are stable between  $896 \times 512$  and  $3584 \times 2048$  inputs, while ViT-L barely starts drifting on the largest resolution  $7168 \times 4096$ . ViT-H+ remains stable throughout the whole tested range.

This result not only validates the effectiveness of our distillation process but also demonstrates that, when guided by a high-quality teacher, smaller models can learn to deliver comparable levels of performance. This finding reinforces our belief that *training very large models benefits the broader community*. The strength of larger models can be successfully distilled into more efficient, smaller models with little or no loss of quality.

## 7.2 Efficient ConvNeXts for Resource-Constrained Environments

In this section, we evaluate the quality of our ConvNeXt (CNX) models distilled from the 7B teacher. ConvNeXt models are highly efficient in terms of FLOPs and are well-suited for deployment on devices optimized for convolutional computations. Furthermore, transformer models often do not lend themselves well to quantization (Bondarenko et al., 2021), whereas quantization of convolutional nets is a well explored subject. We distill CNX architectures of size T, S, B, and L (see Fig. 16a) and compare them to the original ConvNeXt models (Liu et al., 2022). These baselines achieve high performance on ImageNet-1k as they were trained in a supervised fashion using ImageNet-22k labels, and thus represent a strong competitor. For this experiment, we provide results for global tasks at input resolutions 256 and 512, for ADE20k at resolution 512, and for NYU at resolution 640.

**Table 15:** Evaluation of our distilled DINOV3 ConvNeXt models. We compare our models to off-the-shelf ConvNeXts trained supervised on ImageNet-22k (Liu et al., 2022). For global tasks, we give results at input resolutions 256 and 512, as we found the supervised models to significantly degrade at resolution 512.

Size	Model	Global Tasks						Dense Tasks	
		IN-ReAL		IN-R		Obj.		ADE20k	NYU $\downarrow$
		256	512	256	512	256	512		
T	Sup.	87.3	83.0	45.0	33.0	44.5	27.1	24.8	0.666
T	DINOv3	86.6	87.7	73.7	74.1	52.6	58.7	42.7	0.448
S	Sup.	88.9	86.8	52.8	39.1	50.8	40.0	22.6	0.630
S	DINOv3	87.9	88.7	73.7	74.1	52.6	58.7	44.8	0.432
B	Sup.	89.3	87.8	57.3	46.2	53.6	46.5	26.5	0.596
B	DINOv3	88.5	89.2	77.2	78.2	56.2	61.3	46.3	0.420
L	Sup.	89.6	88.1	58.4	46.6	55.0	47.7	33.3	0.567
L	DINOv3	88.9	89.4	81.3	82.4	59.3	65.2	47.8	0.403

**Results (Tab. 15)** We find that on in-distribution image classification, our models slightly lag behind the supervised ones at resolution 256 (e.g.  $-0.7$  IN-ReAL for CNX-T). However, the trend is reversed at resolution 512, with the supervised ConvNeXts significantly degrading, whereas our models scale with increased input resolution. For out-of-distribution classification (IN-R, ObjectNet), there are significant gaps between the two model families for all sizes—a testament to the robustness of the DINOV3 CNX models. Furthermore, the DINOV3 models offer very large improvement on dense tasks. Indeed, for CNX-T, our model yields a  $+17.9$  mIoU (42.7 versus 24.8) improvement, and for CNX-L, our model gets  $+14.5$  mIoU (47.8 versus 33.3). The combination of high performance and computational efficiency makes the distilled ConvNeXt models especially promising for real-world applications where resource constraints are critical. Aside from that, the distillation of the ViT-7B model into smaller ConvNeXt models is particularly exciting, as it bridges two fundamentally different architectures. While ViT-7B is based on transformer blocks with a CLS token, ConvNeXt relies on convolutional operations without a CLS token, making this transfer of knowledge non-trivial. This achievement highlights the versatility and effectiveness of our distillation process.

### 7.3 Zero-shot Inference with DINOV3-based dino.txt

As detailed in Sec. 5.3, we train a text encoder to align both the CLS token and the output patches of the distilled DINOV3 ViT-L model to text, following the recipe of dino.txt Jose et al. (2025). We evaluate the quality of the alignment both at the global- and patch-level on standard benchmarks. We report the zero-shot classification accuracy using the CLIP protocol (Radford et al., 2021) on the ImageNet-1k, ImageNet-Adversarial, ImageNet-Rendition and ObjectNet benchmarks. For image-text retrieval, we evaluate on the COCO2017 dataset (Tsung-Yi et al., 2017) and report Recall@1 on both image-to-text ( $I \rightarrow T$ ) and text-to-image ( $T \rightarrow I$ ) tasks. To probe the quality of patch-level alignment, we evaluate our model on the open-vocabulary segmentation task using the common benchmarks ADE20k and Cityscapes, for which we report the mIoU metric.

**Results (Tab. 16)** We compare our text-aligned DINOV3 ViT-L with competitors in the same size class. Compared to Jose et al. (2025), which aligns DINOV2 to text, DINOV3 leads to significantly better performance on all benchmarks. On global alignment tasks, we compare favorably to the original CLIP (Radford et al., 2021) and strong baselines such as EVA-02-CLIP (Sun et al., 2023) but slightly behind SigLIP2 (Tschanne et al., 2025) and Perception Encoder (Bolya et al., 2025). On dense alignment tasks, our text-aligned model shows excellent performance on two challenging benchmarks ADE20K and Cityscapes thanks to clean feature maps of DINOV3.

**Table 16:** Comparing our text-aligned DINOv3 ViT-L to the state-of-the-art. Our model achieves excellent dense alignment performance while staying competitive in global alignment tasks. All compared models are of ViT-L size and operate on the same sequence length of 576.

Method	Classification				Retrieval		Segmentation	
	IN1k	A	R	Obj.	I → T	T → I	ADE20k	Cityscapes
CLIP	76.6	77.5	89.0	72.3	57.9	37.1	6.0	11.5
EVA-02-CLIP	80.4	82.9	93.2	78.5	64.1	47.9	10.9	14.1
<code>dino.txt</code>	81.6	83.2	88.8	74.5	62.5	45.0	19.2	27.4
SigLIP 2	83.1	84.3	<b>95.7</b>	84.4	71.4	55.3	10.8	16.3
PE	<b>83.5</b>	<b>89.0</b>	95.2	<b>84.7</b>	<b>75.9</b>	<b>57.1</b>	17.6	21.4
DINOv3 <code>dino.txt</code>	82.3	85.4	93.0	80.5	63.7	45.6	<b>24.7</b>	<b>36.9</b>

## 8 DINOv3 on Geospatial Data

Our self-supervised learning recipe is generic and can be applied to any image domain. In this section, we showcase this universality by building a DINOv3 7B model for satellite images, which have very different characteristics (*e.g.* object texture, sensor noise, and focal views) than the web images on which DINOv3 was initially developed.

### 8.1 Pre-Training Data and Benchmarks

Our satellite DINOv3 7B model is pre-trained on SAT-493M, a dataset of 493 millions of  $512 \times 512$  images sampled randomly from Maxar RGB ortho-rectified imagery at 0.6 meter resolution. We use the exact same set of hyper-parameters that are used for the web DINOv3 7B model, except for the RGB mean and std normalization that are adapted for satellite images, and the training length. Similar to the web model, our training pipeline for the satellite model consists of 100k iterations of initial pre-training with global crops ( $256 \times 256$ ), followed by 10k iterations using Gram regularization, and finalized with 8k steps of high resolution fine-tuning at resolution 512. Also similar to the web model, we distill our 7B satellite model into a more manageable ViT-Large model to facilitate its use in low-budget regime.

We evaluate DINOv3 satellite and web models on multiple earth observation tasks. For the task of global canopy height mapping, we use the Satlidar dataset described in App. D.13, which consists of one million  $512 \times 512$  images with LiDAR ground truths split into train/val/test splits with ratios 8/1/1. The splits include the Neon and São Paulo dataset used by Tolan et al. (2024). For national-scale canopy height mapping, we evaluate on Open-Canopy (Fogel et al., 2025), which combines SPOT 6-7 satellite imagery and aerial LiDAR data over 87,000 km<sup>2</sup> across France. Since images in this dataset have 4 channels including the additional infra-red (IR) channel, we adapt our backbone by taking the average of the three channels in the weights of the patch embed module and adding it to the weights as the fourth channel. We trained a DPT decoder on  $512 \times 512$  crops of images resized to 1667 to match the Maxar ground sample resolution.

Semantic geospatial tasks are assessed with GEO-Bench (Lacoste et al., 2023), which comprises six classification and six segmentation tasks spanning various spatial resolutions and optical bands. The GEO-Bench tasks are diverse, including the detection of rooftop-mounted photovoltaic systems, classifying local climate zones, measuring drivers of deforestation, and detecting tree crowns. For high-resolution semantic tasks, we consider the land cover segmentation dataset LoveDA (Wang et al., 2022a), the object segmentation dataset iSAID (Zamir et al., 2019), and the horizontal detection dataset DIOR (Li et al., 2020).

### 8.2 Canopy Height Estimation

Estimating canopy height from satellite imagery is a challenging metric task, requiring accurate recovery of continuous spatial structure despite random variations in slope, viewing geometry, sun angle, atmospheric scattering, and quantization artifacts. This task is critical for global carbon monitoring and for forest and agriculture management (Harris et al., 2021). Following Tolan et al. (2024), the first work to leverage a SSL

**Table 17:** Evaluation of different backbones for high-resolution canopy height prediction. All models are trained with a DPT decoder. Results are presented either for experiments with the decoder trained on SatLidar and evaluated on IID samples (SatLidar Val) and OOD test sets (SatLidar Test, Neon and São Paulo), or for experiments with the decoder trained and evaluated on the Open-Canopy dataset. We list mean absolute error (MAE) and the block  $R^2$  metric from [Tolan et al. \(2024\)](#). For completeness, we additionally evaluate the original decoder of [Tolan et al. \(2024\)](#) that was trained on Neon dataset (denoted by \*).

Method	Arch.	SatLidar								Open Canopy	
		SatLidar Val		SatLidar Test		Neon Test		São Paulo			
		MAE↓	$R^2\uparrow$	MAE↓	$R^2\uparrow$	MAE↓	$R^2\uparrow$	MAE↓	$R^2\uparrow$		
<a href="#">Tolan et al. (2024)*</a>	ViT-L	2.8	0.86	4.0	0.61	2.7	0.73	<b>5.4</b>	0.42	—	
<a href="#">Tolan et al. (2024)</a>	ViT-L	2.4	0.90	3.4	0.81	2.9	0.69	<b>5.4</b>	0.48	2.42	
DINOv3 Web	ViT-7B	2.4	0.90	3.6	0.74	2.7	0.75	5.9	0.34	2.17	
DINOv3 Sat	ViT-L	<b>2.2</b>	0.91	<b>3.2</b>	0.81	<b>2.4</b>	<b>0.81</b>	5.8	0.42	2.07	
DINOv3 Sat	ViT-7B	<b>2.2</b>	<b>0.92</b>	<b>3.2</b>	<b>0.82</b>	2.6	0.74	5.5	<b>0.51</b>	<b>2.02</b>	

backbone trained on satellite images for this task, we train a DPT head on top of frozen DINOv3 on the SatLidar1M training set, then evaluate it on i.i.d. samples on SatLidar1M validation set as well as out-of-distribution test sets including SatLidar1M test, Neon and São Paulo. We additionally train and evaluate on the Open-Canopy dataset.

**Results (Tab. 17)** We compare different SSL backbones, denoting with “DINOv3 Sat” the model trained the SAT-493M dataset, and with “DINOv3 Web” the model trained on LVD-1689M (see Sec. 3.1). It can be seen that DINOv3 satellite models yield state-of-the-art performance on most benchmarks. Our 7B satellite model sets the new state of the art on SatLidar1M val, SatLidar1M test and Open-Canopy, reducing MAE from 2.4 to 2.2, from 3.4 to 3.2 and from 2.42 to 2.02 respectively. These results show that DINOv3 training recipe is generic and can be effectively applied out-of-the-box to other domains. Interestingly, our distilled ViT-L satellite model performs comparably to its 7B counterpart, achieving comparable results on SatLidar1M and Open-Canopy while faring surprisingly better on Neon test set, reaching the lowest MAE of 2.4 compared to 2.6 of the 7B model and 2.9 of [Tolan et al. \(2024\)](#). Our DINOv3 7B web model reaches decent performance on the benchmarks, outperforming [Tolan et al. \(2024\)](#) on SatLidar1M val, Neon and Open-Canopy but stays behind the satellite model. This highlights the strength of domain-specific pretraining for physically grounded tasks like canopy height estimation, where sensor-specific priors and radiometric consistency are important.

### 8.3 Comparison to the Earth Observation State of the Art

We compare the performance of different methods for Earth observation tasks in Tab. 18 and Tab. 19. The frozen DINOv3 satellite and web models set new state-of-the-art results on 12 out of 15 classification, segmentation, and horizontal object detection tasks. Our Geo-Bench results surpass prior models, including Prithvi-v2 ([Szwarcman et al., 2024](#)) and DOFA ([Xiong et al., 2024](#)), which use 6+ bands for Sentinel-2 and Landsat tasks, as well as task-specific fine-tuning (Tab. 18). Despite using a frozen backbone with RGB-only input, the DINOv3 satellite model outperforms previous methods on the three unsaturated classification tasks and on five of six segmentation tasks. Interestingly, the DINOv3 7B web model is very competitive on these benchmarks. It achieves comparable or stronger performance on many GEO-Bench tasks as well as on large-scale, high-resolution remote sensing benchmarks for segmentation and detection. As shown in Tab. 18 and Tab. 19, the frozen DINOv3 web model establishes new leading results Geo-Bench tasks as well as for segmentation and detection tasks on the LoveDA and DIOR datasets.

These findings have broader implications for the design of geospatial foundation models. Those have recently emphasized heuristic techniques such as multitemporal aggregation, multisensor fusion, or incorporating satellite-specific metadata ([Brown et al., 2025; Feng et al., 2025](#)). Our results show that general-purpose SSL can match or exceed satellite-specific approaches for tasks that depend on precise object boundaries (seg-

**Table 18:** Comparison of our DINOV3 models against strong baselines DOFA (Xiong et al., 2024), Prithvi-v2 (Szwarcman et al., 2024), and Tolan et al. (2024) in Geo-Bench tasks. While Prithvi-v2 and DOFA leverage all available optical bands, our models achieve significantly better performance with only RGB inputs.

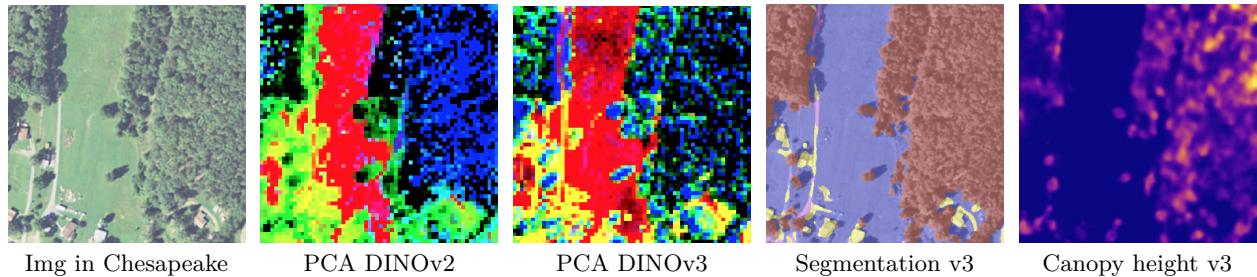
(a) Classification tasks.

Method	Arch.	FT	Bands	m-BENet	m-brick-kiln	m-eurosat	m-forestnet	m-pv4ger	m-so2sat	Mean
DOFA	ViT-L	🔥	all	68.7	98.4	96.6	55.7	98.2	61.6	79.9
Best of Prithvi-v2	ViT-L/H	🔥	all	71.2	<b>98.8</b>	96.4	54.1	98.1	59.1	79.6
Tolan et al. (2024)	ViT-L	✳️	RGB	66.0	97.1	95.2	56.3	94.3	58.1	77.8
DINOv3 Sat	ViT-L	✳️	RGB	73.0	96.5	94.1	60.6	96.0	57.4	79.6
DINOv3 Sat	7B	✳️	RGB	74.0	97.2	94.8	<b>62.3</b>	96.1	62.1	81.1
DINOv3 Web	7B	✳️	RGB	<b>74.6</b>	97.7	<b>97.0</b>	57.9	<b>98.3</b>	<b>63.8</b>	<b>81.6</b>

(b) Segmentation tasks.

Method	Arch.	FT	Bands	m-cashew*	m-chesapeake	m-NeonTree	m-nz-cattle	m-pv4ger-seg	m-SA-crop	Mean
DOFA	ViT-L	🔥	all	81.2	61.6	58.5	77.4	95.1	35.7	68.3
Best of Prithvi-v2	ViT-L/H	🔥	all	90.2	69.4	59.1	81.0	95.3	<b>41.9</b>	72.8
Tolan et al. (2024)	ViT-L	✳️	RGB	92.8	73.7	58.1	83.1	94.7	35.1	72.9
DINOv3 Sat	ViT-L	✳️	RGB	94.2	75.6	61.8	<b>83.7</b>	95.2	36.8	74.5
DINOv3 Sat	7B	✳️	RGB	94.1	<b>76.6</b>	62.6	83.4	95.5	37.6	75.0
DINOv3 Web	7B	✳️	RGB	<b>96.0</b>	76.5	<b>66.4</b>	<b>83.7</b>	<b>95.9</b>	36.8	<b>75.9</b>

\* Conversion to 6 classes following Szwarcman et al. (2024).



**Figure 18:** Illustration of versatile applications in remote sensing made possible by a single DINOV3 model. The PCA on DINOV3 features shows finer details than DINOV2. The segmentation map was computed using only GEO-Bench chesapeake labels. The canopy height model decoder was trained on the Open-Canopy dataset using 4 channels (RGB + InfraRed), while inference was performed on RGB channels only.

mentation or object detection). This supports emerging evidence finding that domain-agnostic pretraining can offer strong generalization even in specialized downstream domains (Lahrichi et al., 2025).

Collectively, our results suggest task-dependent benefits of domain-specific pretraining. The DINOV3 satellite model excels in metric tasks like depth estimation, leveraging satellite-specific priors. In contrast, the DINOV3 web model achieves state-of-the-art results on semantic geospatial tasks through diverse, universal representations. The complementary strengths of both models illustrate the broad applicability and effectiveness of the DINOV3 SSL paradigm.

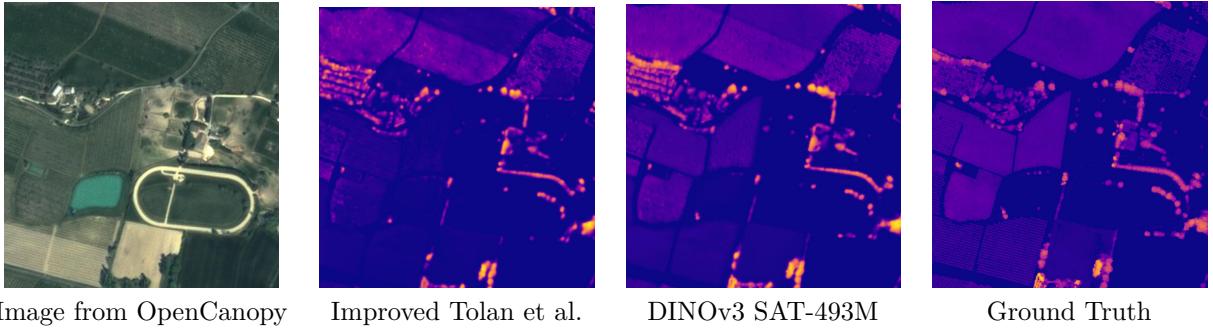
## 9 Environmental Impact

To estimate the carbon emission of our pre-training, we follow the methodology used in previous work in natural language processing (Strubell et al., 2019; Touvron et al., 2023) and SSL (Oquab et al., 2024). We fix the value of all exogenous variables, *i.e.* the Power Usage Effectiveness (PUE) and carbon intensity factor of a power grid to the same value as used by Touvron et al. (2023), *i.e.* we assume a PUE of 1.1 and a carbon intensity factor of the US average of 0.385 kg CO<sub>2</sub>eq/KWh. For the power consumption of GPUs, we take

**Table 19:** We compare the performance of DINOv3 to state-of-the-art models Privthi-v2 (Szwarcman et al., 2024), BillionFM (Cha et al., 2024) and SkySense V2 (Zhang et al., 2025) for high resolution semantic geospatial tasks. We report mIoU for the segmentation datasets LoveDA (1024×) and iSAID (896×), and mAP for the detection dataset DIOR (800×).

Method	Arch.	FT	LoveDA	iSAID	DIOR
Prev. SotA		🔥	BillionFM, ViT-G 54.4	SkySense V2, Swin-G* <b>71.9</b>	SkySense V2, Swin-G* 79.5
Decoder Arch.			UPerNet	UPerNet	Faster-RCNN
Privthi-v2	ViT-H	🔥	52.2	62.8	—
DINOv3 Sat	ViT-L	✳️	54.4	62.9	72.7
DINOv3 Sat	ViT-7B	✳️	55.3	64.8	76.6
DINOv3 Web	ViT-7B	✳️	<b>56.2</b>	71.4	<b>80.5</b>

\* Uses modified DINOv2 SSL with supervised pretraining alignment on OpenStreetMap, reporting +0.8 mIoU on iSAID.



**Figure 19:** A qualitative comparison of the DINOv3 7B satellite model to Tolan et al. (2024) on the Open Canopy dataset. For both models, the decoder is trained on 448×448 input images. It can be seen that DINOv3 produces more accurate maps, for example the accurate height for the trees on the field.

their thermal design power: 400W for A100 GPUs and 700W for H100 GPUs. We report the details of the computation for the pre-training of our ViT-7B in Tab. 20. For reference, we provide the analogous data for DINOv2 and MetaCLIP. As another point of comparison, the energy required to train one DINOv3 model (47 MWh) is roughly equivalent to that required for 240,000 km of driving with an average electric vehicle.

**Carbon Footprint of the Whole Project** In order to compute the carbon footprint of the whole project, we use a rough estimate of a total 9M GPU hours. Using the same grid parameters as presented above, we estimate the total footprint to be roughly 2600 tCO<sub>2</sub>eq. For comparison, a full Boeing 777 return flight between Paris and New York corresponds to approximately 560 tCO<sub>2</sub>eq. Supposing 12 such flights per day, the environmental impact of our project represents half of all flights between these two cities for one day. This estimate only considers the electricity for powering the GPUs and ignores other emissions, such as cooling, manufacturing, and disposal.

**Table 20:** Carbon footprint of model training. We report the potential carbon emission of reproducing a full model pre-training, computed using a PUE of 1.1 and carbon intensity factor of 0.385kg CO<sub>2</sub>eq/KWh.

Model	Arch.	GPU type	Power (W)	Steps	GPU hours	PUE	Total power (MWh)	Emission (tCO <sub>2</sub> eq)
MetaCLIP	ViT-G	A100-40GB	400W	390k	368,640	1.1	160	62
DINOv2	ViT-g	A100-40GB	400W	625k	22,016	1.1	9.7	3.7
DINOv3	ViT-7B	H100-SXM5	700W	1,000k	61,440	1.1	47	18

---

## 10 Conclusion

DINOv3 represents a significant advancement in the field of self-supervised learning, demonstrating the potential to revolutionize the way visual representations are learned across various domains. By scaling dataset and model size through meticulous data preparation, design, and optimization, DINOv3 showcases the power of self-supervised learning to eliminate the dependency on manual annotations. The introduction of the Gram anchoring method effectively mitigates the degradation of dense feature maps over extended training periods, ensuring robust and reliable performance.

Together with the implementation of post-hoc polishing strategies, such as high-resolution post-training and distillation, we achieve state-of-the-art performance across a wide range of visual tasks with no fine-tuning of the image encoder. The DINOv3 suite of vision models not only sets new benchmarks but also offers a versatile solution across various resource constraints, deployment scenarios, and application use cases. The progress made with DINOv3 is a testament to the promise of self-supervised learning in advancing the state of the art in computer vision and beyond.

---

## References

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Systematic outliers in large language models. *ICLR*, 2025.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

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. *arXiv preprint arXiv:2301.08243*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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. *arXiv preprint arXiv:2202.03555*, 2022.

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *ICML*, 2023.

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 *CVPR*, 2024.

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

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. *arXiv preprint arXiv:2210.01571*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *CoRR*, abs/2006.07159, 2020.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. In *ICCV*, Oct 2017.

Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017.

---

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

Christopher F. Brown, Michal R. Kazmierski, Valerie J. Pasquarella, William J. Ruckridge, Masha Samsikova, Chenhui Zhang, Evan Shelhamer, Estefania Lahera, Olivia Wiles, Simon Ilyushchenko, Noel Gorelick, Lihui Lydia Zhang, Sophia Alj, Emily Schechter, Sean Askay, Oliver Guinan, Rebecca Moore, Alexis Boukouvalas, and Pushmeet Kohli. Alphaearth foundations: An embedding field model for accurate and efficient global mapping from sparse label data, 2025.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.

Zhi Cai, Songtao Liu, Guodong Wang, Zheng Ge, Xiangyu Zhang, and Di Huang. Align-detr: Enhancing end-to-end object detection with aligned loss, 2024.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Keumgang Cha, Junghoon Seo, and Taekyung Lee. A billion-scale foundation model for remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, page 1–17, 2024. ISSN 2151-1535. doi: 10.1109/jstars.2024.3401772.

François Charton and Julia Kempe. Emergent properties with repeated examples, 2024.

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Bowen Chen, Andrew Zhang, Daniel Shao, Andrew H Song, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *preprint arXiv:2002.05709*, 2020a.

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, 2020b.

---

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *preprint arXiv:2011.10566*, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

Yinjie Chen, Zipeng Yan, Chong Zhou, Bo Dai, and Andrew F Luo. Vision transformers with self-distilled registers. *arXiv preprint arXiv:2505.21501*, 2025.

Zhe Chen, Yuchen Duan, Wenhui Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023.

Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *NeurIPS*, 2022.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latent patches for improved masked image modeling. *TMLR*, 2025.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim M. Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin F. Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023.

---

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *preprint arXiv:1810.04805*, 2018.

Barry M Dillon, Gregor Kasieczka, Hans Olischlager, Tilman Plehn, Peter Sorrenson, and Lorenz Vogel. Symmetries, safety, and self-supervision. *SciPost Physics*, 12(6):188, 2022.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. In *CVPR*, 2023.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 2016.

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. *preprint arXiv:2010.11929*, 2020.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *ICML*, 2023.

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021.

Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes challenge 2007 (VOC2007) results, 2007.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, 2012.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

David Fan, Shengbang Tong, Jiachen Zhu, Koustuv Sinha, Zhuang Liu, Xinlei Chen, Michael Rabbat, Nicolas Ballas, Yann LeCun, Amir Bar, et al. Scaling language-free visual representation learning. *arXiv preprint arXiv:2504.01017*, 2025.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024a.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024b.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

---

Zhengpeng Feng, Clement Atzberger, Sadiq Jaffer, Jovana Knezevic, Silja Sormunen, Robin Young, Madeleine C Lisaius, Markus Immittzer, David A. Coomes, Anil Madhavapeddy, Andrew Blake, and Srinivasan Keshav. TESSERA: Temporal embeddings of surface spectra for earth representation and analysis, 2025.

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, Alexander T Toshev, Marcin Eichner, Moin Nabi, Yinfai Yang, Joshua M. Susskind, and Alaaeldin El-Nouby. Multimodal autoregressive pre-training of large vision encoders. *arXiv preprint arXiv:2411.14402*, 2024.

Fajwel Fogel, Yohann Perron, Nikola Besic, Laurent Saint-André, Agnès Pellissier-Tanon, Martin Schwartz, Thomas Boudras, Ibrahim Fayad, Alexandre d'Aspremont, Loic Landrieu, et al. Open-canopy: Towards very high resolution forest monitoring. In *CVPR*, 2025.

Stephanie Fu, Mark Hamilton, Laura E Brandt, Axel Feldmann, Zhoutong Zhang, and William T Freeman. Featup: A model-agnostic framework for features at any resolution. In *ICLR*, 2024.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018.

Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019.

Priya Goyal, Mathilde Caron, Benjamin Lefauze, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *preprint arXiv:2103.01988*, 2021.

Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022a.

Priya Goyal, Adriana Romero Soriano, Caner Hazirbas, Levent Sagun, and Nicolas Usunier. Fairness indicators for systematic assessments of visual feature extractors. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 70–88, 2022b.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022.

Nancy Harris, David Gibbs, A. Baccini, Richard Birdsey, Sytze de Bruin, Mary Farina, Lola Fatoyinbo, Matthew Hansen, Martin Herold, Richard Houghton, Peter Potapov, Daniela Requena Suarez, Rosa Maria Roman-Cuesta, Sassan Saatchi, Christy Slay, Svetlana Turubanova, and Alexandra Tyukavina. Global maps of twenty-first century forest carbon fluxes. *Nature Climate Change*, 11:1–7, 03 2021. doi: 10.1038/s41558-020-00976-6.

---

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Greg Heinrich, Mike Ranzinger, Hongxu, Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. RADIov2.5: Improved baselines for agglomerative vision foundation models. *arXiv preprint arXiv:2412.07679*, 2025.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *preprint arXiv:1905.09272*, 2019.

Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. *arXiv preprint arXiv:2103.10957*, 2021.

Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. In *ECCV*, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021b.

Byeongho Heo, Song Park, Dongyo Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020.

Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, Andre Araujo, Ricardo Martin-Brualla, Kaushal Patel, Daniel Vlasic, Vittorio Ferrari, Ameesh Makadia, Ce Liu, Yuanzhen Li, and Howard Zhou. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. In *NeurIPS*, 2023.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *CVPR*, June 2014.

Chao Jia, Yinfai Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *arXiv preprint arXiv:2506.08010*, 2025.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcret, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. DINOv2 meets text: A unified framework for image-and pixel-level vision-language alignment. In *CVPR*, 2025.

---

Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *arXiv preprint arXiv:2505.09358*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamchetti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Vladislav Kim, Nikolaos Adaloglou, Marc Osterland, Flavio M Morelli, Marah Halawa, Tim König, David Gnutt, and Paula A Marin Zapata. Self-supervision advances morphological profiling by unlocking powerful image representations. *Scientific Reports*, 15(1):4876, 2025.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *CVPR*, 2023.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, pages 491–507. Springer, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

Alexandre Lacoste, Nils Lehmann, Pau Rodriguez, Evan Sherwin, Hannah Kerner, Björn Lütjens, Jeremy Irvin, David Dao, Hamed Alemohammad, Alexandre Drouin, et al. Geo-bench: Toward foundation models for earth monitoring. *NeurIPS*, 2023.

Saad Lahrichi, Zion Sheng, Shufan Xia, Kyle Bradbury, and Jordan Malof. Is self-supervised pre-training on satellite imagery better than imagenet? a systematic study with sentinel-2, 2025.

Yann LeCun. A path towards autonomous machine intelligence. *openreview*, 2022.

Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Learning visual n-grams from web data. In *ICCV*, 2017.

Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159: 296–307, 2020.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2023a.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

---

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

Yutong Lin, Yuhui Yuan, Zheng Zhang, Chen Li, Nanning Zheng, and Han Hu. Detr does not need multi-scale or locality design. In *ICCV*, 2023b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Xiaofeng Mao, Yuefeng Chen, Yao Zhu, Da Chen, Hang Su, Rong Zhang, and Hui Xue. COCO-O: A benchmark for object detectors under natural distribution shifts. In *ICCV*, 2023.

Tomáš Mikolov, Martin Karafiat, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. SPair-71k: A large-scale benchmark for semantic correspondence, 2019.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 69–84, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46466-4.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

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. *TMLR*, 2024.

Valentinos Pariza, Mohammadreza Salehi, Gertjan J Burghouts, Francesco Locatello, and Yuki M Asano. Near, far: Patch-ordering enhances vision foundation models’ scene understanding. In *ICLR*, 2025.

Liam Parker, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Rudy Morel, et al. Astroclip: a cross-modal foundation model for galaxies. *Monthly Notices of the Royal Astronomical Society*, 531(4):4990–5011, 2024.

---

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

Fernando Pérez-García, Harshita Sharma, Sam Bond-Taylor, Kenza Bouzid, Valentina Salvatelli, Maximilian Ilse, Shruthi Bannur, Daniel C Castro, Anton Schwaighofer, Matthew P Lungren, et al. Exploring scalable medical image encoders beyond text supervision. *Nature Machine Intelligence*, pages 1–12, 2025.

Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. In *NeurIPS*, 2020.

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

Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos. *arXiv preprint arXiv:2501.05453*, 2025.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 44(3):1623–1637, 2020.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *CVPR*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *ICLR*, 2025.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pages 5389–5400, 2019.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Hamid Rezatofighi, Nathan Tsoi, Jun Young Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 2019.

---

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Bryan Russell, William Freeman, Alexei Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Mohammadreza Salehi, Efstratios Gavves, Cees G. M. Snoek, and Yuki M. Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *ICCV*, 2023.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

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.

Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

Oriane Siméoni, Éloi Zablocki, Spyros Gidaris, Gilles Puy, and Patrick Pérez. Unsupervised object localization in the era of self-supervised vits: A survey. *IJCV*, 133(2):781–808, 2025.

Walter Simoncini, Andrei Bursuc, Spyridon Gidaris, and Yuki Asano. No train, all gain: Self-supervised gradients improve deep frozen representations. *NeurIPS*, 2024.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting Weakly Supervised Pre-Training of Visual Perception Models. In *CVPR*, 2022.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

---

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. EVA-CLIP-18B: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, Thorsteinn Elí Gíslason, Benedikt Blumenstiel, Rinki Ghosal, Pedro Henrique de Oliveira, Joao Lucas de Sousa Almeida, Rocco Sedona, Yanghui Kang, et al. Prithvi-eo-2.0: A versatile multi-temporal foundation model for earth observation applications. *arXiv preprint arXiv:2412.02732*, 2024.

Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *preprint arXiv:1905.11946*, 2019.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

Yonglong Tian, Olivier J Henaff, and Aäron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *ICCV*, 2021.

Jamie Tolan, Hung-I Yang, Benjamin Nosarzewski, Guillaume Couairon, Huy V Vo, John Brandt, Justine Spore, Sayantan Majumdar, Daniel Haziza, Janaki Vamaraju, et al. Very high resolution canopy height maps from rgb imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar. *Remote Sensing of Environment*, 300:113888, 2024.

Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *NeurIPS*, 2019.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *preprint arXiv:2012.12877*, 2020.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmen, 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.

Tsung-Yi, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays Georgia, Pietro Perona, Deva Ramanan, Larry Zitnick, and Piotr Dollár. COCO 2017: Common Objects in Context, 2017.

---

Tinne Tuytelaars, Christoph Lampert, Matthew Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. *IJCV*, 2010.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12884–12893, 2021.

Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Bassart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *ArXiv 1908.00463*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Shashanka Venkataramanan, Valentinos Pariza, Mohammadreza Salehi, Lukas Knobel, Spyros Gidaris, Elias Ramzi, Andrei Bursuc, and Yuki M. Asano. Franca: Nested matryoshka clustering for scalable visual representation learning. *arXiv preprint arXiv:2507.14137*, 2025.

Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019.

Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Huy V. Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2021.

Huy V. Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprise, Maxime Oquab, Armand Joulin, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Automatic data curation for self-supervised learning: A clustering-based approach. *TMLR*, 2024.

Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature medicine*, 30(10):2924–2935, 2024.

Di Wang, Jing Zhang, Minqiang Xu, Lin Liu, Dongsheng Wang, Erzhong Gao, Chengxi Han, Haonan Guo, Bo Du, Dacheng Tao, and Liangpei Zhang. Mtp: Advancing remote sensing foundation model via multi-task pretraining, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025.

Junjue Wang, Zhuo Zheng, Ailong Ma, Xiaoyan Lu, and Yanfei Zhong. Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation, 2022a.

Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023a.

Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023b.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhajit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022b.

---

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

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 *CVPR*, pages 14543–14553, 2022c.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE TPAMI*, 45(12):15790–15801, 2023c.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. In *ECCV*, 2024b.

Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks. *ECCV*, 2024.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, June 2010. doi: 10.1109/CVPR.2010.5539970.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding, 2018.

Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. In *NeurIPS*, 2021.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Zhitong Xiong, Yi Wang, Fahong Zhang, Adam J Stewart, Joëlle Hanna, Damian Borth, Ioannis Papoutsis, Bertrand Le Saux, Gustau Camps-Valls, and Xiao Xiang Zhu. Neural plasticity-inspired foundation model for observing the Earth crossing modalities. *arXiv preprint arXiv:2403.15356*, 2024.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *ICLR*, 2024.

Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-VOS: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 585–601, 2018.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024a.

---

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 2024b.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *CVPR*, 2025.

Burak Yildiz, Seyran Khademi, Ronald Maria Siebes, and Jan van Gemert. Amstertime: A visual place recognition benchmark dataset for severe domain shift. *arXiv preprint arXiv:2203.16291*, 2022.

Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *CVPR*, 2021.

Yongseon Yoo, Seonggyu Kim, and Jong-Min Lee. Sagagan: Style applied using Gram matrix attribution based on stargan v2. In *BMVC*, 2024.

Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahim, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 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. In *ICLR*, 2025.

Sukmin Yun, Hankook Lee, Jaehyung Kim, and Jinwoo Shin. Patch-level representation learning for self-supervised vision transformers. In *CVPR*, 2022.

Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images, 2019.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022a.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18123–18133, 2022b.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

Yingying Zhang, Lixiang Ru, Kang Wu, Lei Yu, Lei Liang, Yansheng Li, and Jingdong Chen. SkySense V2: A Unified Foundation Model for Multi-modal Remote Sensing. *arXiv preprint arXiv:2507.13812*, 2025.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NeurIPS*, 2014.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

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.

---

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018.

Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, 2022.

Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *preprint arXiv:2006.06882*, 2020.

---

# Appendix

## A Artifacts and Outliers in Large-Scale Training

This section provides a discussion about the emergence of artifacts and outliers that has recently been observed in the training of large models in both the LLM (An et al., 2025) and the visual domains (Darcet et al., 2024). Borrowing the definition from An et al. (2025), outliers are typically characterized as network’s activations whose values deviate significantly from the average of their distribution. During the training of DINOv3, we identified such outliers at different levels: some occurring at the patch level and others at the feature dimension level. We discuss bellow the different types of outlier observed, their impact on the training and results. We also discuss our different attempts at fixing them and our first conclusions.

### A.1 High-Norm Patch Outliers

Darcet et al. (2024) discovered that patch outliers negatively affect performance in DINOv2. These outliers are primarily characterized as high-norm tokens, often located in low-information background regions of an image. These tokens are observed to play a key role in the internal communication between patches and the CLS token. Additionally, this phenomenon affects other models as well, whether trained with supervision or not, such as CLIP (Radford et al., 2021). When scaling to a 7B model, we observe the emergence of such high-norm patches, predominantly in the background area. In this section, we present results from 7B models trained for 150k iterations, which, although limited, provide us with initial signals to guide our decisions. We plot the output patch norms (before the layer norm) in Fig. 20a, in the column ‘ $\emptyset$ ’, with high-norm patches in yellow appearing in the sky and other low-information areas.

**Token Registers** In order to mitigate the appearance of such token outliers, (Darcet et al., 2024) proposes a simple yet effective solution: introducing additional tokens, called registers, into the input sequence of the ViT. Their role is to take over the internal communication between patches and the CLS. Following the conclusions, we use 4 registers and do not ablate further due to the high experimental cost. Figure 20a illustrates examples of this strategy in action, where we observe the elimination of high-norm outliers, as further confirmed by the corresponding histogram of the norm distribution. Moreover, we quantitatively observe in Fig. 20b the benefit of incorporating additional register tokens on the ImageNet-1k (IN1k) benchmark.

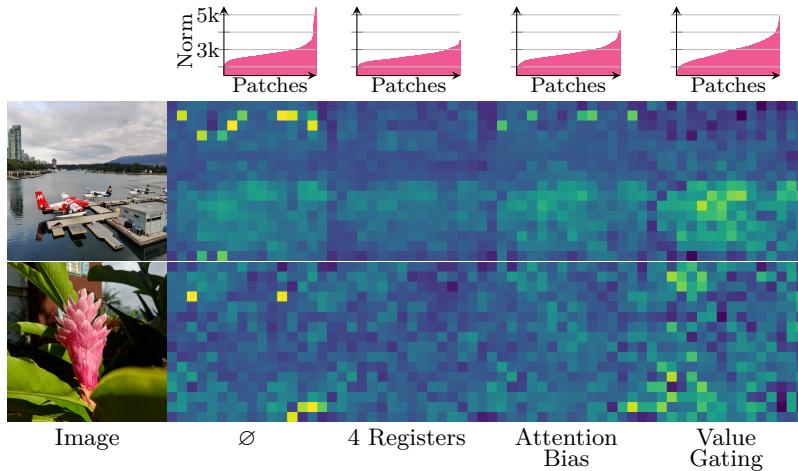
**Integrating Biases in the Attention Mechanism** Recent work by An et al. (2025) investigates the appearance of outliers in the LLM realm across different models and architectures. The authors analyze different types of outliers which are observed to be intrinsically linked to the attention mechanism. They propose to mitigate the problem with several solutions from which we select two promising solutions which seem relevant and require minimal changes to the attention, specifically the explicit fixed bias, which we call ‘value gating’, and the attention bias strategies. The value gating strategy amounts to adding a learnable value bias  $\mathbf{v}' \in \mathbb{R}^d$  to the output of the attention, specifically by redefining the attention mechanism as

$$\text{Attn}(Q, K, V; \mathbf{k}', \mathbf{v}') = \text{softmax}\left(\frac{Q[K^T]}{\sqrt{d}}\right)V + \mathbf{v}', \quad (4)$$

with  $Q, K, V \in \mathbb{R}^{T \times d}$  the query, key, and value matrices and  $d$  the dimensionality of the hidden space. Alternatively, the attention bias, defined in Eq. 5, consists in integrating two learnable bias terms  $\mathbf{k}', \mathbf{v}' \in \mathbb{R}^d$  in the keys and values matrices, respectively. It is defined as follows:

$$\text{Attn}(Q, K, V; \mathbf{k}', \mathbf{v}') = \text{softmax}\left(\frac{Q[K^T \mathbf{k}']}{\sqrt{d}}\right) \begin{bmatrix} V \\ \mathbf{v}' \end{bmatrix}. \quad (5)$$

We observe in Fig. 20a that the value gating strategy substantially modifies the distribution of patch norms, resulting in generally higher norm values and the elimination of clear outliers. While the attention mechanism mitigates the presence of high-norm tokens, it does not completely resolve the issue, as some high-norm patches persist—as visible in the top row image—when compared to our results using register tokens. Notably, the best performance is achieved with the incorporation of the register tokens, which is why we adopt this strategy for all experiments reported in the paper.



(a) Visualization of patch norms by outlier strategy. The bottom two rows share a colormap per row, from dark blue (low) to yellow (high).

Outlier Strategy	IN1k (Linear)	ADE20k mIoU
∅	86.4	<b>53.2</b>
4 Registers	<b>86.6</b>	53.0
Attention Bias	86.5	52.7
Value Gating	86.3	52.2

**Figure 20:** Impact of the different strategies to mitigate the presence of high-norm patch outliers, evaluated both (a) qualitatively and (b) quantitatively. We produce results with a 7B model trained with our recipe for 150k iterations, without any high-norm handling strategy ‘∅’, when using four register tokens (Darcet et al., 2024), or the attention bias and value gating strategies (An et al., 2025). In (a, first row), we plot the distribution of the output patch norms (sorted by ascending values) computed for three images. We also visualize the output patch norms per image (bottom two rows), with the same colormap–min and max values are computed per image over the different outlier strategies.

## A.2 Feature Dimension Outliers

The introduction of additional registers into the model architecture effectively resolves the issue of high-norm patch outliers. However, during the training of 7B models, we observe a distinct type of outlier that emerges not across patches, but within the feature (channel) dimension of the learned representations. Specifically, analysis of patch activations across transformer layers and training iterations reveals that a small subset of feature dimensions attain exceptionally large magnitudes, even as the norms across patches remain stable. Interestingly, these feature dimension outliers exhibit consistently high values across different patches and images, a behavior that contrasts with observations reported in (An et al., 2025). Moreover, these outlier dimensions consistently persist across the layers of a given model, increasing in magnitude with depth and reaching their maximum values in the output layer. They also progressively increase in magnitude throughout the course of training.

We conduct experiments attempting to neutralize these dimensions during both training and inference. Our findings indicate that these dimensions play a significant role during training, as applying L2-regularization to suppress them results in a performance drop. However, removing these dimensions at inference time does not lead to significant performance changes, suggesting that they primarily carry trivial or non-informative signals. Additionally, we observe that the final layer normalization is trained to substantially scale down these outlier dimensions. Thus, we recommend to apply the final layer norm to the features of the final layer for downstream use. Alternatively, applying batch normalization can also suppress these feature dimension outliers, as their elevated values are consistent across patches and images.

A word of caution applies to using features from earlier layers. As discussed above, these earlier layers are also affected by feature dimension outliers which can lead to ill-conditioned features. While the final layer normalization is well-suited to normalize the distribution of the final features, its learned parameters may be suboptimal for applying it to the features of earlier layers. Indeed, we observe performance decreases for some tasks from doing so. In these cases, we found standard feature scaling techniques (*e.g.* normalization with batch norm or principal component analysis) to be effective in dealing with the feature dimension outliers.

**Table 21:** Details of year of publication, performance, and reference of the numbers used in Fig. 1. For all papers, we report top-1 accuracy of this algorithm with the largest model on ImageNet. For weakly- and self-supervised models, we provide linear probing performance. For dates, we use the year of first appearance on arXiv.

Year	Supervised		Weakly-Supervised		Self-Supervised	
	Top-1	Reference	Top-1	Reference	Top-1	Reference
2012	59.3	Krizhevsky et al. (2012)				
2013						
2014						
2015	78.6	He et al. (2016)	34.9	Joulin et al. (2016)		
2016						
2017	80.9	Xie et al. (2017)				
2018			83.6	Mahajan et al. (2018)	38.2	Caron et al. (2018)
2019	84.3	Tan and Le (2019)			68.6	He et al. (2020)
2020	87.5	Kolesnikov et al. (2020)			75.3	Caron et al. (2020)
2021	88.6	Dosovitskiy et al. (2020)	88.4	Radford et al. (2021)	82.3	Zhou et al. (2021)
2022						
2023	89.5	Dehghani et al. (2023)			86.5	Quab et al. (2024)
2024						
2025			89.3	Bolya et al. (2025)	88.4	This work

For example, for our semantic segmentation (Sec. 6.3.2) and depth estimation experiments (Sec. 6.3.3) using features from intermediate layers, we apply batch normalization.

## B Additional Results

### B.1 Evolution Over Years

In Fig. 1, we provide a rough evolution of state-of-the-art performance along years. Here, we provide the precise references and performances that we reported in the figure. Please find it in Tab. 21.

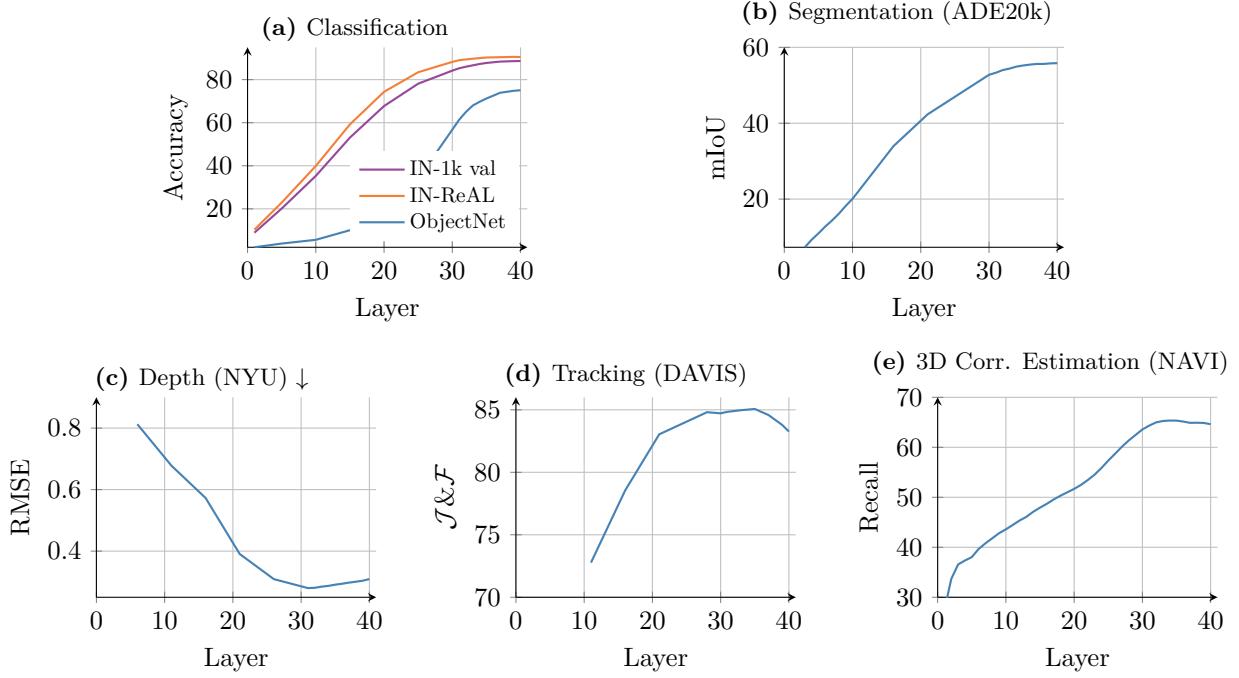
### B.2 Per-Layer Analysis

In this section, we evaluate the quality of our features across the various layers of the DINOv3 7B model. Specifically, we present results from five representative tasks: classification (IN-1k val, ImageNet-REAL and ObjectNet), segmentation (ADE20k), depth estimation (NYU), tracking (DAVIS), and 3D correspondence estimation (NAVI). For the first 3 benchmarks, a linear layer is trained on the outputs of each backbone layer to assess feature performance as in Secs. 6.1.2 and 6.2.1. For tracking and correspondence estimation, we use non-parametric approaches as in Secs. 6.1.3 and 6.1.5.

The results are shown in Fig. 21. We find that for classification and dense tasks, performance increases smoothly over the layers. Depth estimation, tracking, and 3D correspondence estimation peak around layer 32, indicating that, for tasks where geometry plays a significant role, the downstream performance of DINOv3 can be improved by considering earlier layers. On the other hand, the performance of intermediate layers only slightly improves compared to the last one, making it a good default choice.

### B.3 Additional Results to Main Results Section

We give additional experimental results complementing the main results in Sec. 6. In Tab. 22, we show per-dataset results for finegrained classification on small datasets with linear probing (Fine-S, see Sec. 6.2.1). In Tab. 23, we give full results for the instance recognition evaluation (Sec. 6.2.2), adding more metrics.



**Figure 21:** Results on five benchmarks using features from intermediate layers of DINOv3 7B. Evaluations (a-c) use a linear layer (see Secs. 6.1.2 and 6.2.1), while (d, e) use a non-parametric approach (see Secs. 6.1.3 and 6.1.5).

**Table 22:** Per-dataset results for finegrained classification on small datasets with linear probing (Fine-S, see Sec. 6.2.1), following Oquab et al. (2024).

Method	ViT	Food	C10	C100	SUN	Cars	Aircr.	VOC	DTD	Pets	Cal101	Flowers	CUB	Avg
<i>Agglomerative backbones</i>														
AM-RADIOv2.5	g/14	96.5	99.5	95.0	82.8	95.4	91.7	90.3	88.6	96.7	98.8	99.7	91.5	93.9
<i>Weakly-supervised backbones</i>														
SigLIP	g/16	97.7	99.3	92.7	85.1	96.5	88.7	91.0	87.7	98.7	90.3	99.7	90.3	93.7
PE-core	G/14	97.8	99.5	95.3	85.2	96.5	92.0	90.5	88.2	98.7	93.3	99.5	93.3	94.5
AIMv2	3B/14	96.6	99.3	93.3	83.4	95.6	84.2	90.5	87.4	96.8	90.7	99.7	90.7	92.9
EVA CLIP	18B/14	96.9	99.5	95.4	85.0	95.4	81.6	90.2	87.1	98.4	90.6	99.6	90.6	92.9
<i>Self-supervised backbones</i>														
Franca	g/14	89.2	98.6	90.4	73.7	89.7	74.1	89.4	80.6	93.2	97.6	97.8	78.4	87.7
DINOv2	g/14	95.6	99.5	94.5	78.9	94.6	88.5	88.4	86.8	96.8	95.9	99.7	91.6	92.6
Web-DINO	7B/14	96.1	99.5	93.4	77.5	95.0	88.8	87.0	79.9	92.9	93.1	99.6	78.9	90.2
DINOv3	7B/16	96.9	99.6	96.0	81.1	95.0	88.2	88.2	87.2	97.0	94.8	99.7	92.4	93.0

Finally, in Tab. 24, we give complementary results for our state-of-the-art semantic segmentation model (Sec. 6.3.2) on the COCO-Stuff (Caesar et al., 2018), PASCAL VOC 2012 (Everingham et al., 2012) and Cityscapes (Geiger et al., 2013) datasets.

#### B.4 Classification on OCR-Heavy Datasets

In this experiment, we evaluate DINOv3 on classification tasks that require some form of character recognition. These tasks include street-sign, logo, and product classification. We compare our model to the best self-supervised model (DINOv2 g) and the best weakly-supervised one (PE-core G). We run this evaluation on images of resolution 512 for our model, and adjust for patch size to match the sequence length for the others. We report the result of this experiment in Tab. 25.

**Table 23:** Full results for instance recognition, presenting additional metrics for Sec. 6.2.2.

Method	ViT	Oxford		Paris		Met			AmsterTime
		M	H	M	H	GAP	GAP-	ACC	mAP
<i>Agglomerative backbones</i>									
AM-RADIOv2.5	g/16	72.8	50.7	93.3	85.3	30.5	65.9	69.0	46.7
<i>Weakly-supervised backbones</i>									
SigLIPv2	g/16	49.3	25.1	79.3	60.9	0.0	0.0	0.2	15.5
PE-core	G/14	57.4	32.7	83.6	68.9	10.6	34.8	44.9	23.1
AIMv2	3B/14	55.0	28.8	85.6	71.4	29.5	67.3	69.9	23.1
EvaCLIP	18B/14	55.2	27.1	81.8	65.6	0.5	4.3	11.0	18.9
<i>Self-supervised backbones</i>									
Franca	g/14	44.6	14.3	73.8	51.6	27.2	54.3	57.7	21.1
DINOv2	g/14	78.2	58.2	92.7	84.6	44.6	73.0	75.2	48.9
Web-DINO	7B/14	64.1	31.2	89.8	80.3	35.2	67.3	71.3	30.6
DINOv3	7B/16	81.1	60.7	93.3	87.1	55.4	77.7	80.7	56.5

We see that our new model DINOv3 drastically outperforms its predecessor DINOv2. However, the gap with weakly-supervised models remains large. Since our model does not leverage pair image-text data during training, it has a much harder time learning glyph associations. Recent work from [Fan et al. \(2025\)](#) hints at the impact of training data on the performance on this type of tasks. Since the main focus of our work is on improving dense features, we leave closing this gap for future work.

## B.5 Fairness Analysis

We evaluate the geographical fairness and diversity of the DINOv3 7B model across different income buckets and regions, following the protocol of [Goyal et al. \(2022b\)](#). For reference, we include the results obtained with DINOv2 and SEERv2. The results indicate that DINOv3 delivers somewhat consistent performance across income categories, although there is a notable performance drop of 23% in the low-income bucket compared to the highest-income bucket. The medium and high-income buckets exhibit comparable performance. Regionally, DINOv3 achieves relatively good scores across different regions; however, a relative difference of over 14% is observed between Europe and Africa, which is an improvement over the relative difference of more than 17% seen with DINOv2.

## C Implementation Details

We use multi-crop ([Caron et al., 2020](#)) with 2 global crops ( $256 \times 256$  px) and 8 local crops ( $112 \times 112$  px) seen by the student model, resulting in a total sequence length of 3.7M tokens. The teacher EMA (exponential

**Table 24:** Comparison with state-of-the-art systems on semantic segmentation on other datasets, complementary to the ADE20k results in Tab. 11. We report mIoU scores when evaluating the model in a single- or multi-scale (TTA) setup and compare against the best previously published result for each dataset: [Fang et al. \(2023\)](#) for COCO-Stuff 164k, [Wang et al. \(2023b\)](#) for Cityscapes, and [Zoph et al. \(2020\)](#) for VOC 2012. We use input resolutions of 1280 for COCO-Stuff, 1024 for VOC 2012, and 1280 for Cityscapes. All baselines require finetuning of the backbone, while we keep the DINOv3 backbone frozen.

Method	FT	COCO-Stuff 164k		Cityscapes		VOC 2012	
		Single	TTA	Single	TTA	Single	TTA
Previous Best	🔥	53.7	53.7	<b>86.3</b>	<b>87.0</b>	—	90.0
DINOv3	✳️	<b>53.8</b>	<b>54.0</b>	86.1	86.7	<b>90.1</b>	<b>90.4</b>

**Table 25:** Comparison of DINOv3 classification performance on OCR-heavy datasets. These are notoriously hard datasets for SSL. We compare DINOv3 with the best DINOv2 model (g), along with the best weakly-supervised PE-core model (G).

Model		GTSRB	Logo-2K+	FlickrLogos-32	RP2K	Products-10K	SOPProducts
DINOv2	ViT-g	78.2	52.9	83.6	91.4	70.8	57.6
PE-core	ViT-G	94.8	93.2	99.0	93.1	80.6	80.7
DINOv3-7B	ViT-7B	87.5	86.0	86.3	94.7	74.5	65.2

**Table 26:** Geographical fairness and diversity analysis across income buckets and regions, following the protocol of [Goyal et al. \(2022b\)](#).

Method	Arch.	Income Buckets			Regions			
		low	medium	high	Africa	Asia	Americas	Europe
SEERv2	RG-10B	59.7	78.5	86.6	65.9	76.3	81.1	85.6
DINOv2	ViT-g/14	67.4	83.3	90.5	74.0	81.6	86.2	89.7
DINOv3	ViT-7B	69.6	85.7	90.9	76.7	83.0	88.0	90.7

moving average of the student) processes the global crops only. We apply the  $\mathcal{L}_{\text{DINO}}$  loss on the class token of all student local crops and both teacher global crops, and between pairs of different global crops for both models. A random proportion in  $[0.1, 0.5]$  of the global crops patch tokens seen by the student are masked with 50% probability, and we apply the  $\mathcal{L}_{\text{iBOT}}$  loss between these and the visible tokens seen by the teacher EMA. We apply the  $\mathcal{L}_{\text{DKoleo}}$  loss to small batches of 16 class tokens of the first global crop seen by the student. We train for 1M iterations using a fully-sharded data-parallel setup in Pytorch, using bfloat16 and 8-bit floating-point matrix multiplications. We use a constant learning rate of 0.00004 with a warmup of 100k iterations, a weight decay of 0.04, a learning rate decay factor of 0.98 per layer, a stochastic depth (layer dropout) value of 0.4 and an EMA factor of 0.999 for the teacher. Remaining hyperparameters can be found in the configuration files in the code release.

For the Gram anchoring step, we use a loss weight of  $w_{\text{Gram}} = 2$  and update the Gram teacher every 10k steps for a maximum of three updates. For high-resolution adaptation (Sec. 5.1), we sample from the following pairs of global/local/Gram teacher crop resolutions with the following probabilities: (512, 112, 768) with  $p = 0.3$ , (768, 112, 1152) with  $p = 0.3$ , (768, 168, 1152) with  $p = 0.3$ , (768, 224, 1152) with  $p = 0.05$ , and (768, 336, 1152) with  $p = 0.05$ . These values were obtained empirically.

## D Experimental Details

In this section, we provide detailed descriptions of the datasets and evaluation metrics used across all benchmarks in this paper.

### D.1 Semantic Segmentation: Linear Probing

**Datasets and Metrics** We evaluate semantic segmentation performance of DINOv3 obtained via linear probing on three benchmark datasets: ADE20k ([Zhou et al., 2017](#)), VOC12 ([Everingham et al., 2012](#)), and Cityscapes ([Cordts et al., 2016](#)). The evaluation metric reported is the standard mean Intersection-over-Union (mIoU).

**Evaluation Protocol** To assess the quality of the dense features, we train a linear classifier on the training set of each benchmark. This linear layer is applied on top of the patch output features (after layer normalization) of the frozen backbone, with the features further normalized using a trained batch normalization layer. For all backbones, we perform a hyperparameter sweep using the AdamW optimizer, varying the learning rate over  $\{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}\}$  and weight decay over  $\{1 \times 10^{-4}, 1 \times 10^{-3}\}$ .

---

## D.2 Depth Estimation: Linear Probing

**Datasets and Metrics** We evaluate the quality of DINOv3 features for geometric tasks on the depth benchmarks NYUV2 (Silberman et al., 2012) and KITTI (Geiger et al., 2013) datasets. Results are reported using the Root Mean Squared Error (RMSE) metric.

**Evaluation Protocol** To assess the quality of the dense features, we train a linear classifier on the training set of each benchmark. This linear layer is applied on top of the patch output features (after layer normalization) of the frozen backbone, with the features further normalized using a trained batch normalization layer. For all backbones, we perform a hyperparameter sweep using the AdamW optimizer, varying the learning rate over  $[1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}]$  and weight decay over  $[1 \times 10^{-4}, 1 \times 10^{-3}]$ .

## D.3 3D Keypoint Matching

**Datasets and Metrics** Geometric correspondence is evaluated on the NAVI dataset (Jampani et al., 2023), and semantic correspondence on the SPair dataset (Min et al., 2019). For NAVI, we use images resized to a side length of 448/512 pixels for models with patch size 14/16. For SPAir, we use images resized to a side length of 896/1024 pixels for models with patch size 14/16. To measure performance, we report the correspondence recall, *i.e.* the percentage of correspondences falling into a specified distance.

**Evaluation Protocol** For NAVI, we follow the protocol defined in Probe3D (Banani et al., 2024). Specifically, we subsample 1/4 of the object views, and for each source view select another dest. view within a maximum rotation of 120 degrees to create an image pair (source, dest.) to perform patch matching on. For each image pair, each patch of source (within the object) is matched to a patch in dest. The top-1000 matches with highest cosine similarity are kept for evaluation, and a 3D distance error is computed for each match based on the known camera pose and depth maps of both images. This allows to compute recall errors with varying thresholds, for which we use thresholds of 1cm, 2cm, and 5cm. We then compute the average recall across thresholds as the correspondence recall.

For each evaluated backbone, we use the features of the final layer, and evaluate them with and without the final layer norm applied. This is because we noticed bad performance for some models when applying the final layer norm. We report the maximum of both results.

## D.4 Unsupervised Object Discovery

**Datasets and Metrics** For this task, the objective is to generate a single bounding box per image that highlights any object depicted in the scene. We follow the protocol of Siméoni et al. (2021) for unsupervised object discovery and evaluate all backbones on the detection benchmarks VOC07 (Everingham et al., 2007), VOC12 (Everingham et al., 2012), and COCO20K (Lin et al., 2014; Vo et al., 2020). COCO20K is a subset of the COCO2014 trainval dataset (Lin et al., 2014), consisting of 19,817 randomly selected images, as proposed in (Vo et al., 2020) and commonly used for this task. For each image, a single bounding box is generated. For evaluation, we use the *Correct Localization* (CorLoc) metric, which computes the percentage of correctly localized boxes. A predicted box is considered correct if its *intersection over union* (IoU) with any ground-truth bounding box exceeds 0.5.

**Evaluation Protocol** To evaluate the quality of the image encoders, we employ the TokenCut strategy (Wang et al., 2023c). This method organizes image patches into a fully connected graph, where the edges represent similarity scores between pairs of patches, computed using backbone features learned by the transformer. The salient object patches are identified by applying the Normalized Cut algorithm, which solves a graph-cut problem. A bounding box is then fitted to the resulting salient object mask. All images are input to the encoders at their full resolution, and we use the patch output features for all image encoder. To account for differences in feature distributions among models, we perform a sweep over TokenCut’s unique hyperparameter: the similarity threshold used in graph construction. Specifically, we vary the threshold between 0 and 0.4 in increments of 0.05.

---

## D.5 Video Segmentation Tracking

**Datasets and Metrics** For this task, we use the DAVIS 2017 (Pont-Tuset et al., 2017), YouTube-VOS (Xu et al., 2018) and MOSE (Ding et al., 2023) datasets. DAVIS defines a training set of 60 videos and a validation set of 30 videos for which all frames are annotated with ground-truth instance segmentation masks. For YouTube-VOS, only the training set is annotated and publicly available, while the validation set is gated behind an evaluation server. To mimic the DAVIS setup, we take a random subset of 2758 annotated videos (80%) as the training set and the remaining 690 videos (20%) as the validation set. In a similar fashion, we split the MOSE dataset into 1206 videos for validation and 301 for testing. For all datasets, we evaluate performance using the standard  $\mathcal{J}$ & $\mathcal{F}$ -mean metric (Perazzi et al., 2016), which combines the region similarity ( $\mathcal{J}$ ) and contour accuracy ( $\mathcal{F}$ ) scores. Only the objects annotated in the first frame are tracked and evaluated, while objects that appear later in the video are ignored, even if their ground-truth masks are annotated.

**Evaluation Protocol** Similar to Rajasegaran et al. (2025), we implement a non-parametric protocol for label propagation based on patch similarity, which is computed as a cosine similarity between features extracted from a frozen DINOv3 backbone. We assume that the first frame of the video is labeled with instance segmentation masks, which we represent as a one-hot vector per patch. For each frame, we compute the cosine similarity between all its patch features, all patches of the first frame, and all patches of a small number of past frames. Focusing on a single patch in the current frame, we consider the  $k$  most similar patches within a spatial neighborhood, and compute a weighted average of their labels to obtain a prediction for the current patch. After processing one frame, we move to the next one, treating the previous predictions as soft instance segmentation labels. When forwarding individual frames through the backbone, we resize the image such that the shortest side matches a certain size, preserving aspect ratio up to the nearest multiple of the patch size.<sup>2</sup> Patch similarity and label propagation are computed at the resolution of the resulting features, then the mask probabilities are bilinearly resized to the native resolution for computing  $\mathcal{J}$ & $\mathcal{F}$ . We consider several hyperparameter combinations, *e.g.* the number of past frames to use as context, the number of neighbors  $k$ , and the size of the spatial neighborhood, as summarized in Tab. 27. We perform hyperparameter selection on the training set of DAVIS, and then apply the best combination to the test splits of all datasets.

## D.6 Video Classification

**Datasets and Metrics** We evaluate DINOv3 on video classification using the UCF101 (Soomro et al., 2012), Something-Something V2 (Goyal et al., 2017), and Kinetics-400 (Kay et al., 2017) datasets. At a high level, we extract a fixed number of frames from each video, encode them with a frozen backbone, collect all patch features into a flat sequence, which we then feed to a shallow transformer-based classifier trained with regular supervision on a set of labeled videos. In previous work, *e.g.* (Assran et al., 2025), this protocol is referred to as an *attentive probe*, a hint to the *linear probes* used for image classification. In the following paragraphs, we describe our implementation of the protocol.

**Training** At training time, we select 16 frames at random temporal locations from each video, keeping track of the corresponding timestamps. We also sample the parameters of a spatial crop that covers between 40% and 100% of the area—these parameters will be shared across all frames of the video to avoid jittering. We then process each frame with DINOv3 as an independent  $256 \times 256$  image, extracting  $16 \times 16$  patch features and discarding the CLS token. For each patch, we keep track of its spatial coordinates defined on a  $[0, 1]^2$  box. The patch features from all frames are linearly projected to 1024 dimensions, concatenated into a flat sequence of length  $16 \times 16 \times 16 = 4096$ , and then fed to four self-attention blocks that model the spatial and temporal relationships between the patches. To ensure the model has access to positional information, we inject the timestamp and spatial coordinates of each patch both as an additive sin-cos embedding before the blocks (Vaswani et al., 2017), and as a 3D factorized RoPE with random spatial rotations in each attention

<sup>2</sup>For example, DAVIS videos are natively  $480 \times 854$  and we want to process them at resolution 960. For a model with patch size 16, we resize the frames to  $960 \times 1712$  with a slight horizontal stretch, resulting in a  $60 \times 107$  feature map. Instead, for a model with patch size 14, we resize the frames to  $966 \times 1708$  with a slight vertical stretch, resulting in a  $69 \times 122$  feature map.

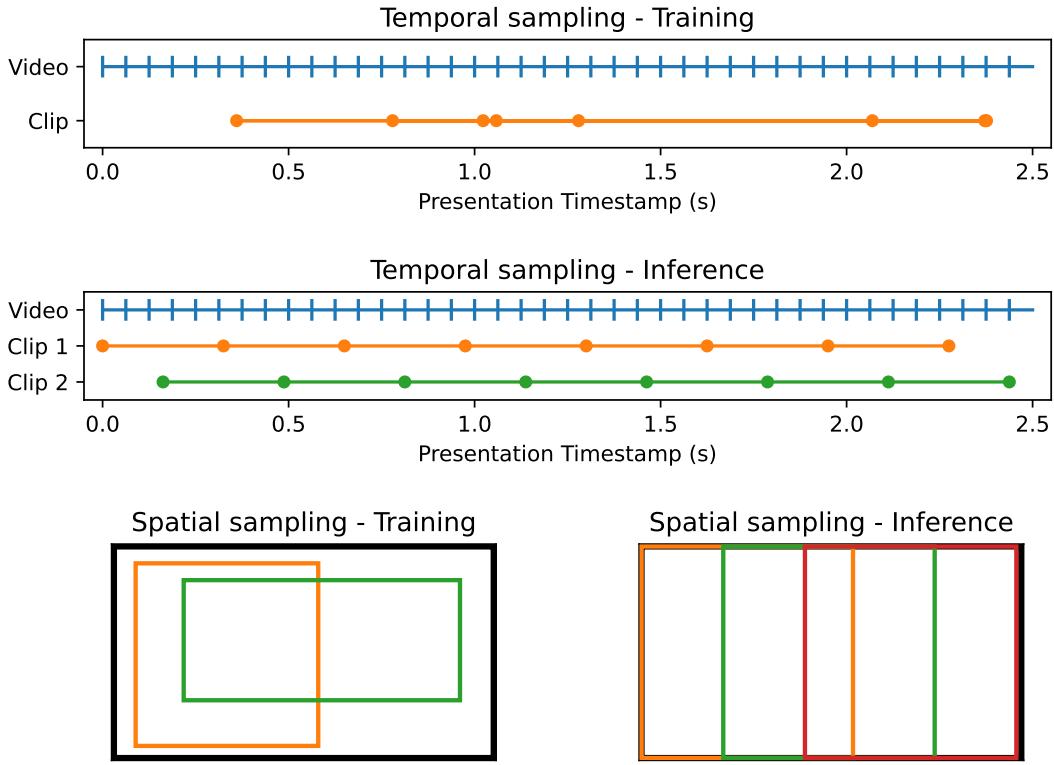
**Table 27:** List of hyperparameters evaluated for video segmentation tracking on the training split of DAVIS 2017 (Pont-Tuset et al., 2017). The best hyperparameters, highlighted, are applied to all datasets.

Max context length	Neighborhood mask size	Neighborhood mask shape	Top-K	Temperature
7	12	Square	5	0.2
7	12	Circle	5	0.2
7	5	Square	5	0.2
7	24	Square	5	0.2
7	$\infty$	—	5	0.2
7	12	Square	5	0.01
7	12	Square	5	0.1
7	12	Square	5	0.7
4	12	Square	5	0.2
10	12	Square	5	0.2
15	12	Square	5	0.2
7	12	Square	3	0.2
7	12	Square	10	0.2
7	12	Square	15	0.2
15	12	Circle	10	0.1
15	24	Circle	10	0.1
15	36	Circle	10	0.1
15	$\infty$	—	10	0.1

head (Heo et al., 2024). After the four blocks, we apply a cross-attention block with a single position-less learnable query to aggregate the information from all patches into a single vector, which is then linearly projected to obtain the final classification logits. The stack of self-attention blocks, the cross-attention block, the positional embeddings, and the final projection constitute the video classifier, which we train for 20 epochs with batch size 64 with a standard cross-entropy loss. In practice, we train a set of classifiers in parallel, one for each combination of learning rate  $\{1 \cdot 10^{-4}, 2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 1 \cdot 10^{-3}, 2 \cdot 10^{-3}, 5 \cdot 10^{-3}\}$  and weight decay  $\{10^{-3}, 10^{-2}, 10^{-1}\}$ . For each dataset, we use 90% of the training set to update the model parameters, 10% of the training set to choose the best combination of learning rate and weight decay, and finally report performance of the chosen model on the validation split.

**Inference** At inference time, we follow a deterministic strategy to sample a single clip per video: we take the first frame, the last frame, and uniformly-spaced frames in between, for a total of 16. From each frame, we crop the largest center square and resize it to  $256 \times 256$  pixels, possibly losing information from the sides of rectangular videos. We then feed these frames to DINOv3 and to the classifier to obtain a prediction for the video. Alternatively, we follow Assran et al. (2025) and perform test-time augmentation (TTA) by selecting multiple frame sequences and multiple spatial crops, processing them independently and then averaging the class probabilities to obtain the final prediction. Clip sampling is exemplified in Fig. 22.

**Baselines** For the chosen baseline models, we use the same evaluation protocol, *i.e.* feature extraction, classifier architecture, training procedure and inference protocol, with a few differences. The input resolution is  $256 \times 256$  pixels for models that use patch size 16, and  $224 \times 224$  pixels for patch size 14. This way, all backbones yield an identical number of tokens, and therefore afford the same amount of computation in the classifier. All models process videos frame by frame independently, since they are trained on images. The only exception is V-JEPA 2, to which we feed whole clips to extract time-aware features. Since V-JEPA 2 reduces the temporal axis by a factor of two, *e.g.* yielding 8 time steps given 16 frames as input, we duplicate each patch token to match the sequence length of other models.



**Figure 22:** Sampling clips for video classification. Choosing a clip for training or inference means determining the coordinates of a spatial crop and which frames/timestamps to sample. At training time, we sample clips at random by choosing random frames from the whole video and by applying a spatial crop that covers  $\geq 40\%$  of the area. At inference time, we select clips in a deterministic way. Spatially, we take the three largest square crops aligned to the left, middle and right. Temporally, we take two overlapping sets of frames such that they cover as much of the video as possible and their timestamps interleave.

## D.7 Image Classification with Linear Probing

**Datasets and Metrics** We evaluate the global quality of the DINOv3 model using the widely adopted linear probing evaluation. We train a linear transform on the training set of ImageNet-1k (Deng et al., 2009) and evaluate results on the val set. We assess the generalization quality of the model by evaluating the transfer to classification test sets: ImageNet-**V2** (Recht et al., 2019) and **Real** (Beyer et al., 2020), which provide alternative sets of images and labels for ImageNet, designed to test for overfitting on the original ImageNet validation set. Additionally, we consider the **Rendition** (Hendrycks et al., 2021a) and **Sketch** (Wang et al., 2019) datasets, which present stylized and artificial versions of ImageNet classes; the **Adversarial** (Hendrycks et al., 2021b) and **ObjectNet** (Barbu et al., 2019) datasets, which contain deliberately challenging examples; and the **Corruptions** (Hendrycks and Dietterich, 2019) dataset, which measures robustness to common image corruptions. We report top-1 classification accuracy as the evaluation metric for all datasets but ImageNet-C, for which we report the mean corruption error (mCE, see (Hendrycks and Dietterich, 2019)).

For fine-grained datasets, we consider the same collection of 12 datasets from Oquab et al. (2024), which we call Fine-S here: Food-101 (Bossard et al., 2014), CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), SUN397 (Xiao et al., 2010), StanfordCars (Krause et al., 2013), FGVC-

---

Aircraft (Maji et al., 2013), VOC 2007 (Everingham et al., 2007), DTD (Cimpoi et al., 2014), Oxford Pets (Parkhi et al., 2012), Caltech101 (Fei-Fei et al., 2004), Flowers (Nilsback and Zisserman, 2008), and CUB200 (Welinder et al., 2010), as well as the larger datasets Places205 (Zhou et al., 2014), iNaturalist 2018 (Van Horn et al., 2018), and iNaturalist 2021 (Van Horn et al., 2021).

**Evaluation Protocol** For the larger datasets ImageNet, Places205, iNaturalist 2018 and iNaturalist 2021, we use the following procedure. For each baseline, we train a linear layer on the final features of the CLS token (after the layer norm) using the ImageNet-1k training set (Deng et al., 2009). Specifically, we use SGD with a momentum of 0.9, and train for 10 epochs with a batch size of 1024. We sweep the learning rates  $\{1 \times 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}, 2 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}, 2 \times 10^{-1}, 5 \times 10^{-1}, 1 \times 10^0, 2 \times 10^0, 5 \times 10^0\}$  and weight decay values  $\{0, 1e-5\}$  and use the validation set of ImageNet-1k to select the best combination. During training, we use random resize crop augmentation with standard Inception-crop parameters. For the datasets in Fine-S, following Oquab et al. (2024), we use a lighter weight evaluation using scikit-learn’s LogisticRegression implementation with the L-BFGS solver.

In both cases, we evaluate models at resolutions resulting in 1024 patch tokens, that is,  $448 \times 448$  for patch size 14, and  $512 \times 512$  for patch size 16. The images are resized such that the shorter side matches the chosen side length, then take the central square crop.

## D.8 Instance Recognition

**Datasets and Metrics** We use the Oxford and Paris datasets for landmark recognition (Radenović et al., 2018), the Met dataset featuring artworks from the Metropolitan Museum (Ypsilantis et al., 2021), and AmsterTime, which consists of modern street view images matched to historical archival images of Amsterdam (Yıldız et al., 2022). In Tab. 9, we report mean average precision (mAP) for Oxford-Hard, Paris-Hard, and AmsterTime, and global average precision (GAP) for Met. In Tab. 23, we additional give mAP for Oxford-Medium and Paris-Medium, as well as the additional metrics GAP- and accuracy (see (Ypsilantis et al., 2021)). For Oxford and Paris, we resize all images such that the larger side is 224 pixels long, keeping the aspect ratio, then take a full center crop, yielding an image of resolution of  $224 \times 224$ . For AmsterTime, we resize all images such that the shorter side is 256 pixels long, keeping the aspect ratio, then take a center crop of size  $224 \times 224$ . For Met, we evaluate all images close to their original resolution, resizing both to the nearest multiple of patch size (resulting in a long side 508/512 for patch size 14/16).

**Evaluation Protocol** The image similarity is computed using cosine distance between the CLS tokens computed for query and target images. We follow the evaluation protocols of (Radenović et al., 2018) for Oxford and Paris, of (Yıldız et al., 2022) for AmsterTime, and of Ypsilantis et al. (2021) for Met. For Met, this includes tuning the hyperparameters  $k$  and  $\tau$  with a grid search, optimizing GAP on the validation set of Met, and whitening the features using a PCA estimated on the training set of Met.

## D.9 Object Detection

**Datasets and Metrics** We evaluate DINOv3 on object detection on the COCO (Lin et al., 2014) and COCO-O (Mao et al., 2023) datasets. COCO is a standard benchmark for object detection, covering 80 object categories, and containing 118k training images and 5k validation images. COCO-O is an evaluation-only dataset with the same categories as COCO, but in more challenging visual conditions, such as scenes with significant occlusions, cluttered backgrounds, and varying lighting conditions. For training the object detection model, we also leverage the Objects365 (Shao et al., 2019) dataset, which contains 2.5M images and covers 365 object categories, a subset of which maps directly to COCO classes. For both COCO and COCO-O, we report mean Average Precision (mAP) computed at IoU thresholds of  $[0.5 : 0.05 : 0.95]$ .

**Architecture** Our approach builds upon the Plain-DETR (Lin et al., 2023b) implementation, with several modifications. We do not fuse the transformer encoder into the *backbone*, but keep it as a separate module, similar to the original DETR (Carion et al., 2020). This allows us to keep the DINOv3 backbone completely frozen during training and inference, making it the first competitive detection model to do so. From a DINOv3 ViT-7B/16 backbone, we extract features from four intermediate layers, namely  $[10, 20, 30, 40]$ . For

---

each patch, we concatenate intermediate features channel-wise, giving a feature dimension of  $4 \cdot 4096 = 16384$ , which is further increased by the windowing strategy described below. The backbone features feed into the *encoder*, which is a stack of 6 self attention blocks with embedding dimension of 768. The *decoder* is a stack of 6 cross attention blocks with the same embedding dimension, where 1500 “one-to-one” queries and 1500 “one-to-many” queries attend to the patch tokens of the encoders to predict bounding boxes and class labels.

**Image Pre-Processing** Training is performed in three stages as described below, one with a base image resolution of 1536 pixels and two with a base resolution of 2048. Following DETR, we apply random horizontal flipping ( $p = 0.5$ ), followed by either (i) random resizing, where the shortest side is uniformly sampled between 920 pixels (resp. 1228) and the base resolution of the stage (1536 or 2048), or (ii) a random crop retaining 60–100% of the original image area, followed by resizing as in (i). At evaluation time, images are resized so that the shortest side is 2048 without additional augmentation, and both sides are rounded up to the nearest multiple of the patch size.

**Windowing strategy** We then apply a windowing strategy that combines a global view of the image with smaller views, to allow the backbone to process objects at all scales. The number of windows is fixed to  $3 \times 3$ , and their sizes vary according to the input resolution. As an example, for an image of size  $1536 \times 2304$ :

1. The image is divided into  $3 \times 3$  non-overlapping windows of size  $512 \times 768$ . Each window is forwarded through the backbone, resulting in  $32 \times 48$  patch tokens of dimension 16384. The features of all windows are spatially reassembled into a  $(3 \cdot 32) \times (3 \cdot 48)$  feature map.
2. The whole image is resized to  $512 \times 768$  and forwarded through the backbone, resulting in a feature map of  $32 \times 48$  patch tokens of dimension 16384. These features are then bilinearly upsampled to  $96 \times 144$ , matching the size of the windowed feature map.
3. Finally, the features maps from steps 1 and 2 are concatenated channel-wise, resulting in a  $96 \times 144$  feature map of dimension  $2 \cdot 16384 = 32768$ . This feature map is then flattened as a sequence of  $96 \cdot 144$  tokens and fed to the encoder.

**Training** We follow a training curriculum in three stages, using the Objects365 dataset (Shao et al., 2019) and the COCO dataset (Lin et al., 2014) at increasing resolutions. Throughout training, we use the AdamW optimizer (Loshchilov and Hutter, 2017) with a weight decay of 0.05. Following DETR, we use the Focal Loss (Lin et al., 2018) as classification loss, with a weight of 2, L1 loss as bounding box loss with a weight of 1, complemented by the GIoU (Rezatofighi et al., 2019) loss with a weight of 2. The stages are as follows:

1. We begin training on Objects365 at base resolution 1536 pixels. We train for 22 epochs with global batch size of 32, which we distribute over 32 GPUs. After an initial warmup of 1000 steps, the learning rate is set to  $5 \cdot 10^{-5}$  and is divided by 10 after the 20th epoch.
2. We then continue training on Objects365 at base resolution 2048 pixels. We train for 4 epochs with learning rate  $2.5 \cdot 10^{-5}$ .
3. We finish training by doing 12 epochs on COCO at base resolution 2048. After a linear warmup of 2000 iterations, the learning rate follows a cosine decay schedule, starting at  $2.5 \cdot 10^{-5}$  and reaching  $2.5 \cdot 10^{-6}$  at the 8th epoch. In this part we use the IA-BCE classification loss (Cai et al., 2024) instead of the simple Focal Loss from DETR. We observed this loss to increase the model performance at transfer time, but not at pretraining time. As this loss mixes class and box information, it brings its full potential if the box predictions are already well initialized. The GIoU loss weight is set to 4 in this part to encourage better box alignment.

**Test-Time Augmentation** At test time, we follow the inference procedure described above, resizing images such that the short side is 1536 or 2048. At those resolutions, the COCO mAP is 65.4 and 65.6, respectively. Alternatively, we can apply the test-time augmentation (TTA) strategy from Bolya et al. (2025), which consists in flipping and resizing the image to multiple resolutions, and merging the predictions with SoftNMS (Bodla et al., 2017). Specifically, each image is processed at resolutions of [1536, 1728, 1920, 2112, 2304, 2496, 2688, 2880], yielding an mAP of 66.1.

---

## D.10 Semantic Segmentation

**Datasets and Metrics** We evaluate DINOv3 on semantic segmentation on the ADE20k (Zhou et al., 2017), Cityscapes (Cordts et al., 2016), COCO-Stuff (Caesar et al., 2018), and VOC 2012 (Everingham et al., 2012) datasets. ADE20k is a widely used benchmark for semantic segmentation with 150 semantic categories, varying from outdoor scenery to images of people and objects inside a house. In addition, COCO-Stuff and Hypersim (Roberts et al., 2021) datasets are used for pre-training the model. COCO-Stuff is a larger dataset (118k training images) than ADE20k containing 80 thing classes and 91 stuff classes, while Hypersim is a photorealistic synthetic dataset presenting indoor scenes with 40 semantic categories, with sharper and more accurate annotations. More than half of the Hypersim images contain 21 or more objects, making it a good candidate for helping the model learn rich information of the scenes. The evaluation metric reported is mIoU for all datasets.

**Architecture** We adapt the ViT-Adapter and Mask2former configurations that other baselines use (Wang et al., 2023a), with several differences. First, to ensure that our backbone remains frozen and its activations are not altered, we remove the injector component of the ViT-Adapter. This makes our backbone output features to be directly used in the extractor module. Second, the embedding dimensions in the Mask2former decoder are scaled to 2048 instead of the default 1024 to adapt to our backbone output dimension of 4096, while other baselines’ backbones usually present an output dimension of 1024 or 1536. As inputs to the decoder, we extract features from four intermediate layers of the DINOv3 7B/16 backbone, namely layers [10, 20, 30, 40]. We apply the final layer norm to the features of all layers and add a learned batch normalization.

**Training Protocol** For generating results on COCO-Stuff, we train the model using a cosine scheduler, with a 6k linear warmup and a maximum learning rate of 1.5e-5. The model is trained for 80k iterations, at resolution 1280 pixels. As for training on the other datasets—ADE20k, Cityscapes and VOC 2012—we first pre-train the decoder on COCO-Stuff for 80k iterations, with a 6k linear warmup and a learning rate of 1.5e-5, following a cosine scheduler. This helps the model learn diverse semantic categories (171 categories) on a larger dataset than ADE20k. The model is then trained on Hypersim for 10k iterations at a learning rate of 2.5e-5 following a cosine scheduler with a 1.5k linear warmup. Corresponding to roughly 2 epochs, this step helps our model learn high-quality image-to-mask correspondence due to their photorealistic synthetic nature. Finally, our model is trained on ADE20k for 20k iterations with a learning rate of 3e-5, again with a 1.5k linear warmup and a cosine schedule. We report our final result on the validation set. For Cityscapes and VOC 2012, learning rates of 1.5e-5 and 1e-5 are used respectively. For all training, a batch size of 16 and the AdamW optimizer is used.

**Inference** For single-scale evaluation, sliding inference is used for evaluating the models—the image is first resized at the training resolution (*e.g.* a  $400 \times 500$  image will be resized to  $896 \times 1120$  pixels for ADE20k, since the model was trained at resolution 896). Then, a sliding window method is used with a stride (*e.g.* stride of 596 pixels for ADE20k) on square crops (*e.g.*  $896 \times 896$  pixels) to generate a prediction for each crop, sliding through the image. These results are then aggregated and rescaled to the original image size to generate a final prediction. For test-time augmentation, both ADE20k and VOC 2012 images were rescaled to ratios of [0.9, 0.95, 1.0, 1.05, 1.1] of the evaluation resolution, and each image was also flipped horizontally to generate a total of 10 predictions per sample. After sliding inference on each image, they were rescaled to the original image shape and averaged. COCO-Stuff 164K’s TTA mIoU was obtained by simply using an additional horizontally flipped image per sample, and for Cityscapes, ratios of [1.0, 1.5, 2.0] of the evaluation resolution were applied.

## D.11 Monocular Depth Estimation

**Implementation Details** Our approach differs from Depth Anything v2 (DAv2) (Yang et al., 2024b) primarily in the configuration of image resolution, which is set to  $768 \times 1024$  pixels, and the network architecture. The backbone is kept frozen throughout training, while a dropout rate of 0.05 is applied at the end of the DPT head (Ranftl et al., 2021). As input to the decoder, we extract features from four intermediate

---

layers of the DINOv3 7B/16 backbone, namely layers [10, 20, 30, 40]. We apply the final layer norm to the features of all layers and add a learned batch normalization. The depth estimation output is discretized into 256 uniformly distributed bins covering the range from 0.001m to 100m. Training employs a base learning rate of 1e-3, scheduled using PolyLR with a power of 3.5 and an initial linear warmup phase lasting 12k iterations. To enhance robustness and generalization, we apply a suite of augmentations: Gaussian blur, Gaussian noise, AutoContrast, AutoEqualise, ColorJitter, rotation, and left-right flip.

**Datasets and Metrics** We train the model on the dataset of DAv2, which consists of synthetically generated images from the IRS, TartanAir, BlendedMVS, Hypersim, and VKITTI2 datasets. We evaluate on five datasets: NYUv2 (Silberman et al., 2012), KITTI (Geiger et al., 2013), ETH3D (Schöps et al., 2017), ScanNet from (Ke et al., 2025), and DIODE (Vasiljevic et al., 2019). We adopt the zero-shot scale-invariant depth setup, and report the standard metrics absolute relative error and  $\delta_1$  (see (Yang et al., 2024a)).

## D.12 Visual Geometry Grounded Transformer with DINOv3

**Implementation Details** Compared to the original VGGT (Wang et al., 2025), we adopt the following changes: (1) we use an image size of 592 instead of 518; this is to match the number of patch tokens that DINOv2 produces, (2) adopting a smaller learning rate, specifically from 0.0002 to 0.0001, and (3) using a concatenation of the four intermediate layers of DINOv3 ViT-L rather than just the last layer as input to the downstream modules. Interestingly, we found that using four intermediate layers brings a benefit for DINOv3, whereas doing the same for DINOv2 brings no additional performance gains. We also experimented with a version closer to the original VGGT setup (image size 512, same learning rate, final layer), and already found this untuned version to improve over the original VGGT work across all tested benchmarks.

## D.13 Geospatial

**Evaluation details** In all of the evaluations, we keep the backbone frozen and only train lightweight classifiers or decoders that are specialized for the tasks. For GEO-Bench classification, we train a linear classifier for 2400 iterations with a batch size of 32. We use SGD optimizer, cosine learning rate annealing, and select the best learning rate between 1e-5 and 1. Unless otherwise specified, segmentation evaluations use a DPT decoder (Ranftl et al., 2021), with a learning rate selected based on performance on the validation set with a grid search of four values in [3e-5, 1e-4, 3e-4, 1e-3].

On LoveDA and iSAID datasets, we train an UPerNet decoder (Xiao et al., 2018) for 80k iterations, with a batch size of 8, and a linear warm-up of 1500 iterations in line with (Wang et al., 2024a). All other hyper-parameters such as crop size and weight decay are the same as in (Wang et al., 2024a). Following previous work (Tolan et al., 2024; Wang et al., 2022a), we use a DPT head for canopy height prediction evaluations and train a Faster RCNN (Ren et al., 2015) detector for 12 epochs for object detection tasks.

**Satlidar dataset** The Satlidar dataset consists in one Million of  $512 \times 512$  Maxar images and corresponding dense lidar measurements collected from different locations as described in Table Tab. 28. The images were extracted from larger tiles, the numbers of tiles for each sub-dataset are specified in the table.

**Table 28:** Description of the Satlidar dataset.

Subdataset	Path	Amount of tiles	Purpose
Kalimantan	<a href="https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1540">https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1540</a>	86	train/val/test
OpenDC	<a href="https://opendata.dc.gov/datasets/2020-lidar-classified-las/about">https://opendata.dc.gov/datasets/2020-lidar-classified-las/about</a>	68	train/val/test
Brazil	<a href="https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1644">https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1644</a>	37	train/val/test
Mozambique	<a href="https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1521">https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1521</a>	144	train/val/test
Neon	<a href="https://data.neonscience.org/data-products/DP3.30015.001">https://data.neonscience.org/data-products/DP3.30015.001</a>	5366	train/val/test
CA20Graup	<a href="https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.092021.6339.1">https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.092021.6339.1</a>	99	train/val/test
CA17Duvall	<a href="https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.042020.6339.2">https://portal.opentopography.org/datasetMetadata?otCollectionID=OT.042020.6339.2</a>	56	train/val/test
Netherlands	<a href="https://geotiles.citg.tudelft.nl/">https://geotiles.citg.tudelft.nl/</a>	13	train/val/test
Sao Paulo	<a href="https://daac.ornl.gov/CMS/guides/LiDAR_Forest_Inventory_Brazil.html">https://daac.ornl.gov/CMS/guides/LiDAR_Forest_Inventory_Brazil.html</a>	4	test
CA brande	<a href="https://doi.org/10.5069/G9C53J18">https://doi.org/10.5069/G9C53J18</a>	1	test