MASKED CROSS-ATTENTION ADAPTERS ENABLE THE CHARACTERIZATION OF DENSE FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning meaningful representations is a core topic of deep learning. Throughout the last decade, many strategies for learning image representations have been proposed involving supervision and self-supervision and various data sources. In most current work, evaluation is focused on classification tasks while neglecting dense prediction tasks, possibly because linear probing is more challenging in the latter case. Furthermore, dense prediction heads are often large and come with specific inductive biases that distort performance measurement further. In this work we propose masked cross-attention adapters (MAXA), a minimal adapter method that is capable of dense prediction independent of the size and resolution of the encoder output. This allows us to make dense predictions using a small number of additional parameters ($< 0.3\%$) while allowing for fast training using frozen backbones. Using this adapter, we run a comprehensive evaluation assessing instance awareness, local semantics and spatial representation of a diverse set of backbones. We find that DINOv2 outperforms all other backbones tested – including those supervised with masks and language – across all three task categories.
Code is available at https://to.be.released.

1 INTRODUCTION

Computer vision builds to a large extent on a transfer learning-based paradigm. Instead of training specific tasks from scratch, large-scale models (foundation models) are pre-trained on a large dataset once. These models are often called backbone or feature extractor as they are used to address multiple specific tasks, for example through in-context learning, adapters or (often costly) fine-tuning. For pre-training the backbone, different variants exist, including classic supervised, self-supervised, and vision-language training. The latter two variants tend to scale better as they are not constrained by availability of labels and can use the internet as a data source. Many factors influence the quality of the resulting backbone: the pre-training paradigm, model architecture, the training data. For both, computer vision scientists and practitioners, it is crucial to work out strengths and weaknesses of individual backbones through systematic benchmarks. Established approaches are linear and attentive probing. For (whole-image) classification performance such benchmarks are readily available (ImageNet (Russakovsky et al., 2014), VTAB (Zhai et al., 2019)). However, for dense tasks, where the model output has spatial dimensions (for example semantic segmentation or monocular depth estimation) such an evaluation is more challenging: Linear and attentive probing can be applied but the prediction has the same resolution as the feature volume which is usually low and varies across backbones. On the other hand, using standard task heads for dense prediction adds a large number of parameters and introduces its own inductive biases. The measure performance would depend to a large extent on the chosen task head and not on the underlying backbone.

Here we address the problem of measuring performance of feature backbones as directly as possible by introducing a dense equivalent to attentive probing that requires only a small number of parameters. Our goal is to obtain a holistic characterization of strengths and weaknesses of common feature extractors. To this end, we propose a novel method that uses masked cross-attention (Fig. 1) to extract relevant features from the backbone activations. By using cross-attention, we decouple the size and resolution of the input image and encoder output from that of the dense output, i.e. generate outputs at any resolution. We introduce a learnable masking radius in the cross-attention

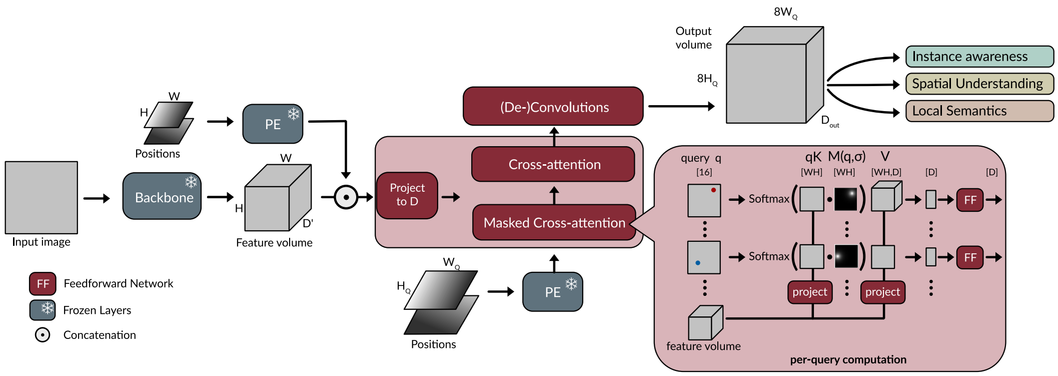

Figure 1: Masked cross-attention adapter (MAXA) design: Queries that consist only of positions (bottom) attend to features extracted from an arbitrary backbone. The transformed queries are processed by a small CNN to yield a task-specific output. Our adapter decouples the output size from the feature volume.

layer, which allows the readout adapter to adapt to varying feature locality. Intuitively, our adapter can be viewed as analogous to linear probing for dense prediction tasks.

We use the new adapter to characterize features along these three dimensions: (1) instance disentanglement, i.e. how well are individual instances recognizable from the features; (2) local semantics, evaluating how meaningful the features are for a local classification; (3) spatial understanding, how well is the 3d structure of the scene captured. Our main contributions are:

- MAXA, a lightweight adapter based on masked cross-attention, designed to operate on frozen features. As features are frozen, training is fast and activations can be cached for interactive use-cases. The adapter decouples feature resolution from output resolution.
- Characterization of several state-of-the-art feature extractors on dense prediction tasks. It allows gaining insights into what is learned by different learning paradigms and datasets, which can be used as a guide for practitioners and an informative signal for researchers.

2 Related Work

Representation learning Self-supervised representation learning was a popular research topic with multiple approaches that can roughly be categorized into joint-embedding (Chen et al., 2020; 2021; Caron et al., 2021) and reconstruction-based (He et al., 2022). DINOv2 is based on the iBot (Zhou et al., 2021) method which uses a joint-embedding architecture in combination with self-distillation and reconstruction. VicRegL (Bardes et al., 2021) and many other recent methods explicitly addresses local features by modeling losses at the token level. There has been a discussion about which techniques leads to better and more efficient features for perception tasks (Balestriero & LeCun, 2024). While the approaches discussed above mainly address classification, another stream of research focused on representation learning that disentangles objects, called object-centric learning. While early methods only worked on synthetic data (Burgess et al., 2019; Locatello et al., 2020) more recent approaches succeed on natural images (Zadaianchuk et al., 2024; Aydemir et al., 2023). Recently, a new method for evaluating such object-centric representations was proposed: Didolkar et al. (2024). The main difference to object-centric methods is that we assume features encode object instances whereas object-centric methods explicitly represent objects in their architecture (e.g. in slots). The seminal CLIP model (Radford et al., 2021) introduced another stream of research called vision-language models, where the model is trained on aligning text-image pairs. Later, this training paradigm was simplified to use a sigmoid-based loss function (Zhai et al., 2023) instead of a softmax-basde loss, making the method less dependent on the batch size. Recently, the role of data is investigated more closely in the context of vision-language models (Gadre et al., 2024; Xu et al., 2024; Fang et al., 2024).

Feature evaluation Evaluations on features predate the deep learning era in computer vision. Recently, there have been numerous attempts at characterizing and comparing common feature backbones but with different objectives. The works by Bonnen et al. and El Banani et al. focus on 3d shape understanding. Chen et al. design a zero-shot benchmark for image encoders in contrastive vision-language pre-training setting and propose the ViTamin architecture. Goldblum et al. evaluate classification, instance segmentation, object detection and retrieval. Our work differs in focusing on dense prediction tasks without large heads enabling a more direct measurement of the feature quality. Further efforts to characterize vision backbones include the timm leaderboard (Wightman, 2019) for image classification, CLIP benchmark (LAION-AI, 2022) for vision-language models and CV-Bench for multimodal large language models (MLLMs) (Tong et al., 2024).

Adapters and parameter-efficient fine-tuning Adapters are (often small) sub-networks that are trained to take generic features and use them to solve a specific task. For computer vision problems, many adapters were proposed that address whole-image classification (Chen et al., 2022; Steitz & Roth, 2024). Also, the attentive probing (or attentional pooling) used in the CoCa (Yu et al., 2022) and V-JEPA evaluation (Bardes et al., 2023) can be considered a minimal adapter for whole image classification. These methods are not straightforward applicable for dense prediction. Based on the upsampling method FeatUp (Fu et al., 2024), linear evaluation can be applied in higher resolutions. To our knowledge this has not been done before but we compare to a baseline that uses this approach. The methods by Bhattacharjee et al. and Yang et al. adapt to dense images but address multiple tasks at once. In both methods, the backbones are not entirely frozen.

In another research stream, learnable parameters are added inside the frozen backbone network, for instance in Adapter (Houlsby et al., 2019), low-rank adaptation (Hu et al., 2021) and scaling-and-shifting (Lian et al., 2022). ViT-Adapter applies this paradigm for dense prediction tasks but builds on established task heads for segmentation (UperNet) and detection (Mask RCNN and HTC++). Furthermore, the number of parameters introduced by the adapter depends on the backbone, ranging from 2.5M to 23.7M parameters. For full review on adapters we refer to the survey of Yu et al..

3 Masked Cross-attention Adapter (MAXA)

In this section, we introduce MAXA. It is designed to be parameter and compute efficient adapter model by using cross-attention to make dense predictions and operating on a frozen backbone. An arbitrary (frozen) image backbone ϕ receives an image $\mathbf{x}$ of size $(Hs, Ws, 3)$ and generates features of size (H, W, D') with s indicating the backbone's stride. These features are concatenated with a fixed positional encoding P. The resulting activations are projected to the internal dimension D and flattened along the spatial dimensions (both by ψ), yielding $\mathbf{F}(\mathbf{x})$ of size (H, W, D):

$$\mathbf{F}(\mathbf{x}) = \psi(P(\phi(\mathbf{x}))) \tag{1}$$

To generate a dense output, we use a cross-attention-based approach: All spatial queries $\mathbf{Q}$ of size $(H_Q W_Q, 16)$, attend to the feature volume $\mathbf{F}$, where each query $\mathbf{Q}_j \in \mathbb{R}^{16}$ is responsible for generating the output of a certain region. The queries $\mathbf{Q}$ are fixed, 16-dimensional positional encodings of the respective output positions and thus have no learnable parameters.

We modify the cross-attention under consideration of spatial proximity by adding $\mathbf{M}(\mathbf{q}, \sigma)$. The computation per head is described by

$$\mathbf{q}' = \text{softmax}\left(\frac{W_q \mathbf{Q} W_k \mathbf{F}(\mathbf{x})}{\sqrt{d_k}} + \mathbf{M}(\mathbf{q}, \sigma)\right) W_v \mathbf{F}(\mathbf{x}) \tag{2}$$

with $W_v, W_k \in \mathbb{R}^{D \times D}$ and $W_q \in \mathbb{R}^{D \times 16}$.

The attention operates over all backbone pixels for each query, hence $\mathbf{M}(\mathbf{q}, \sigma)$ has size $(HW, H_Q W_Q)$. Each element M_{ij} depends on the euclidean distance d_{ij} between pixel i in the feature volume and j in the output (i.e. $\mathbf{q}$) through a Gaussian function

$$M_{ij} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right). \tag{3}$$

Here, σ is a learned parameter per attention head. This means the size of the region around each query position from which features are considered is adaptable for each model. In our model, we use

Supervised

Supervised	ViT-B (86M)	ImageNet
SAM2	Sam B+	SA-1B

Self-supervised

MoCoV3	J	ViT-B (86M)	ImageNet
MAE	R	ViT-B (86M)	ImageNet
Hiera	R	Hiera B+ (69M)	ImageNet
DINO	J	ViT-B (86M)	ImageNet
DINOv2	J & R	ViT-B (86M)	LVD-142M
DINOv2	J & R	ViT-L (304M)	LVD-142M

Vision-language

CLIP	ViT-B (86M)	CLIP
MetaCLIP	ViT-B (86M)	CC-400M
SigLIP	ViT-B (86M)	Webli
SigLIP (SO)	ViT (414M)	Webli
SigLIP 512	ViT-B (86M)	Webli
Aim2 (300M)		many
ViTamin-L2 (333M)		DataComp-1
ConvNeXt-L		LAION-2B
Phi3.5	ViT-L (304M)	many

Table 1: Models, their backbone architectures (with parameters) and their training datasets. J and R denote joint-embedding and reconstruction self-supervised learning methods.

two cross-attention layers, with only the first one using adaptable spatial regions while the second layer can attend to any position.

For the final adapter, we use two cross-attention layers with masking activated only in the first one. Instead of using a separate query for every output pixel, regions of size 8×8 are processed jointly for efficiency reasons, i.e. each query $\mathbf{q}$ generates 64 pixels of the output. This is realized through a small CNN operating on the output of all queries using transposed convolutions to increase spatial size. The number of channels in this CNN is given by $D_{\text{CNN}} = \max(D/4, D_{\text{out}})$, with the number of output channels D_{out} being task-dependent.

The queries only receive a position as input and can attend to all features, thus the architecture resembles implicit networks (Mescheder et al., 2019; Park et al., 2019), especially PiFU (Saito et al., 2019) and PixelNerf (Yu et al., 2021) which involve feature extraction. The modification of the attention through a bias term is similar to GraphDINO (Weis et al., 2021).

4 Experiments

Choice of models and training datasets We select a broad range of feature backbones that encompass different training paradigms and were trained on different datasets (Tab. 1). This enables us to conduct controlled comparisons along several axes, for instance, datasets and pre-training task. In general, we differentiate between three broad classes of methods: supervised (Dosovitskiy et al., 2021; Ravi et al., 2024), self-supervised (Caron et al., 2021; He et al., 2022; Ryali et al., 2023), and vision-language (Radford et al., 2021; Xu et al., 2024; Fini et al., 2024; Chen et al., 2024; Liu et al., 2022). The latter involves training on image-caption pairs often obtained from the Internet, while self-supervised training operates on images only. Most models are trained on the ImageNet dataset (Russakovsky et al., 2014), but there are several exceptions: All SigLIP models are trained on the Webli dataset, a Google-internal dataset of 10 billion images with 12 billion multi-lingual text-image pairs. MetaCLIP uses a selection of the open LAION dataset (Schuhmann et al., 2021), CLIP is trained on the unpublished CLIP dataset by OpenAI. DINOv2 (Oquab et al., 2023) is trained on the LVD-142M, a Meta-internal dataset of 142M images which were deduplicated and curated to be similar to ImageNet-22k images. The data mix of Aim2 Fini et al. (2024) contains DFN-2B, COYO, the proprietary HQITP dataset and synthetic data.

We decide to mainly focus on vision transformers as many approaches share the same architecture and checkpoints are available for a large number of training paradigms. To ensure comparability with other work and control for model architecture, we primarily use ViT-B/16 and similarly sized models in our experiments. We also include larger models in some cases to obtain an estimate of how much performance can be improved simply by scaling-up model size. Pretrained-weights are obtained from timm (Wightman, 2019).

Experiment design We provide images in the native resolution of the respective backbones to prevent out-of-distribution input. For generating predictions, we make use of the capability of our model to decouple input and output resolution (see Sec. 3): The output size is fixed to 224×224 for all models, ensuring a fair comparison across models and limiting the advantage of large input sizes for the backbone. In the masked cross attention we use a dimension $D = 32$ and eight attention heads (i.e. four dimensions per head). Although the adapter could attend to backbone tokens at different levels we opt for using only the last layer, motivation by the simple pyramid from Li et al..

We pursue a straightforward approach to comparison: We freeze the backbone features, train the small readout adapter in a supervised way and evaluate on a hold-out test set. The rationale is that the low expressivity and capacity of the adapter forces the adapter to directly rely on the features volume for making a dense prediction. This is different from conventional task heads (e.g. in detection) which are able to execute more complex computations on the features. For example, Faster R-CNN with a ResNet50 backbone add around 18 million parameters to the backbone [1]

4.1 Evaluation tasks

To characterize a broad spectrum of traits of the features we implement the following tasks:

Instance awareness In this task we evaluate how well the features are able to disentangle individual instances. In the field of object-centric machine learning (Burgess et al., 2019) models are designed to disentangle instances, here we ask to which degree this is already achieved in different backbones. Individual instances can be encoded in various ways in the features. We consider the following three notions of how instances are encoded (Fig. 2):

- Instance boundaries: The objective is to outline individual objects in the image. We frame this problem as a binary segmentation and consequently use an output dimension $D_{\text{out}} = 1$ as well as the binary cross-entropy loss function on the adapter output.
- Distance Transform: This is similar to boundaries but computes a single dense map ($D_{\text{out}} = 1$) of normalized distances to the instance boundaries. Here we apply the mean squared error as loss and metric.
- Instance discrimination: Another way to encode instances is to generate a latent space where features within instances are the same (or similar) while being different to all other instances. If this works perfectly, clustering the latent vectors of all pixels would yield instances. This task is sometimes also called coloring (Novotny et al., 2018). We train on only 8,000 sample images and treat every instance as an individual class (resulting in a around 60,000 classes). The dense adapter maps the features to a latent space (in our case 32, $D_{\text{out}} = 32$). Then, a linear layer maps each local 32-dimensional feature to a probability over all instances in the dataset. Thus, the problem is essentially framed as semantic segmentation with 60,000 classes. This way, the latent features before the classification head learn to discriminate instances. For testing, we cluster these features obtained from unseen images. For clustering we use k-means and provide the ground-truth number of instances as well as a foreground mask. Then we compare the predicted foreground instances with the ground truth instance segmentation based on the adjusted rand index.

For these experiments, we use the COCO dataset (Lin et al., 2014), with the 5,000 images from the validation set being used for testing. For instance discrimination, we consider only images with at least three large objects (resulting in a subset of 754 images). Note, these tasks do not involve classifying the instances into object categories, unlike typically done in instance segmentation (this is assessed below in "local semantics").

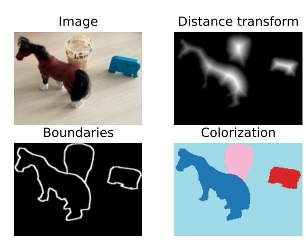

Figure 2: We use three tasks to probe instance awareness. Top-left: input image.

Local semantics A natural choice for evaluating local semantics is a semantic segmentation task. Here we rely on two benchmarks: Pascal VOC 2012 (Everingham et al., 2015) and COCO Stuff (Caesar et al., 2018). The Pascal VOC 2012 encompasses a fairly small set of only 1,464 training images. For COCO Stuff, we train on 100,000 images. We account for the larger number of classes in COCO stuff by setting the internal dimension of the CNN, D_{CNN}, to 64.

Spatial understanding To assess how well the features capture the 3D structure of the scene, we implement the well-known monocular depth estimation task: The adapter needs to infer the depth

[1]These calculations were obtained using the Faster R-CNN implementation in PyTorch vision (Paszke et al., 2019), in more detail FPN: 3.3M, RPN: 0.6M, ROI heads: 14.3M parameters.

					Inst. Disc.			Boundaries		Distance Transform	
0	Backbone	I	F	P	ARI	P_{learn}	CE	IoU	P_{learn}	MSE	P_{learn}
	Random (untrained)	224	14	86.0	20.7	0.221	0.3021	2.6	0.184	0.0708	0.184
	ImageNet	224	14	86.0	35.5	0.221	0.1951	20.7	0.184	0.0173	0.184
	SAM V2 B+	1024	64	80.8	50.4	0.151	0.1644	28.9	0.115	0.0155	0.115
	MoCo V3	224	14	86.0	40.9	0.221	0.1759	24.9	0.184	0.0158	0.184
	Dino	224	28	86.0	44.5	0.225	0.1607	29.1	0.188	0.0145	0.188
	Dino V2	518	37	86.8	54.5	0.230	0.1461	33.3	0.193	0.0109	0.193
	Dino V2 (ViT-L)	518	37	304.6	**55.0**	0.280	**0.1450**	**33.8**	0.243	**0.0108**	0.243
	MAE	224	14	86.0	48.3	0.221	0.1657	28.9	0.184	0.0150	0.184
	Hiera B+	224	7	69.2	46.5	0.244	0.1880	21.1	0.208	0.0159	0.208
	CLIP	224	14	86.0	40.6	0.221	0.1820	23.1	0.184	0.0153	0.184
	CLIP (ViT-L)	336	24	303.8	44.0	0.274	0.1697	27.7	0.237	0.0137	0.237
	MetaCLIP	224	14	86.0	41.0	0.221	0.1816	22.8	0.184	0.0153	0.184
	SigLIP-224	224	14	86.0	39.1	0.221	0.1862	22.0	0.184	0.0153	0.184
	SigLIP-384	384	24	86.3	40.4	0.224	0.1711	26.5	0.187	0.0139	0.187
	SigLIP-512	512	32	86.6	41.2	0.227	0.1645	28.4	0.190	0.0134	0.190
	SigLIP-SO	512	36	413.9	44.5	0.305	0.1639	28.8	0.268	0.0131	0.268
	Aim2	336	24	309.8	42.5	0.274	0.1706	26.9	0.237	0.0134	0.237
	ViTamin	384	24	333.2	40.5	0.274	0.1682	27.2	0.237	0.0134	0.237
	ConvNeXt	320	10	196.6	33.9	0.370	0.1897	20.0	0.333	0.0152	0.333
	ConvNeXt (2 layers)	320	20	196.7	32.6	0.523	0.1723	25.8	0.486	0.0131	0.486
	Phi-3.5V	336	24	303.8	43.8	0.274	0.1630	28.9	0.237	0.0133	0.237

Table 2: Instance awareness results in all three categories. I denotes image size, F denotes feature volume size, P and P_{learn} refer to all and only learnable parameters. Metrics are intersection over union (IoU), cross-entropy (CE), mean-squared error (MSE) and adjusted rand index (ARI).

(i.e. position along the z-axis) for every pixel of the visible scene based on the features provided by the backbone. We frame this as a depth map problem, i. e. $D_{\text{out}} = 1$, relying on the NYUv2 dataset (Nathan Silberman & Fergus, 2012) for training and testing the adapter.

4.2 Comparison on backbones

Results (Tab. 2) indicate that DINOv2 has the best instance awareness. The backbone of SAM2 shows a fairly low performance despite being trained for instance discrimination. This suggest that objects are not disentangled before SAM2's mask decoder. Despite following the same reconstruction-based training, MAE performs better than Hiera which could potentially be due to the more intensive spatial compression in Hiera. Among the vision-language models CLIP, MetaCLIP and SigLIP we did not find meaningful differences. The evaluation on spatial understanding shows mixed results. Larger backbones tend to perform better, with exception of Hiera-B+. Again, DINOv2 performs best, in this case by a large margin. Vison-language tend to show stronger local semantics (Tab. 3). While all VLM 224px models show similar performance, the larger versions of SigLIP (i.e. 384 and SO) perform better but at a higher cost. Also in this evaluation, DINOv2 achieves the best scores. All things considered, possibly the most striking finding is the dominance of DINOv2. While one might argue that this is due to large image sizes and feature volumes, the mediocre performance of SigLIP-512 and Hiera-B+ show that cannot be the only factor.

4.3 Adapter design

We next explore design choices of MAXA by varying relevant hyperparameters of our readout adapter (Tab. 4). The two-layer cross-attention and the introduction of the σ parameter are crucial for good performance. The latter suggests that information is organized locally in the feature volume. Reducing the dimensionality impacts instance discrimination, possibly as space is scarce for embedding 60,000 instances. Thus, an even smaller adapter could be used for the other tasks.

					Pascal VOC2012			COCO Stuff			Depth	
0	Backbone	I	F	P	CE	IoU	P_{learn}	CE	IoU	P_{learn}	MSE	P_{learn}
	Random (untrained)	224	14	86.0	2.2651	2.2	0.200	6.1879	0.2	0.337	416.9	0.184
	ImageNet	224	14	86.0	0.3468	62.7	0.200	1.3855	31.8	0.337	54.4	0.184
	SAM V2 B+	1024	64	80.8	0.6209	35.5	0.130	1.8030	18.0	0.267	52.7	0.115
	MoCo V3	224	14	86.0	0.2938	67.0	0.200	1.3256	32.0	0.337	49.6	0.184
	Dino	224	28	86.0	0.2756	70.2	0.204	1.2134	35.2	0.341	44.1	0.188
	Dino V2	518	37	86.8	0.1263	85.3	0.209	0.9766	45.6	0.346	22.8	0.193
	Dino V2 (ViT-L)	518	37	304.6	0.1152	86.1	0.259	0.9572	47.2	0.396	21.1	0.243
	MAE	224	14	86.0	0.3649	62.2	0.200	1.3326	30.9	0.337	40.9	0.184
	Hiera B+	224	7	69.2	0.3589	63.6	0.223	1.3565	30.6	0.360	33.0	0.208
	CLIP	224	14	86.0	0.2416	71.6	0.200	1.2182	36.5	0.337	45.4	0.184
	CLIP (ViT-L)	336	24	303.8	0.2043	77.1	0.253	1.1294	40.7	0.390	32.9	0.237
	MetaCLIP	224	14	86.0	0.2404	70.8	0.200	1.2215	36.5	0.337	42.2	0.184
	SigLIP-224	224	14	86.0	0.2908	70.4	0.200	1.2149	37.5	0.337	42.2	0.184
	SigLIP-384	384	24	86.3	0.2023	76.4	0.203	1.1498	39.8	0.340	38.6	0.187
	SigLIP-512	512	32	86.6	0.1973	78.2	0.206	1.1381	39.8	0.343	36.8	0.190
	SigLIP-SO	512	36	413.9	0.1764	80.4	0.284	1.0680	43.3	0.421	31.5	0.268
	Aim2	336	24	309.8	0.2029	77.4	0.253	1.0870	41.7	0.390	31.3	0.237
	ViTamin	384	24	333.2	0.1834	79.2	0.253	1.0738	42.7	0.390	30.6	0.237
	ConvNeXt	320	10	196.6	0.3077	68.0	0.349	1.2153	37.3	0.486	35.8	0.333
	ConvNeXt (2 layers)	320	20	196.7	0.2409	74.5	0.502	1.1222	40.8	0.639	29.7	0.486
	Phi-3.5V	336	24	303.8	0.2353	76.1	0.253	1.1475	40.1	0.390	37.1	0.237

Table 3: Local semantics results on Pascal and COCO Stuff as well as spatial understanding on NYUv2 (right). I denotes image size, F denotes feature volume size, P and P_{learn} refer to all and only learnable parameters.

	Inst. Disc.			Semseg			Depth ↓		
	Dino V2	SigLIP-384	MAE	Dino V2	SigLIP-384	MAE	Dino V2	SigLIP-384	MAE
base (our)	**0.5287**	**0.4180**	**0.4832**	**85.5**	**77.1**	60.9	0.2330	**0.3850**	**0.4051**
$D = 16$	0.4777	0.3650	0.3799	83.7	74.5	60.1	**0.2284**	0.3887	0.4156
no σ	0.3303	0.2833	0.3744	64.3	46.5	53.5	0.3039	0.4391	0.4409
no indiv. σ	0.4887	0.3699	0.4360	85.0	75.3	**62.6**	0.2488	0.4091	0.4175
single MCA layer	0.4581	0.3755	0.4301	85.0	75.5	61.4	0.2296	0.4025	0.4077

Table 4: Ablation. The base model is the variant we use in all other experiments in this work.

Oquab et al. report scores for different readout methods in the depth prediction task. As this is a dense prediction tasks, we can directly compare against their scores. They compare against a 1-layer, a 4-layer readout as well as the DPT method (Ranftl et al., 2021). Our scores are consistently better, even when comparing ViT-H with ViT-B features of MAE.

A natural approach for dense prediction is to employ a convolutional neural network with transposed convolutional layers on top of the feature volume. We implement such a baseline that first applies self-attention on the feature volume to enable context integration and then uses a convolutional neural network to generate the output. Furthermore, we employ FeatUp (Fu et al., 2024) for image-aware feature volume upsampling and then apply linear probing. The results (Fig. 3) show that MAXA is more parameter efficient and achieves better scores than this baseline in all three tasks. An additional advantage of our method over CNNs is decoupling input and output resolution, in a CNN, a higher feature volume size would cause a larger output. FeatUp is highly parameter efficient has high memory demands and requires long computation times (factor 4 compared to MAXA).

Variable output size To ensure a fair comparison, the readout size is fixed in the previous experiments. However, it is possible to generate outputs at an arbitrary resolution, because the adapter

	lin. 1	lin. 4	DPT	ours
MAE ViT-H	0.52	0.48	0.42	-
MAE ViT-B	-	-	-	0.41
DINO	0.56	0.54	0.49	0.44
DINOv2	0.40	0.36	0.32	0.23

Table 5: MSE comparison against readouts on depth prediction. Scores for other models are taken from from Oquab et al. (2023).

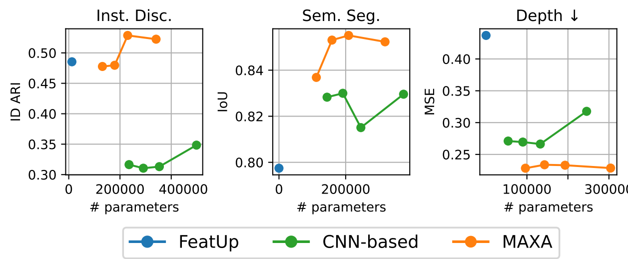

Figure 3: Comparison between a CNN-based readout and our adapter (instances: ↑ = better; depth ↓, semantic seg. ↑).

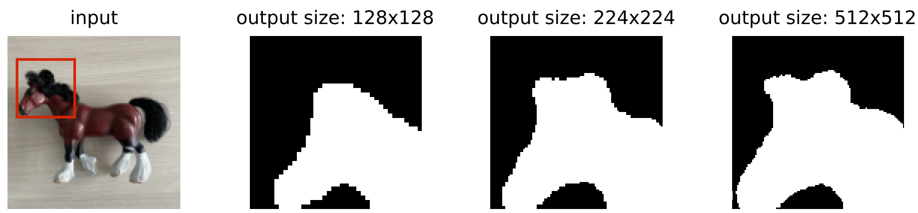

Figure 4: After training, our adapter can be queried to output different resolutions from the same backbone. Here we use a DINOv2 backbone trained on Pascal VOC.

takes positions as inputs (similar to implicit neural fields). Instead of using the standard query position grid for a 224 px output, we can sample different query coordinates at test time (Fig. 4).

4.4 Adapter result correlate with downstream tasks

To assess the reliability of our findings, we consider previous work in object-centric representation learning and a classification-based evaluation (Fig. 5 left and middle). Object-centric learning shares the goal of disentangling instances but tries to achieve this through specific model architectures whereas we evaluate model-agnostic features for instance-specific signals. Relating to the instance clustering performance by Aydemir et al. we find an almost linear relationship between their and our scores. Comparing with DINOSAUR (Seitzer et al., 2023), we find the ordering of the scores to be consistent. Note, no statement regarding better performance can be made since the evaluation protocols do not match. The evaluation of Goldblum et al. shares the goal of characterizing current backbones with our work but put more emphasis on out-of-distribution and backbone architecture. We found our results (Fig. 5, right) to be consistent with their COCO fine-tuning scores, except for DINO. Note, to obtain their scores, a resource-intensive object detection training is required. Summed up, our method can be used to obtain similar insights on relative backbone performance to more complex evaluations, but much faster.

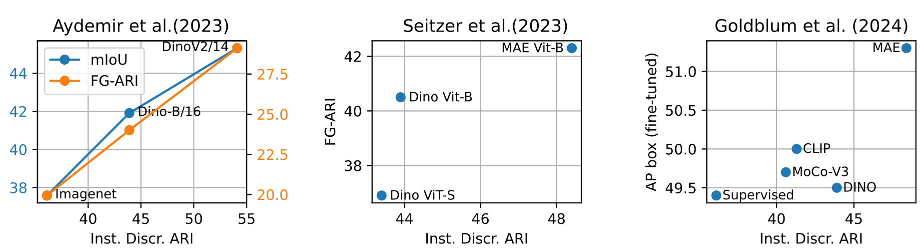

Figure 5: Our findings are consistent with previous work where object-centric learning is used (left and middle) and on fine-tuned Faster R-CNN detection heads (right).

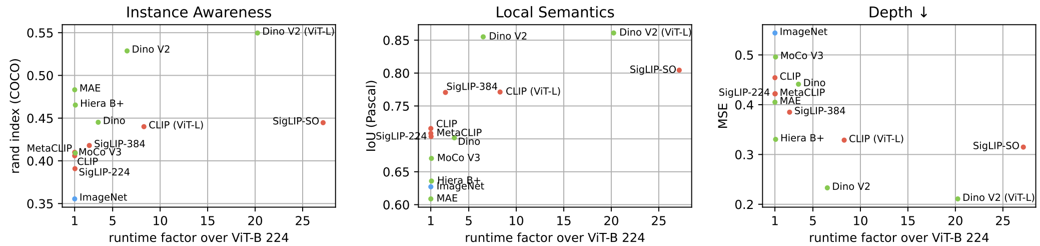

Figure 6: Inference speed over performance on three tasks, relative to the ViT-B/16 with 224px image input (the fastest and most frequent architecture in our evaluation).

4.5 Speed-performance tradeoff

In Fig. 8 we report inference speed over performance. We measure the time to ran ten batches with eight samples each in inference mode (i.e. without gradient computation). The fastest model in our evaluation set is the ViT-B/16 at a resolution of 224, therefore we indicate the factor by which the runtime is extended with respect to this model. For example, the slowest model, SigLIP-SO requires 26 times as long as the reference model.

5 Discussion

In this work we proposed the masked cross-attention adapter, a fast and parameter-efficient method for evaluating backbones. For example, our standard training for an adapter on a ViT-B/16 224 pixel backbone on Pascal VOC adds less than 200,000 parameters and takes less than 16 minutes (using a single Nvidia RXT2080 GPU). We use this method to systematically analyze common vision backbones with respect to the three complementary aspects: instance awareness, depth and local semantics. Our results suggest that DINOv2 is a highly capable backbone, it is the best ViT-B model across all experiments. Using DINOv2 with a ViT-L backbone performs improves performance further but at a three times longer runtime. Classic supervised pre-training on ImageNet results in fairly poor performance. Based on our results, a promising direction would be to use DINOv2-like objective function for pre-training in object detection, where ImageNet is currently the standard. Also VLMs and MLLMs could benefit from adopting the DINOv2 loss into their training algorithms. We identified a trend that local semantics is better captured by language-vision models while reconstruction-based self-supervised learning appears to have better instance awareness. We also found the input image resolution to play a significant role, despite decoupling input and output resolution.

For practitioners, DINOv2 is a natural choice if enough compute is available. For compute-constrained cases the decision is more complex. Vision-language models generally perform well on tasks that require local semantics, while for instance discrimination reconstruction-based self-supervised learning methods excel. We plan to retain an online leaderboard where new backbones can easily be incorporated to help tracking future progress of dense prediction performance.

Limitations While we use a fairly small decoder (in terms of parameters), even this decoder can have inductive biases and favor certain backbones such that results might get distorted. Using more complex task heads would enable more complex feature processing. In fact, this could be the reason why SAM performs comparably poorly in instance awareness in our hands. A more direct comparison to object-centric approaches would be interesting, but is challenging as these approaches explicitly encode objects (e.g. in attention slots) which can be compared to ground truth. The current selection of tasks we evaluated is limited to three broad categories and a few instances of those. Adding additional task categories (e.g. as in Taskonomy (Zamir et al., 2018)) would be desirable for a more detailed characterization of backbones.

REFERENCES

Görkay Aydemir, Weidi Xie, and Fatma Guney. Self-supervised object-centric learning for videos. *Advances in Neural Information Processing Systems*, 36:32879–32899, 2023.

Randall Balestriero and Yann LeCun. Learning by reconstruction produces uninformative features for perception. *arXiv preprint arXiv:2402.11337*, 2024.

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. 2023.

Deblina Bhattacharjee, Sabine Süsstrunk, and Mathieu Salzmann. Vision transformer adapters for generalizable multitask learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19015–19026, 2023.

Tyler Bonnen, Stephanie Fu, Yutong Bai, Thomas O'Connell, Yoni Friedman, Nancy Kanwisher, Joshua B Tenenbaum, and Alexei A Efros. Evaluating multiview object consistency in humans and image models. *arXiv preprint arXiv:2409.05862*, 2024.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

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

Jieneng Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Vitamin: Designing scalable vision models in the vision-language era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12954–12966, 2024.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

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

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

Aniket Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Mike Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. Zero-shot object-centric representation learning. *arXiv preprint arXiv:2408.09162*, 2024.

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. *ICLR*, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21795–21806, 2024.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *International Conference on Learning Representations (ICLR)*, 2024.

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, Yinfei Yang, Joshua M. Susskind, and Alaaeldin El-Nouby. Multimodal autoregressive pre-training of large vision encoders, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GkJiNn2QDF.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

Micah Goldblum, Hossein Souri, Renkun Ni, Manli Shu, Viraj Prabhu, Gowthami Somepalli, Prithvijit Chattopadhyay, Mark Ibrahim, Adrien Bardes, Judy Hoffman, et al. Battle of the backbones: A large-scale comparison of pretrained models across computer vision tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

LAION-AI. Clip benchmark. https://github.com/LAION-AI/CLIP_benchmark, 2022.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pp. 280–296. Springer, 2022.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35: 109–123, 2022.

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

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

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

David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Semi-convolutional operators for instance segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 86–102, 2018.

Maxime Oquab, Timoth'ee Darcet, Th'eo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12179–12188, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. *International Conference on Machine Learning (ICML)*, 2023.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2304–2314, 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. *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, et al. Bridging the gap to real-world object-centric learning. 2023.

Jan-Martin O Steitz and Stefan Roth. Adapters strike back. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23449–23459, 2024.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

Marissa A. Weis, Laura Pede, Timo Lüddecke, and Alexander S. Ecker. Self-supervised representation learning of neuronal morphologies. *ArXiv*, abs/2112.12482, 2021. URL https://api.semanticscholar.org/CorpusID:245424769.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

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. 2024.

Xuan Yang, Liangzhe Yuan, Kimberly Wilber, Astuti Sharma, Xiuye Gu, Siyuan Qiao, Stephanie Debats, Huisheng Wang, Hartwig Adam, Mikhail Sirotenko, et al. Polymax: General dense prediction with mask transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1050–1061, 2024.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4578–4587, 2021.

Bruce XB Yu, Jianlong Chang, Haixin Wang, Lingbo Liu, Shijie Wang, Zhiyu Wang, Junfan Lin, Lingxi Xie, Haojie Li, Zhouchen Lin, et al. Visual tuning. *ACM Computing Surveys*, 56(12): 1–38, 2024.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. 2022.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2024.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

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*, pp. 11975–11986, 2023.

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.

A Appendix

A.1 Implementation

We use the Adam optimier with a learning rate of 0.001, except for boundary prediction and depth where it is set to 0.002. We use 8 attention heads in all models. On COCO and Pascal we use the validation sets for testing, while model selection is carried out on a separate part of the training set via validation loss.

A.2 Comparison with Open-vocabulary Segmentation

We report the performance of our method and state-of-the-art open vocabulary segmentation methods on Pascal VOC2012 (with background, also called VOC-21) in Tab. 6. Please note, this is not a fair comparison as our method was trained on Pascal VOC2012.

	Pascal VOC-21	COCO-Stuff
CaR	67.6	-
SCLIP	61.7	22.4
MaskCLIP	38.8	16.7
MAXA-ImageNet	62.7	31.8
MAXA-DinoV2	85.5	45.6

Table 6: Comparison with open-vocabulary segmentation.

A.3 Relation Semantic Segmentation and Image Classification Performance

In Fig. 7 we show the semantic segmentation performance in relation to timm leaderboard**?** ImageNet accuracy.

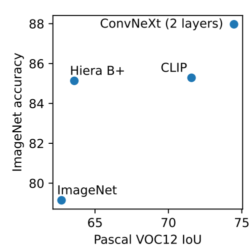

Figure 7: Comparison of semantic segmentation with ImageNet accuracy.

A.4 Inference Speed

In Fig. 8 we show the inference speeds relative to the fastest model (ViT-B/16).

A.5 Feature Visualization

We visualize the backbones output features (Fig. 9 by reducing the number of feature dimensions to three and interpreting these three dimensions as RGB color.

A.6 CNN Code

Below the source code of the CNN baseline in the adapter is shown. D_{CNN} is referred to by `dim_internal`.

```
def __init__(self):
    ...
```

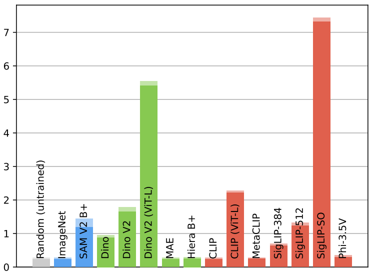

Figure 8: Inference speed of selected methods. Light bars on the top represent runtime of the MAXA adapter.

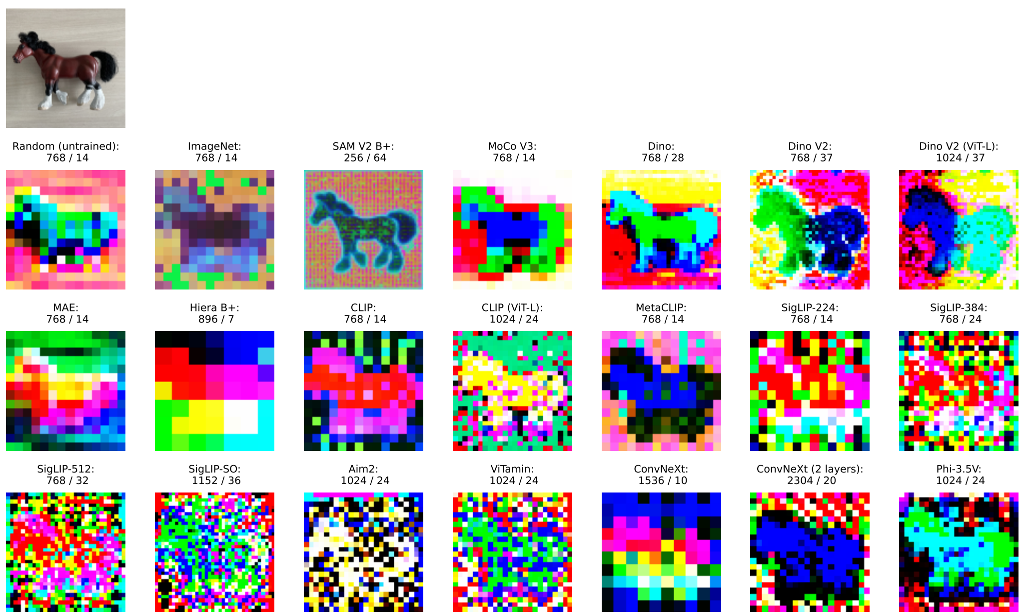

Figure 9: PCA Visualization of the backbone features. The numbers below the backbone names denote feature dimension / spatial size of the feature volume.

```
        self.cnn = nn.Sequential(
            nn.Conv3d(p.dim, dim_interm, kernel_size=(1,3,3), padding=(0,1,1)),
            nn.ReLU(),
            nn.ConvTranspose3d(dim_interm, dim_interm, kernel_size=(1, p.up[0], p.up[0]), stride=(1,p.up[0],p.up[0])),
            nn.ReLU(),
            nn.Conv3d(dim_interm, dim_interm, kernel_size=(1,3,3), padding=(0,1,1)),
            nn.ReLU(),
            nn.ConvTranspose3d(dim_interm, dim_interm, kernel_size=(1, p.up[1], p.up[1]), stride=(1,p.up[1],p.up[1])),
            nn.ReLU(),
            nn.Conv3d(dim_interm, n_classes, kernel_size=1)
        )
        self.skip_cnn = nn.Sequential(
            InterpolateFac(8, bilinear=True),
            nn.Conv3d(p.dim, n_classes, kernel_size=1),
        )

    def forward(self, x):
        ...
        x = self.skip_cnn(x) + self.cnn(x)
```