Scalable Pre-training of Large Autoregressive Image Models

Alaaeldin El-Nouby [1] **Michal Klein** [1] **Shuangfei Zhai** [1] **Miguel Angel Bautista** [1] **Vaishaal Shankar** [1]
Alexander Toshev [1] **Joshua M Susskind** [1] **Armand Joulin** [† 1]

https://github.com/apple/ml-aim

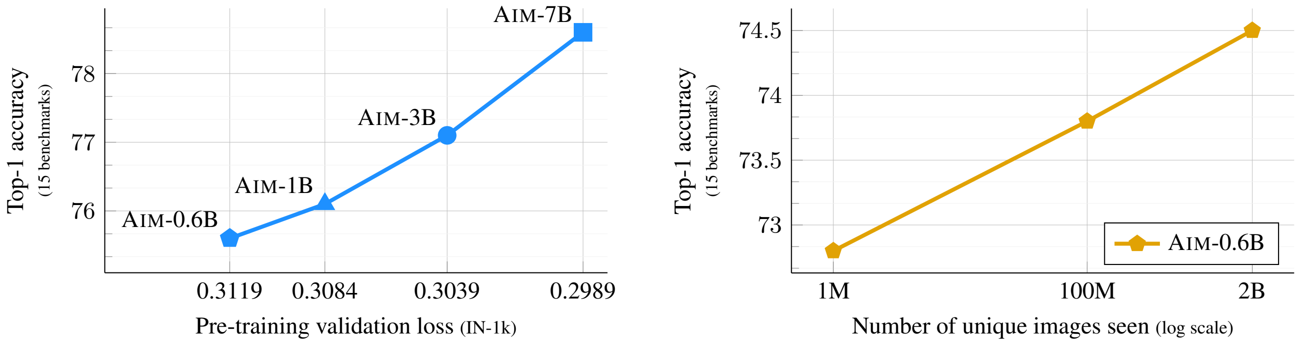


Figure 1: AIM scaling behavior (Left) As we scale the capacity of AIM, we observe improved performance for the pre-training objective which directly correlates with stronger downstream performance. (Right) AIM exhibits stronger downstream performance when trained using larger sets of uncurated web data [Gadre et al., 2023; Fang et al., 2023]. The downstream performance is the average attentive probe top-1 accuracy over a diverse set of **15 image recognition benchmarks**. All models are trained for the same number of updates.

Abstract

This paper introduces AIM, a collection of vision models pre-trained with an autoregressive objective. These models are inspired by their textual counterparts, i.e., Large Language Models (LLMs), and exhibit similar scaling properties. Specifically, we highlight two key findings: (1) the performance of the visual features scale with both the model capacity and the quantity of data, (2) the value of the objective function correlates with the performance of the model on downstream tasks. We illustrate the practical implication of these findings by pre-training a 7 billion parameter AIM on 2 billion images, that achieves 84.0% on ImageNet-1k with a frozen trunk. Interestingly, even at this scale, we observe no sign of saturation in performance, suggesting that AIM potentially represents a new frontier for training large-scale vision models. The pre-training of AIM is similar to the pre-training of LLMs, and does not require any image-specific strategy to stabilize the training at scale.

1. Introduction

Pre-training task agnostic models has become the standard in Natural Language Processing with the recent revolution of large language models (LLMs) [Radford et al., 2019; Brown et al., 2020; Touvron et al., 2023]. These models can solve complex reasoning tasks from a few examples [Brown et al., 2020], follow instructions [Ouyang et al., 2022], and now serve as the engine of widely used AI assistants such as ChatGPT. A key factor contributing to their success is the ability to consistently improve as the capacity (*i.e.* number of parameters) or the amount of pre-training data [Radford et al., 2019] increases.

The scaling behavior of these models is remarkable for two key reasons. First, even though these models are trained with a simple objective – predicting the next word in a sentence given its past – they can learn intricate patterns over long contexts. Second, the scalability of this autoregressive objective is observed when used in conjunction with certain architectures, and in particular Transformers [Vaswani et al., 2017], highlighting the potential synergy between autoregressive pre-training and this architecture.

These observations raise the question of whether the success of scaling Transformers with an autoregressive objective is exclusive to text. This is particularly significant considering that none of the aforementioned elements are inherently specific to language modeling. Autoregressive objectives take their roots in the data compression [Shannon,

[1] Apple. [†] Work done while with Apple. Now at Google DeepMind. Correspondence to: <alaaeldin_ali@apple.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1951], and similar approaches have been investigated in audio [Oord et al., 2018] and images [Van den Oord et al., 2016; Chen et al., 2020a]. The Transformer architecture has also been successfully used in other domains, in particular, computer vision with the success of the Vision Transformers (ViT) [Dosovitskiy et al., 2021]. Therefore, as a first step towards generalizing the findings of LLMs, we explore if training ViT models with an autoregressive objective leads to competitive performance, in terms of learning representations, with the same scaling ability as LLMs.

In this paper, we introduce Autoregressive Image Models (AIM), an autoregressive approach for large-scale pre-training for visual features. We revisit prior work in autoregressive representation learning such as iGPT [Chen et al., 2020a] using a modern toolset that includes vision transformers, collections of large-scale web data [Gadre et al., 2023; Fang et al., 2023], and recent advances in LLM pre-training [Touvron et al., 2023; Hoffmann et al., 2022]. Additionally, we introduce two architectural modifications to adapt autoregressive pre-training to visual features. First, instead of restricting the self-attention to be causal as is typically the case of LLMs, we adopt prefix attention, as in T5 [Raffel et al., 2020b]. This choice enables moving to bidirectional attention during downstream tasks. Second, we use a heavily parameterized token-level prediction head, inspired by the heads used in contrastive learning [Chen et al., 2020b]. We observe that this modification significantly improves the quality of the subsequent features. Overall, the training of AIM is similar to the training of recent LLMs and does not rely on any stability-inducing techniques [Touvron et al., 2021b; Huang et al., 2016; Dehghani et al., 2023] that supervised [Touvron et al., 2021b; Dehghani et al., 2023] or self-supervised [Bao et al., 2022; Oquab et al., 2023] methods need.

We provide a study of a series of models, ranging from 600M to 7B parameters pre-trained using 2B uncurated images with permissive licenses. AIM exhibits strong scaling behavior w.r.t. the model size as shown in Figure 1 where higher capacity models achieve better downstream performance, measured as the average accuracy over 15 image recognition benchmarks. More importantly, there is a correlation between the value of our objective function on a validation set and the quality of the subsequent frozen features. This observation confirms that the autoregressive objective is adequate for the training of visual features. Furthermore, we observe consistent improvement in downstream performance as we train on more images, with no sign of saturation. Overall, these observations are aligned with the previous studies on scaling large language models.

2. Related Work

Autoregressive models. While most of the literature on autoregressive models come from language modeling [Mikolov et al., 2010; Bengio et al., 2000; Radford et al., 2019] or speech [Oord et al., 2018; 2016], few works have explored the potential of this approach for images [Larochelle & Murray, 2011; Parmar et al., 2018; Van den Oord et al., 2016; Salimans et al., 2017; Chen et al., 2020a; Parmar et al., 2018]. Of particular interest, Van den Oord et al. [2016] show that using an architecture adapted to images, *e.g.*, a convolution network, significantly improved over autoregressive models built with more generic architecture [Van Den Oord et al., 2016], *e.g.*, a recurrent network [Elman, 1990]. Parmar et al. [2018] further improve the quality of autoregressive models by adopting transformer architecture. More recently, Chen et al. [2020a] have shown that scaling with more compute leads to continuous improvements. Our work follows this line of research, and we benefit from training on significantly more data, further improvement in architecture design [Dosovitskiy et al., 2021], training [Touvron et al., 2021a; 2023] and understanding of the scaling law [Hoffmann et al., 2022]. Concurrent to our work, Bai et al. [2023] demonstrate the effectiveness of large-scale autoregressive vision models for in-context pixel prediction tasks (*e.g.*, segmentation, depth estimation).

Self-supervised pre-training. Pre-training vision models on datasets of images without supervision has been a fruitful area of research in recent years [Doersch et al., 2015; Misra & Maaten, 2020; Zhang et al., 2016; Bojanowski & Joulin, 2017; Gidaris et al., 2018; Zhou et al., 2022; Dosovitskiy et al., 2014]. Different approaches have been employed, focusing on various proxy tasks for feature learning. For example, Noroozi & Favaro [2016] learn to re-arrange the order of shuffled image patches. Some other works have relied on clustering [Bautista et al., 2016; Caron et al., 2018; Yan et al., 2020; Caron et al., 2021]. Another popular approach involves the use of a contrastive objective, resembling predictive coding, where the objective is to identify each image [Chen et al., 2020b; He et al., 2020]. Most recent contrastive approaches include DINO [Oquab et al., 2023], BYOL [Grill et al., 2020] or iBot [Zhou et al., 2022]. In a similar vein, some works have proposed predictive approaches [Assran et al., 2023; Bardes et al., 2022] or a form of feature whitening [Zbontar et al., 2021]. Closer to our approach are works inspired by BERT [Devlin et al., 2018] where patches are masked and predicted with an autoencoder in either their discrete [Bao et al., 2022] or pixel [He et al., 2022] form.

Other generative pre-training. Autoregressive modeling is a form of generative modeling, and few other generative approaches have been considered to learn visual features. The first category leverages some form of autoencoding where the pretext task corresponds to some denoising task. For instance, the noise can be salt-and-pepper [Vincent et al., 2010] or masking [Pathak et al., 2016; Bao et al., 2022]. Another line of work leverages Generative Adver-

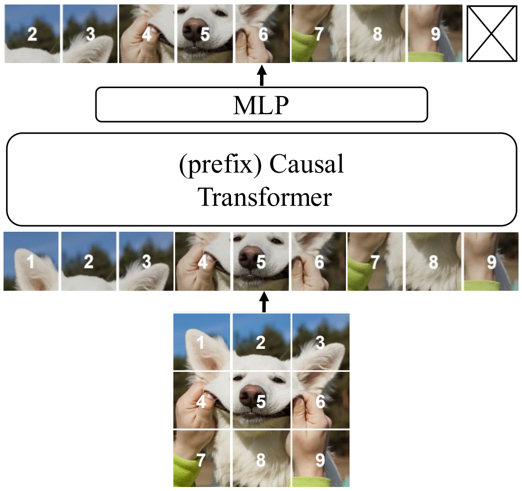


Figure 2: AIM pre-training overview.. Input images are split into non-overlapping patches and embedded linearly following Dosovitskiy et al. [2021]. The patch features are fed to a transformer in which the self-attention operation is causally masked to prevent attending to preceding positions. Afterward, a heavily parameterized MLP processes each of the patch features independently and finally projects it to pixel space. The targets correspond to the input sequence shifted one position to the left, requiring the model to predict the next patch in raster order.

sarial Networks (GANs) [Goodfellow et al., 2014]. Most notably, BigGAN [Brock et al., 2018] trains a large GAN and re-uses the image discriminator to produce image features. More recently, DiffMAE [Wei et al., 2023] used diffusion models to learn image features.

Pre-training at scale. There are numerous works on scaling the pre-training of visual features without supervision [Oquab et al., 2023; Singh et al., 2023; Tian et al., 2021; Goyal et al., 2019; 2022; Caron et al., 2019]. The most salient work in this area is DINOv2 where it produces the best self-supervised features by scaling the iBot method [Zhou et al., 2022] on a private dataset of 142M images. This work concludes that a carefully tuned contrastive method scales reasonably well, but they do not exhibit the scaling law that we observe with language modeling. They also rely on an intricate implementation of contrastive learning to avoid the pitfalls described by Chen et al. [2021]. In parallel, Singh et al. [2023] study the scaling of Masked Autoencoders [He et al., 2017]. While the study focuses on a weakly-supervised setup, it does not showcase strong improvements to the self-supervised pre-training by scaling the data to billions of images. In contrast, we observe a clear benefit of scale on the quality of our features, even at a scale of a few billions of parameters and billions of images.

3. Pre-training Dataset

We pre-train our models on the DFN dataset introduced by Fang et al. [2023]. This dataset is composed of a larger collection of 12.8B image-text pairs [Gadre et al., 2023] filtered from Common Crawl. The data has been preprocessed to remove NSFW content, blur faces, and reduce contamination by deduplicating against the evaluation sets. A data filtering network [Fang et al., 2023] ranks the samples in the 12.8B collection according to the alignment score between images and their corresponding caption. A subset of 2B images, called DFN-2B, has been extracted from the DataComp 12.8B dataset [Gadre et al., 2023] by keeping the top 15% samples. Note that, other than the privacy and safety filters, this process does not include any additional curation based on the image content. Since our pre-training does not require text, our method could be pretrained using larger image collections that are not paired with captions or have low image-text alignment such as the rest of DataComp 12.8B.

Motivated by the common practice in LLM pretraining [Touvron et al., 2023] of oversampling high-quality data sources such as Wikipedia and Books, we sample images from DFN-2B with a probability of $p = 0.8$ and sample images from ImageNet-1k with a probability of $p = 0.2$. We refer to such dataset as DFN-2B+.

4. Approach

4.1. Training Objective

Our training objective follows that of a standard autoregressive model applied on a sequence of image patches. More precisely, an image x is split into a grid of K non-overlapping patches x_k, $k \in [1, K]$, which collectively form a sequence of tokens. We assume that the sequence order is fixed across all images, and we use a raster (row-major) ordering by default unless otherwise specified. Given the above order, the probability of an image can be factorized as a product of patch conditional probabilities:

$$P(x) = \prod_{k=1}^{K} P(x_k \mid x_{<k}), \tag{1}$$

where $x_{<k}$ denotes the set of the first $k-1$ patches, and is the context used to predict the k^{th} patch. As opposed to language modeling, our sequences have a fixed length of K that fits in memory and hence we do not need to truncate the context length. The training loss over a set $\mathcal{X}$ of images is then defined as the negative log-likelihood (NLL):

$$\sum_{x \in \mathcal{X}} \sum_{k=1}^{K} -\log P(x_k \mid x_{<k}).$$

Minimizing this objective over an infinite amount of images, with no further assumptions, is theoretically equivalent to learning the true underlying image distribution.

Prediction loss Our training objective gives rise to certain variants of losses, each corresponding to a choice of the distribution $P(x_k \mid x_{<k})$. By default, we adopt a normalized

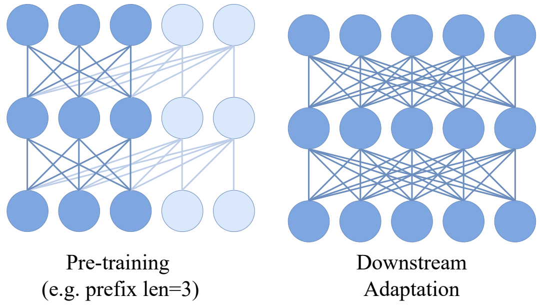


Figure 3: Prefix causal attention. During pre-training we uniformly sample a prefix length S. The attention for the first S patches are set to be bidirectional and loss is only computed for the remaining patches in the image. During adaptation to downstream tasks, this allows us to drop the attention causal mask, improving the downstream performance.

Model	#Params	Hidden size	Layers	LR	#Patches	Batch size
AIM-0.6B	0.6B	1536	24	$1e^{-3}$	0.5T	4096
AIM-1B	1.2B	2048	24	$1e^{-3}$	1.2T	4096
AIM-3B	2.7B	3072	24	$1e^{-3}$	1.2T	4096
AIM-7B	6.5B	4096	32	$1e^{-3}$	1.2T	4096

Table 1: Model specifications. We provide the embedding dimension, number of layers, and parameter count for all AIM variants. We also provide the learning rate and batch size during pre-training. For AIM with 1B parameters and higher, the pre-training process involves 1.2M iterations, which corresponds to 1.2 trillion patches, or 5B images, seen during pre-training.

pixel-level regression loss similar to He et al. [2022]. This loss corresponds to setting $P(x_k \mid x_{<k})$ as Gaussian distributions with a constant variance. Namely, given $\hat{x}_k(\theta)$ as the prediction of the k^{th} patch from a network parameterized with θ, and x_k as its corresponding ground-truth value, our objective is to minimize the sum ℓ_2 squared distance between the prediction and the ground-truth:

$$\min_{\theta} \frac{1}{K} \sum_{k=1}^{K} \|\hat{x}_k(\theta) - x_k\|_2^2. \tag{2}$$

We also consider a cross-entropy loss with patches converted to discrete tokens using an offline tokenizer. Our ablation studies show that these designs work, although they do not produce as strong features as the pixel-wise loss.

4.2. Architecture

As the backbone, we adopt the Vision Transformer architecture (ViT) [Dosovitskiy et al., 2014]. For scaling in the model capacity, we follow the common practice in language modeling and we prioritize expanding width rather than depth [Radford et al., 2019; Touvron et al., 2023]. In Table 1, we provide an overview of the design parameters of AIM, including its depth and width, as well as the amount of data and optimization scheme for each model capacity. The overall model is illustrated in Figure 2.

During pre-training, we apply causal masks to the self-attention layers to model the probability of a patch given the preceding patches. More precisely, for a self-attention layer, the embedding for the i^{th} patch is computed by:

$$y_i = \sum_{k=1}^{K} a_{ik} v_i, \tag{3}$$

where a_{ik} is the attention weight and v_k the value embedding. To enforce the desired constraints, we utilize a causal mask for the attention weights, where $a_{ik} = 0$ for $k > i$, and $\sum_{k=1}^{K} a_{ik} = 1$. This approach enables us to process the image with a single forward pass during training, without incurring additional computational overhead.

Prefix Transformer. The autoregressive objective in pre-training requires a causal mask in the self-attention operation. However, this differs from the standard usage of ViT models in downstream tasks, where bidirectional self-attention is employed. This discrepancy leads to a decrease in performance, irrespective of whether the causal mask is retained during downstream adaptation or not (as shown in the ablations presented in Table 3). To address this issue, we propose to consider the initial patches of the sequence, referred to as the prefix, as a context for predicting the remaining patches following the PrefixLM formulation of Raffel et al. [2020a]. The prefix patches are excluded from the autoregressive prediction and therefore are not constrained to be causal. More precisely, we select a prefix length of size $S \in [1, K-1]$, and remove the causal mask, *i.e.*, $a_{i,k} > 0$ for $k < S$. This modification helps the model to work in the absence of causal masking, allowing it to be removed during downstream adaptation. This approach improves the performance of the model in downstream tasks and eliminates the need for architectural changes to ViT. Figure 3 illustrates the difference between causal and prefix attention.

MLP prediction heads. It is a common practice to adopt certain prediction heads during pre-training, which are discarded when transferring to downstream tasks [Chen et al., 2020b; 2021; Caron et al., 2020; 2021; Grill et al., 2020]. The purpose of these heads is to prevent the trunk features from becoming too specialized in the pre-training objective, thus enhancing their suitability for downstream transfer. We opt for a simple design where we use N blocks of MLP on top of the final transformer layer, processing each patch independently. We observed that this design strikes a good balance between performance and the additional costs incurred during pre-training.

Straightforward implementation. It is worth noting that AIM does not require particular optimization stability-inducing mechanisms such as LayerScale [Touvron et al., 2021b], stochastic depth [Huang et al., 2016], QK-Norm [Dehghani et al., 2023], or freezing the patch projector [Chen et al., 2021]. These mechanisms have been crucial for the success of other methods, either supervised

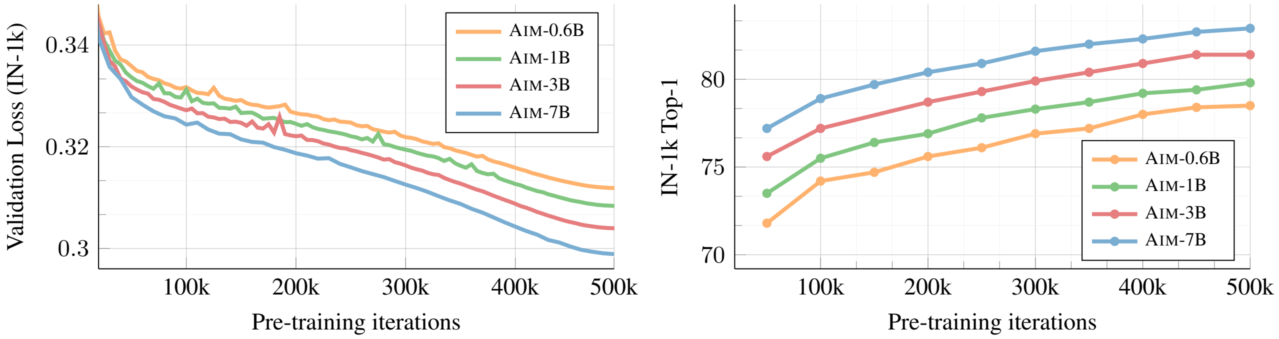


Figure 4: AIM pre-training across model sizes. We observe a clear improvement in the performance of the pre-training objective with increasing the capacity of AIM. Moreover, the downstream performance (IN-1k top-1) is monotonically improving for higher capacity models as well as with longer pre-training. We do not observe clear signs of plateauing during pre-training even after training for 500k iterations, indicating that AIM can benefit from even longer pre-training schedules.

or self-supervised. On the contrary, we observe that AIM scales using the same set of optimization hyperparameters across model sizes with no further tuning (see Table 1).

We add sinusoidal positional embeddings [Vaswani et al., 2017] to the input patches before the transformer and before the MLP head. We use a standard expansion ratio of 4 for all the MLP blocks in the trunk and the head. We drop the bias term for simplicity, and unlike the original ViT, we do not append a classification token to the input. By default, we use 12 blocks for the MLP head for all model capacities. The pixel targets are normalized per patch before the loss computation following He et al. [2022]. We train our model using `bfloat16` precision. We detail the hyperparameters used for pre-training and downstream adaptation in Appendix D.

Downstream adaptation. Pre-training large-scale models is a resource-intensive process, and even fine-tuning them is demanding. Consequently, we focus on scenarios where all model weights are fixed for downstream tasks. In this context, we only train a classification head, which mitigates the risk of overfitting on small downstream datasets and significantly reduces the adaptation cost.

Unlike contrastive learning, our loss is computed independently for each patch. This means that our pre-training does not incorporate any notion of global image descriptors, and hence, we do not have any image level token. While some methods rely on global average pooling to build a global feature from the patch features, we find that our approach, along with other generative approaches like MAE, benefit more from an attention pooling operation [Lee et al., 2019] placed before the linear classifier. Other works [Yu et al., 2022; Touvron et al., 2021b; Anonymous, 2023; Chen et al., 2023] have adopted attention pooling to improve performance with minimal overhead.

Specifically, given a set of patch features $P = \{p_i \mid 1 \leq i \leq K\}$, we compute a global descriptor $\hat{p}$ through multi-head attention pooling over the patch features as:

$$\hat{p_h} = \sum_{i=1}^{K} \frac{\exp(q_h^T W_h^k p_i)}{\sum_{j=1}^{K} \exp(q_h^T W_h^k p_j)} W_h^v p_i, \tag{4}$$

where for each attention head $h = \{1, ..., H\}$, $W_h^k, W_h^v \in R^{d_h \times d}$ correspond to the key and value weight matrices, respectively; q_h is a learnable query vector. And we obtain the pooled feature as $\hat{p} = [p_1, ..., p_H], \hat{p} \in R^d$, which serves as the input to the linear classifier. By default, we set the number of heads $H = \frac{d}{d_h}$, which makes the total number of learnable parameters $2d^2 + d$, a negligible cost compared to the main model size. Including this attention pooling makes the entire operation not strictly linear, and, therefore we refer to it as "Attentive Probe". Nevertheless, the advantages of linear probing, e.g., low additional parameter count and a reduced risk of overfitting, are preserved with this probe.

5. Results

5.1. Impact of scaling

We measure the impact when scaling our approach in terms of parameters and training data. In particular, we investigate whether there is a correlation between the pre-training objective and the downstream performance across benchmarks. We also look at the effect of scaling on the value of the loss function. For all of these experiments, we report the value of our loss function on the validation set of IN-1k.

Loss and performance during training. In Figure 4, we measure for each model the value of the pre-training loss and the classification accuracy on the validations set, as a function of the number of training iterations. We observe that both probes improve accordingly during the entire training, showing that optimizing our objective directly results in better downstream performance.

Number of parameters. We observe that the pre-training loss and the accuracy of the downstream task improve as we scale the capacity of our models. This is consistent with

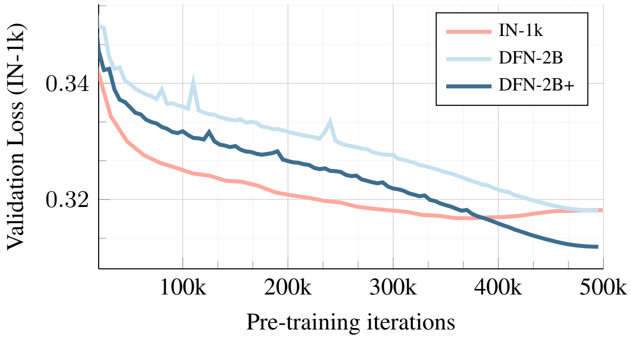


Figure 5: Dataset impact on pre-training performance. On the one hand, pre-training using IN-1k leads to overfitting, even for the AIM-0.6B model. On the other hand, pre-training using the uncurated DFN-2B dataset prevents overfitting but converges to a similar point due to the distributional shift. Pre-training on DFN-2B+ leads to the best performance.

the trend observed in LLMs and can be directly attributed to the optimization of our objective function, which in turn leads to learning stronger representations.

Number of images. In Figure 5, we show the progression of the validation loss as we pre-train on either a small curated dataset of 1M images, *i.e.*, IN-1k, or a larger set of 2B images, *i.e.* DFN-2B+. It is not surprising that training on IN-1k leads rapidly to a low validation loss as measured on the same distribution. However, this loss deteriorates at the end of the training, indicating an overfitting to the training data. When training on the uncurated DFN-2B dataset, the model starts from a higher validation loss but the loss continues to decrease with no sign of overfitting. When the same dataset is augmented with a small amount of IN-1k data, as detailed in § 3, we observe further improvement in the performance that eventually surpasses pre-training on IN-1k. We confirm that the resulting model also leads to a better downstream performance in Table 2.

pre-training dataset	IN-1k	DFN-2B	DFN-2B+
attentive	73.5	74.5	**75.6**

Table 2: Dataset impact of downstream performance (15 benchmarks). The behavior in Figure 5 is consistent with the downstream performance where we observe that using a data mixture of DFN-2B and IN-1k results in the best performance.

Compute-optimal pre-training. Since we do not observe signs of overfitting when we train using the DFN-2B+ dataset, we proceed to examine the impact of extending the length of our pre-training schedule. In Figure 6, we study the impact of increasing the length of the pre-training schedule from 500k to 1.2M iterations, *i.e.*, 2B to 5B images seen during pre-training. We observe that models pre-trained with a longer schedule achieve significantly lower validation loss. This suggests that one can improve the performance of AIM either by increasing the model capacity or by pre-training for longer schedules. Interestingly, we find that lower-capacity models trained for a longer schedule achieve comparable validation loss to higher-capacity models trained for a shorter schedule while using a similar amount of FLOPs. This finding is consistent with Hoffmann et al. [2022] and implies that AIM could follow similar scaling laws. However, we defer further investigations in this aspect for future work.

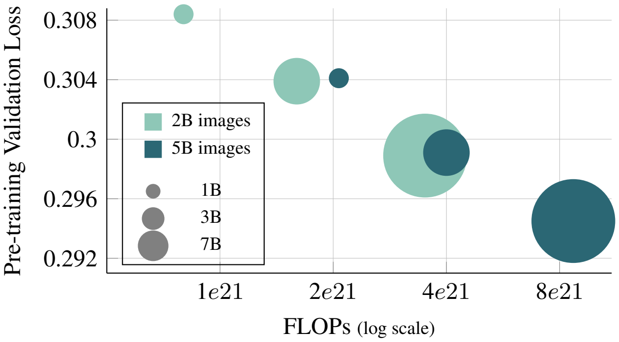


Figure 6: Scaling in FLOPs. That total number of FLOPs during training correlates with the final validation loss, suggesting a compute-driven scaling law similar to Hoffmann et al. [2022].

5.2. Architecture and Design

In this section, we investigate the impact of some variations in our model and training objective. These ablations are conducted using an AIM-0.6B model, which has been pre-trained and evaluated on the IN-1k dataset. The results of these ablations are presented in Table 3.

Targets and objective (a). We explore various potential representations for the target patches. One approach is to utilize the raw pixel values, and training the model with mean squared error (MSE) regression loss. A second option, proposed by He et al. [2022], involves using per-patch normalized pixel values instead of the raw signal, with the same MSE loss. Finally, another option is to use a discretized representation of the patches, either using k-means or a discrete VAE [Ramesh et al., 2021; Van Den Oord et al., 2017]. In this case, the model is trained using a cross-entropy objective similar to language modeling. Our experiments show that AIM performs best when using the MSE objective with normalized pixel values.

Autoregression pattern (b). Autoregressive pre-training typically follows a specific order of traversal to facilitate the prediction of the next token. In the case of language, the traversal pattern is clear, as text is read and written one word at a time in a sequential manner (*e.g.* left to right for English). However, for images, determining the traversal pattern is less obvious. We explore various deterministic patterns, including raster, spiraling out, checkerboard, and randomly pre-sampled patterns. Detailed examples of each pattern are found in Appendix B. Even though our model performs reasonably well with each pattern, we observe

target	pixels	norm. pixel	KMeans	dVAE
linear	67.5	**70.0**	66.6	64.0
attentive	76.2	**78.2**	75.9	74.5

(a) Targets.

pattern	raster	spiral	checkerboard	random
linear	**69.5**	67.7	68.2	65.8
attentive	**77.4**	76.3	76.0	75.7

(b) Autoregression Pattern (causal).

crop scale	0.08	0.4	1.0
linear	68.4	**70.0**	49.6
attentive	77.7	**78.2**	63.5

(c) Crop Scale.

pre-training attn.	causal		prefix	
inference attn.	causal	bidirectional	causal	bidirectional
linear	69.5	30.9	68.4	**70.0**
attentive	77.4	52.3	76.9	**78.2**

(d) Attention Structure.

head	None	MLP	Transformer
linear	64.0	70.0	**70.5**
attentive	75.4	78.2	**78.5**

(e) Head Design.

architecture	deep	wide
linear	68.8	**70.0**
attentive	77.9	**78.2**

(f) Architecture.

Table 3: Ablations We investigate various design choices of AIM. We use an AIM-0.6B model that is pre-trained and evaluated using IN-1k. We report the linear and attentive probing results. The default settings for AIM used for the main results are highlighted in gray .

that the raster pattern leads to significantly higher performance.

width	512	1024	2048
linear	69.4	69.6	**70.0**
attentive	77.7	78.1	**78.2**

(a) MLP width.

depth	6	8	12
linear	65.3	68.1	**70.0**
attentive	76.2	77.1	**78.2**

(b) MLP depth.

Table 4: MLP design. We vary the capacity of the MLP head by changing the number of MLP blocks (*i.e*. depth) or the embedding size (*i.e*. width). Downstream performance improves with more capacity in either width or depth, but depth has more impact.

	autoregressive	masked image modeling	
		ratio=50%	ratio=75%
attentive	**78.2**	70.3	77.8

Table 5: Autoregressive *vs*. Masking We evaluate the IN-1k performance of the autoregressive objective, compared to the masking objective [Devlin et al., 2018; Bao et al., 2022]. We keep all the other architectural and optimization components fixed. We observe that, under the same pre-training settings, the performance of the autoregressive objective outperforms masking.

To gain deeper insights into this result, we examine the difficulty of predicting patches along sequences for each pattern. This can be done by measuring the loss value per patch as we progress along a sequence, as illustrated in Figure 7. Our observation is that patterns that present a more uniform distribution of difficulty across patches result in superior models, as compared to patterns where the prediction becomes progressively easier as the sequence unfolds. We attribute this to the difficulty of predicting patches throughout the sequence that forces the model to retain more information about the image. This leads to better patch features, and consequently, to better image representation as a whole.

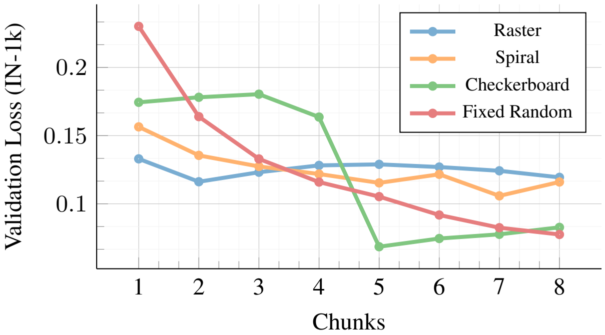


Figure 7: Autoregression patterns We explore a number of patterns for the autoregressive traversal of an image. The set of image patches is broken into equal-sized chunks and the validation loss is measured per chunk. We observe that the way the task difficulty is distributed across chunks varies strongly among patterns.

Cropping scale (c). We explore the impact of the information content of each patch by adjusting the lower bound of the cropping scale. On the one hand, opting for a cropping scale that is too small leads to an easier next-patch-prediction task as neighboring patches' similarity increases. On the other hand, using a large cropping scale can lead to severe overfitting unless the dataset size is sufficiently large. Since this study is conducted using IN-1k, we observe a clear drop in performance due to overfitting.

Causal *vs*. Prefix Attention (d). We measure the impact of incorporating prefix attention during pre-training, as opposed to using standard causal attention. We observe that pre-training with causal self-attention produces models that are effective in downstream transfer tasks only when the causal mask is preserved. Such models experience a significant decline in performance when bidirectional attention is employed. However, pre-training with prefix attention produces models that operate effectively in causal and bidirectional modes. Notably, the best performance is achieved when combining prefix attention during pre-training with bidirectional attention during downstream adaptation.

Head design (e). We consider different types of heads on top of the backbone to make predictions at the pixel level. Using no heads (*i.e. None*) performs reasonably well, but adding an MLP further improves the quality of the backbone. Interestingly, replacing the MLP with a full-fledged transformer of the same depth and width only yields a marginal performance improvement but at a significantly higher computational cost. Therefore, we opt to use an

Model	Arch.	Data	IN-1k	iNAT-18	Cifar10	Cifar100	Food101	DTD	Pets	Cars	iWildCam	Cam17	PCAM	RxRX1	EuroSAT	fMoW	Infographic	Avg
DINO	ViT-B/8	IN-1k	80.1	66.0	97.8	87.3	89.5	78.4	92.3	89.2	58.5	93.7	90.2	6.1	98.2	57.0	41.1	75.0
iBOT	ViT-L/16	IN-21k	83.5	70.5	99.2	93.3	93.5	81.6	92.8	90.8	61.8	94.5	90.0	5.9	98.0	60.3	47.7	77.6
DINOv2	ViT-g/14$_{516}$	LVD	86.4	84.5	99.6	95.2	96.3	86.3	96.4	95.6	68.2	96.5	90.7	8.0	98.6	66.7	58.8	81.9
BEiT	ViT-L/14	IN-21k	62.2	44.4	94.4	78.7	79.0	64.0	80.9	69.5	52.0	92.8	88.2	4.2	97.5	47.7	25.9	65.4
MAE	ViT-H/14	IN-1k	80.9	64.6	97.1	85.8	90.2	78.1	95.0	93.7	58.1	94.2	89.8	5.4	98.1	56.9	42.2	75.3
	ViT-2B/14	IG-3B	82.2	70.8	97.5	87.3	93.4	81.2	95.1	94.9	57.8	94.4	90.3	7.3	98.2	60.1	50.2	77.4
AIM-0.6B	ViT-H/14	DFN-2B+	78.5	64.0	97.2	86.8	90.1	80.1	93.0	93.0	57.9	94.3	90.0	7.8	98.4	58.3	45.2	75.6
AIM-1B	ViT-1B/14		80.6	67.2	98.2	88.3	91.6	81.8	93.4	93.9	58.6	94.5	90.0	9.0	98.6	59.8	47.5	76.9
AIM-3B	ViT-3B/14		82.2	69.7	98.4	89.9	92.7	81.9	94.1	93.8	58.8	94.3	90.4	9.7	98.5	60.9	48.9	77.6
AIM-7B	ViT-7B/14		82.4	70.9	98.6	90.0	93.1	82.3	93.8	92.1	59.5	93.6	90.7	10.1	98.6	61.7	49.6	77.8
AIM-7B†	ViT-7B/14	DFN-2B+	84.0	75.5	98.9	91.8	94.1	85.6	95.4	95.0	61.4	94.2	90.5	8.4	98.5	63.5	57.7	79.6

Table 6: Downstream evaluation with a frozen trunk. We assess the quality of AIM features by evaluating against a diverse set of 15 image recognition benchmarks. AIM and the baseline methods are evaluated using attentive probing with a frozen trunk. AIM models exhibit a strong performance across all benchmarks, especially the AIM-7B. AIM outperforms all other methods, using joint-embedding or generative approaches, except for DINOv2 which utilizes higher-resolution images. †: Extracting features from the 20$^{\text{th}}$ layer instead of the last (32$^{\text{nd}}$), see Table 7 for more details.

MLP head in our approach. We hypothesize that these heads specialize in capturing the low-level signals necessary for accurate pixel-level prediction. By incorporating a head, the trunk can learn higher-level features that are more suitable for downstream transfer. A similar design was employed for contrastive learning to prevent the backbone from specializing in predicting specific image transformations [Chen et al., 2020b].

Deeper *vs*. Wider architecture (f). We present the design specifications of AIM in Table 1, outlining its width and depth. Unlike the original design of ViT [Dosovitskiy et al., 2021], where the depth is scaled more rapidly than the width, we adopt a scaling strategy similar to that of Llama [Touvron et al., 2023]. This allows us to scale our model more gracefully while maintaining a reasonable depth. We validate the effectiveness of a wider architecture in Table 3f. Our findings indicate that even for the relatively small-scale AIM-0.6B model, a wider architecture not only delivers strong performance but also improves training stability. This observation supports the notion that some of the insights gained from training LLMs can be similarly applied to other domains.

Attentive *vs*. Linear probe. For all ablations we report the linear and attentive probing results. We observe that, consistently across all experiments, attentive pooling provides a significant boost to performance as it allows for a more nuanced aggregation of local features circumventing one of the main weaknesses of generative pre-training: the absence of an image-level global descriptor.

Structure of the MLP. The MLP plays an important role as ablated in Table 3e. In Table 4, we further investigate the capacity of the MLP head and how it impacts downstream performance. We vary the capacity of the head by either changing the number of MLP blocks or their width. By default, we use a head of 12 blocks and an embedding dimension of 2048. First, we observe that increasing the capacity of the MLP either through depth or width leads to consistent improvement in the downstream performance. Second, we find that increasing the number of MLP blocks, with a fixed width, leads to a larger improvement compared to increasing the width for a fixed depth. Interestingly, we could not find a point where increasing the MLP capacity failed to yield further improvements. We did not explore higher capacities beyond those reported in Table 4 as it would lead to models with disproportionate head and trunk capacity.

5.3. Pre-training objective

Autoregressive *vs*. Masking We conduct a comparison between our architecture trained with an autoregressive objective and the masking objective popularized by BERT [Devlin et al., 2018] for language, and by BEiT and MAE for vision. It is important to note that we applied the masking objective in the same setting as AIM, thereby isolating the impact on the performance of the pre-training objective from other design choices that differ between AIM and other approaches. In the masking baseline, we randomly sample masks and replace the masked patches with learnable mask tokens. In Table 5, we show that AIM performs better with an autoregressive objective than a masking objective. This is consistent with the results reported by Chen et al. [2020a], providing evidence that our improvements stem from using the autoregressive objective.

5.4. Comparison with other methods

In Table 6, we compare the attentive probing performance of AIM to other state-of-the-art methods across a set of 15 diverse benchmarks that are detailed in Appendix A.

Generative methods. AIM provides a strong performance compared to its generative counterparts. AIM outperforms BEiT [Bao et al., 2022] with a large margin. Additionally, AIM-0.6B provides a better performance, averaged across all benchmarks, compared to MAE-H [He et al., 2022] which has an equivalent capacity. Moreover, we compare against the MAE-2B [Singh et al., 2023] model which has been pre-trained on IG-3B, a private dataset of 3 billion images from Instagram. We find that both AIM-3B and AIM-7B outperform MAE-2B, with AIM-7B exhibiting a particularly large improvement. It is worth noting that, similar to AIM, two other generative approaches, BEiT and MAE, benefit from attentive probing, thereby narrowing the gap between generative and joint embedding methods.

Joint embedding methods. AIM provides a competitive performance with joint embedding methods such as DINO [Caron et al., 2021], iBOT [Zhou et al., 2022], and DINOv2 [Oquab et al., 2023]. In terms of average accuracy across all benchmarks, AIM outperforms DINO and iBOT. However, it falls behind DINOv2 which achieves its results by evaluating with higher-resolution inputs. Note that AIM attains such competitive performance using higher capacity trunks. Nevertheless, AIM's pre-training is significantly simpler and can be trivially scaled in terms of parameters and data, yielding consistent improvements. On the contrary, state-of-the-art joint embedding methods like DINOv2 heavily rely on a number of tricks, such as multi-crop augmentation, KoLeo regularization, LayerScale, Stochastic Depth, schedules for teacher momentum and weight decay, and high-resolution fine-tuning in order to achieve strong performance.

Extracting stronger features. We observe that higher-quality features can be extracted from shallower layers compared to the last layer's features. This is likely due to the generative nature of the pre-training objective that is inherently different than the discriminative downstream tasks and therefore, the features with the highest semantic content do not necessarily concentrate around the last layer. In Table 7, we report the IN-1k top-1 accuracy for features extracted from the last layer compared to the layer with the highest performance.

	AIM-0.6B	AIM-1B	AIM-3B	AIM-7B
last layer	78.5	80.6	82.2	82.4
best layer	**79.4**	**82.3**	**83.3**	**84.0**

Table 7: Feature extraction. The highest quality features after AIM pre-training typically reside in shallower layers than the last. Extracting features from earlier layers leads to a non-negligible boost to the recognition performance on IN-1k.

5.5. Low-Rank Adaptation

In addition to frozen-trunk evaluation, we examine Low-Rank Adaptation (LoRA) [Hu et al., 2021], a popular and efficient finetuning method. We report the results of LoRA fintuning of AIM in Table 8. We observe that LoRA is compatible with AIM, leading to a large boost in performance compared to frozen-trunk evaluation. For example, AIM-7B improves by 3.9% (compared to the last layer's performance) while finetuning only 0.1% percent of the trunk parameters.

	AIM-0.6B	AIM-1B	AIM-3B	AIM-7B
attentive	78.5	80.6	82.2	82.4
LoRA (rank=8)	81.0	83.6	85.5	86.3

Table 8: Low-rank adaptation (IN-1k). AIM is compatible with LoRA showing large gains compared to frozen-trunk evaluations.

6. Discussion

In this paper, we presented a scalable method for pre-training vision models without supervision. We employed a generative autoregressive objective during pre-training and proposed several technical contributions to better adapt it for downstream transfer. Consequently, we observed a number of desirable properties for our Autoregressive Image Models. First, the capacity of our models can be effortlessly scaled to 7 billion parameters using a vanilla transformer implementation, without resorting to stability-inducing techniques or extensive adjustments of hyperparameters for each model scale. Second, AIM's performance on the pre-training task has a strong correlation with downstream performance. Third, AIM achieves strong performance across 15 recognition benchmarks, outperforming prior state-of-the-art methods like MAE and significantly narrowing the gap between generative and joint embedding pre-training approaches. Finally, we did not observe any clear signs of saturation as we scale either in terms of parameters or data, suggesting that there is a potential for further performance improvements with larger models trained for even longer schedules. We hope that AIM serves as a seed for future research in scalable vision models that *effectively leverage uncurated datasets* without any bias towards object-centric images or strong dependence on captions.

7. Limitations.

AIM excels in its seamless scalability and its effective utilization of large volumes of uncurated image data. However, alternative methods can offer different trade-offs. MAE [He et al., 2022] provides high sample efficiency and can learn good representations using a small amount of pre-training data, reducing the risk of overfitting [El-Nouby et al., 2021] in contrast to our approach. Contrastive methods [Oquab et al., 2023; Zhou et al., 2022; Caron et al., 2021] currently result in stronger representations for a given model size compared to generative approaches such as MAE and AIM, but pose significant challenges in terms of scalability and loss tractability due to the complexity of their objective.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

The authors would like to thank Brandon McKinzie, Samira Abnar, Preetum Nakkiran, and Jiatao Gu for valuable feedback throughout the project. We thank Edouard Grave and Hervé Jegou for their inspiring discussions during the earlier stages of the project. We thank Marco Cuturi, James Thornton, Pierre Ablin, and Eugene Ndiaye for their support and for many fruitful discussions throughout the project. Finally, we would like to thank the entire Machine Learning Research team at Apple for many helpful discussions and assistance with infra and data.

References

Anonymous. V-JEPA: Latent video prediction for visual representation learning. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. 5

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023. 2

Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A., Darrell, T., Malik, J., and Efros, A. A. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023. 2

Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Bejnordi, B. E., Lee, B., Paeng, K., Zhong, A., et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018. 13

Bao, H., Dong, L., and Wei, F. BEiT: Bert pre-training of image transformers. In *ICLR*, 2022. 2, 7, 9

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 2

Bautista, M. A., Sanakoyeu, A., Tikhoncheva, E., and Ommer, B. Cliquecnn: Deep unsupervised exemplar learning. *Advances in Neural Information Processing Systems*, 29, 2016. 2

Beery, S., Cole, E., and Gjoka, A. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020. 13

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000. 2

Bojanowski, P. and Joulin, A. Unsupervised learning by predicting noise. In *International Conference on Machine Learning*, pp. 517–526. PMLR, 2017. 2

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 13

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *preprint arXiv:2005.14165*, 2020. 1

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2959–2968, 2019. 3

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 4

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 4, 9

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *ICML*, 2020a. 2, 8, 14

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020b. 2, 4, 8

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 3, 4

Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., and Wang, J. Context autoencoder for self-supervised representation learning. *ICCV*, 2023. 5

Christie, G., Fendley, N., Wilson, J., and Mukherjee, R. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 13

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *CVPR*, 2014. 13

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A. P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al. Scaling vision transformers to 22 billion parameters. In *ICML*. PMLR, 2023. 2, 4

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009. 13

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 2, 7, 8

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014. 2, 4

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 8

El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., and Grave, E. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 9

Elman, J. L. Finding structure in time. *Cognitive science*, 14(2): 179–211, 1990. 2

Fang, A., Jose, A. M., Jain, A., Schmidt, L., Toshev, A., and Shankar, V. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023. 1, 2, 3

Gadre, S. Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023. 1, 2, 3

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

Goyal, P., Mahajan, D., Gupta, A., and Misra, I. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019. 3

Goyal, P., Duval, Q., Seessel, I., Caron, M., Singh, M., Misra, I., Sagun, L., Joulin, A., and Bojanowski, P. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022. 3

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020. 2, 4

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, 2017. 3

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2, 4, 5, 6, 9

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2017. 13

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2, 6

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 9

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *ECCV*, 2016. 2, 4

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009. 13

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 29–37. JMLR Workshop and Conference Proceedings, 2011. 2

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 5

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, 2010. 2

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 2

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 2

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. In *NeurIPS*, 2018. 2

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision, 2023. 2, 3, 9, 14

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. 1

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *CVPR*, 2012. 13

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. Image transformer. In *ICML*, 2018. 2

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 13

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 1, 2, 4

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020a. 4

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020b. 2

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021. 6

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 2

Shannon, C. E. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951. 1

Singh, M., Duval, Q., Alwala, K. V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer, C., Girshick, R., et al. The effectiveness of mae pre-pretraining for billion-scale pretraining. *arXiv preprint arXiv:2303.13496*, 2023. 3, 9

Taylor, J., Earnshaw, B., Mabey, B., Victors, M., and Yosinski, J. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *ICLR*, 2019. 13

Tian, Y., Henaff, O. J., and van den Oord, A. Divide and contrast: Self-supervised learning from uncurated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10063–10074, 2021. 3

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021a. 2

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021b. 2, 4, 5

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2, 3, 4, 8

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 2

Van Den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016. 2

Van Den Oord, A., Vinyals, O., et al. Neurips. *Advances in neural information processing systems*, 2017. 6

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 13

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017. 1, 5

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pp. 210–218. Springer, 2018. 13

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 2

Wei, C., Mangalam, K., Huang, P.-Y., Li, Y., Fan, H., Xu, H., Wang, H., Xie, C., Yuille, A., and Feichtenhofer, C. Diffusion models as masked autoencoders. *arXiv preprint arXiv:2304.03283*, 2023. 3

Yan, X., Misra, I., Gupta, A., Ghadiyaram, D., and Mahajan, D. ClusterFit: Improving Generalization of Visual Representations. In *CVPR*, 2020. 2

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 5

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 2

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 14

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *ECCV*, 2016. 2

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 2, 3, 9

A. Datasets

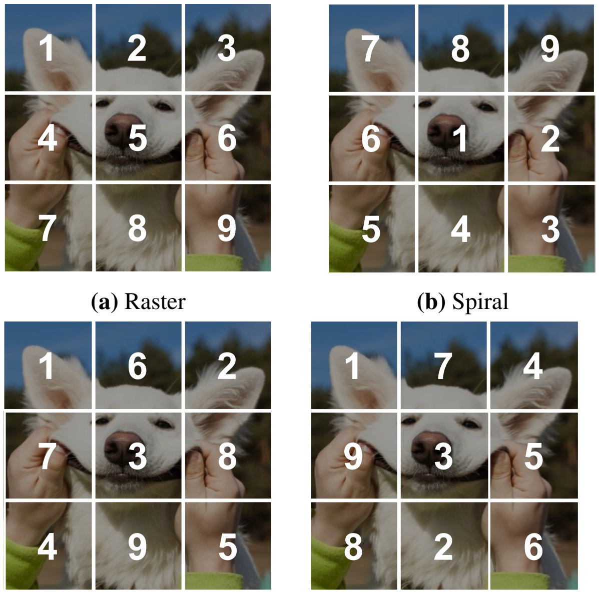
To assess the effectiveness and general applicability of the learned representations by AIM, we measure its recognition accuracy on a varied collection of 15 benchmarks in Table 6. The specifics of each benchmark can be found in Table 9. These benchmarks include datasets for tasks such as fine-grained recognition, medical imaging, satellite imagery, images in natural environments, and infographic images.

Dataset	train	test	classes
Imagenet-1k [Deng et al., 2009]	1,281,167	50,000	1000
iNAT-18 [Van Horn et al., 2018]	437,513	24,426	8142
CIFAR-10 [Krizhevsky et al., 2009]	50,000	10,000	10
CIFAR-100 [Krizhevsky et al., 2009]	50,000	10,000	100
Food101 [Bossard et al., 2014]	75,750	25,250	101
DTD [Cimpoi et al., 2014]	3,760	1,880	47
Pets [Parkhi et al., 2012]	3,680	3,669	37
Cars [Krause et al., 2013]	8,144	8,041	196
iWildCam [Beery et al., 2020]	129,809	14961	182
Camelyon17 [Bandi et al., 2018]	302,436	34904	2
PCAM [Veeling et al., 2018]	262,144	32768	2
RxRx1 [Taylor et al., 2019]	40,612	9854	1139
EuroSAT [Helber et al., 2017]	16,200	5400	10
fMoW [Christie et al., 2018]	76,863	19915	62
Infograph [Peng et al., 2019]	36,023	15,582	345

Table 9: Evaluation benchmarks. We provide the references, the number of images in the train and test sets, and the number of categories of all the 15 recognition benchmarks used in this work.

B. Autoregression Patterns

We investigate different patterns that can be used to traverse an image during pre-training in Table 3b. All patterns used in this investigation are illustrated in Figure 8.



(a) Raster **(b)** Spiral **(c)** Checkerboard **(d)** Random

Figure 8: Autoregression patterns. We illustrate the different autoregression patterns studied in this work including raster, spiral, checkerboard, and fixed random.

C. Additional Analysis

C.1. Raster pattern validation loss

In Figure 7, we noticed that the validation loss of the raster pattern across chunks surprisingly declined for the second chunk before increasing again. We investigated this further in Figure 9 and observed that this behavior is a side-effect of using the IN-1k validation set. In particular, we observed that the top rows of the image, aside from the first one, typically have a lower loss, whether the loss is computed over the regular image or its vertically flipped counterpart.

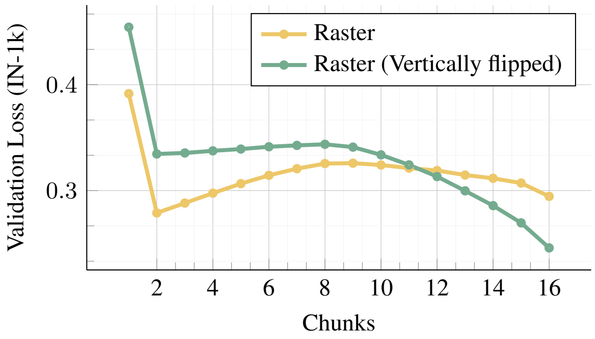


Figure 9: Raster pattern across patches. We compute the IN-1k validation loss per a chunk of 16 patches (*i.e.*, a row) for AIM-0.6B, pre-trained using a raster pattern. We measure the same loss for the vertically flipped images of the validation set. We observe that, for IN-1k validation set, the patches from the top rows in the image are easier to predict with lower loss, likely due to the concentration of background patches in that region.

C.2. Downstream performance across layers

In Tables 7 and 6, we discussed the gain in downstream performance that can be achieved by probing shallower layers in the model rather than the last. We study this in more detail in Figure 10. We find that for all AIM variants, we extract the highest quality features, with respect to the downstream transfer, from layers roughly at two-thirds of the way into the model depth. However, it is important to note that the performance of deeper layers does not experience a steep decline and continues to exhibit strong performance.

D. Hyperparameters

Pre-training. AIM models of all capacities have been pre-trained using the same set of hyperparameters that are reported in Table 10. The AIM-0.6 model however has been trained only for the shorter schedule of 500k iterations. We did not observe any instability while scaling the capacity of our model, thereby not requiring any further tuning of the optimization hyperparameters.

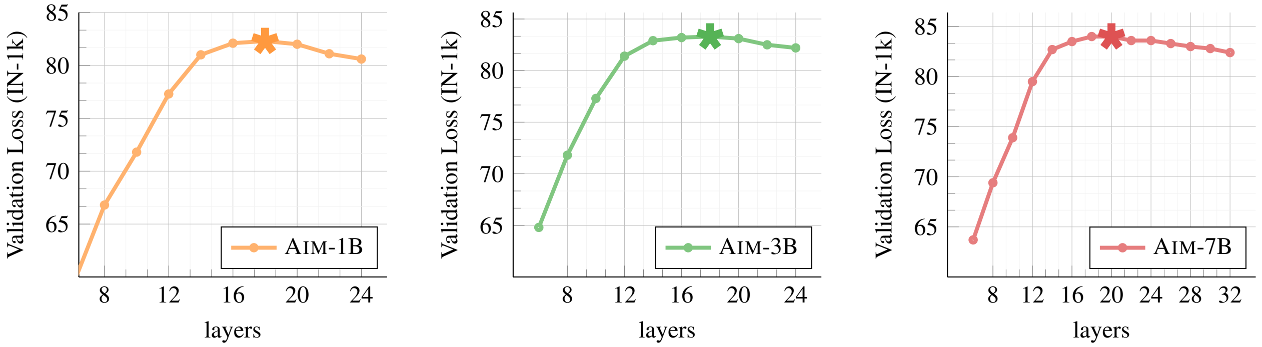


Figure 10: Downstream performance across layers. The highest quality features in terms of transfer to downstream recognition tasks can be extracted from layers different than the last, with the peak performance achieved by extracting from features roughly at two-thirds of the model depth. Deeper layers still retain a strong performance and no sharp decline is observed.

config	value
Optimizer	AdamW
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.95$
Peak learning rate	$1e^{-3}$
Minimum Learning rate	0.0
Weight decay	0.05
Batch size	4096
Patch size	(14, 14)
Gradient clipping	1.0
Warmup iterations	31,2050
Total iterations	1,250,000
Learning rate schedule	cosine decay
Augmentations:	
`RandomResizedCrop`	
`size`	224px
`scale`	[0.4, 1.0]
`ratio`	[0.75, 1.33]
`interpolation`	`Bicubic`
`RandomHorizontalFlip`	$p = 0.5$

Table 10: Pre-training hyperparameters All AIM variants of different capacities have been trained using the same set of hyperparameters detailed above.

config	IN-1k	Others
Optimizer	AdamW	
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	
Peak learning rate grid	[1, 3, 5, 10, 15, 20, 40] $\times 1e^{-4}$	
Minimum Learning rate	$1e^{-5}$	
Weight decay	0.1	
Batch size	1024	512
Gradient clipping	3.0	
Warmup epochs	5	0
Epochs	50	100
Learning rate schedule	cosine decay	
Augmentations:		
`RandomResizedCrop`		
`size`	224px	
`scale`	[0.08, 1.0]	
`ratio`	[0.75, 1.33]	
`interpolation`	`Bicubic`	
`RandomHorizontalFlip`	$p = 0.5$	
`Color Jitter`	0.3	
`AutoAugment`	`rand-m9-mstd0.5-inc1`	

Table 11: Attentive probe hyperparameters. We detail the hyperparameters used for attentive probing AIM as well as the baselines. For all experiments, we search over different learning rate values and report the best for both AIM and baselines.

Attentive Probing. Downstream evaluation for AIM and the baselines has been primarily conducted via attentive probing as described in § 4. We report the hyperparameters used to probe all methods in Table 11. For a fair comparison with other baselines, we search over different values for the learning rate and report the best performance of each method similar to [Oquab et al., 2023]. For AIM and other generative baselines, we average the features for the last 6 layers of the model before feeding them to the attention-probing head which leads to a modest gain in performance. Note that the descriptor dimensionality remains the same which is different from the practice of concatenating features similar to iGPT[Chen et al., 2020a] which indirectly inflates the capacity of the evaluation head.

Low-rank adaptation. For LoRA finetuning, we use the same hyperparameters as reported in Table 11 in addition to mixup [Zhang et al., 2018] (`alpha=0.8`). We apply LoRA adaptation, with `rank=8`, only to the parameters of the attention block. In particular, the weight matrices for the queries, values, and out projection.