CAN: A SIMPLE, EFFICIENT AND SCALABLE CONTRASTIVE MASKED AUTOENCODER FRAMEWORK FOR LEARNING VISUAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce CAN, a simple, efficient and scalable method for self-supervised learning of visual representations. Our framework is a minimal and conceptually clean synthesis of (C) contrastive learning, (A) masked autoencoders, and (N) the noise prediction approach used in diffusion models. The learning mechanisms are *complementary* to one another: contrastive learning shapes the embedding space across a batch of image samples; masked autoencoders focus on reconstruction of the low-frequency spatial correlations in a single image sample; and noise prediction encourages the reconstruction of the high-frequency components of an image. The combined approach results in a robust, scalable and simple-to-implement algorithm. The training process is symmetric, with 50% of patches in *both views* being masked at random, yielding a considerable efficiency improvement over prior contrastive learning methods. Extensive empirical studies on linear evaluation, finetuning, transfer learning, and robustness demonstrate that our approach achieves strong downstream performance. For instance, when pre-training ViT-B encoders on the curated ImageNet dataset, CAN achieves 74.8% top-1 linear probing accuracy, an absolute improvement of 6.8% over MAE and 1.3% over SimCLR with the same architecture and data augmentations. CAN is especially useful for pre-training on larger uncurated datasets such as JFT-300M: the finetuned performance on ImageNet of our ViT-L model is 85.9%, compared to 85.0% for SimCLR, and 85.4% for MAE. For linear probe on ImageNet, CAN achieves 75.4% compared to 71.8% for SimCLR and 64.1% for MAE. The overall FLOPs load is 41% *lower* than SimCLR[1].

1 INTRODUCTION

Self-supervised learning promises continued advances in the state of the art by enabling the use of increasingly large models and datasets. However, interest in larger datasets has precipitated an increased reliance on web-scraped data collection processes, which result in heterogeneous and "uncurated" datasets (Yu et al., 2022; Radford et al., 2021; Jia et al., 2021). Extreme image heterogeneity has made scaling vision models to uncurated datasets a non-trivial challenge (Tian et al., 2021; Cole et al., 2022). There are two families of self-supervised methods for images which have both proven highly effective on curated datasets (e.g., ImageNet), and are therefore natural candidates for scaling to large, uncurated data. First, masked image models such as the masked autoencoder (MAE) (He et al., 2022) are a nascent set of methods based on a mask-and-reconstruct training mechanism. This classical idea (Ballard, 1987) is enjoying a rejuvenation thanks to favourable efficiency when combined with the vision transformer architecture (Dosovitskiy et al., 2021b). Second, contrastive learning (Oord et al., 2018; Chen et al., 2020b; He et al., 2020) trains an encoder to distinguish between pairs of positive samples generated with data augmentations and negative pairs sampled at random. These two approaches have proven to be very powerful self-supervised methods.

Contrastive learning and masked autoencoders (MAE) employ very different learning mechanisms: the former train the encoder to be invariant to semantics-preserving data variations, while MAEs learn spatial statistical correlations. Furthermore, MAE methods treat each sample independently in

[1]Our code will be released at `www.xxx.yyy`.

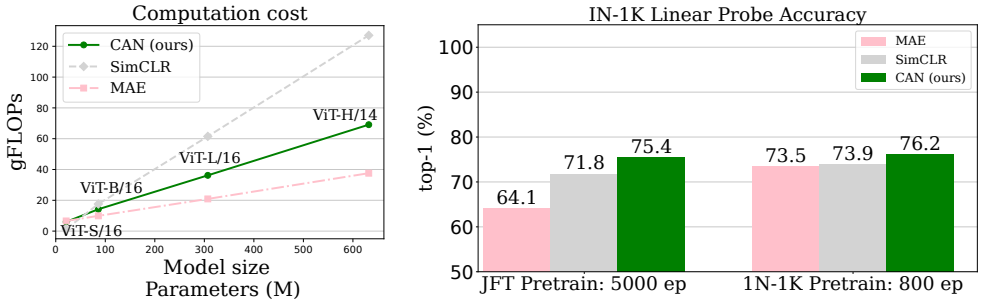

Figure 1: **Left:** CAN scales better than SimCLR since it only uses masked inputs. **Right:** Linear probe performance for ViT-L models pre-trained on JFT-300M (5000 epochs) and ImageNet (800 epochs). CAN outperforms SimCLR and MAE on ImageNet in both pre-training scenarios.

the loss function, while contrastive methods explicitly look at the relationship between all samples in the batch, by either reducing or increasing embedding distance. Given this, we hypothesize that these two approaches are *complementary*, extracting different sets of discriminative features for a given input. If this hypothesis holds, then we expect to see improved performance on various downstream tasks based on the extracted features. This paper therefore explores combining these two approaches into a unified method.

Further, inspired by advances in diffusion models (Ho et al., 2020; Song et al., 2021), we introduce a third loss based on *noise prediction* during the masked autoencoder reconstruction. We add Gaussian noise to unmasked input patches, and train the model to predict the *noise* added to each patch. Denoising encourages the encoder to extract higher-frequency information from the input, while autoencoder reconstructions tend to focus on low-frequency (blurry) information (Hou et al., 2017). This additional loss has two benefits: it improves downstream performance; and it addresses a source of wasted computation in MAE with a negligible impact on FLOPs: that reconstruction of unmasked patches is thrown away unused.

Combining the above approaches, we present CAN, a minimal fusion of contrastive learning, masked autoencoders and denoising diffusion training loss. Our method enjoys stronger performance than its constituent parts do on their own, with the benefits becoming particularly significant on more uncurated datasets such as JFT-300M, which contains 300 million highly heterogeneous images, often containing artifacts (e.g., watermarks). For instance, evaluating JFT-trained ViT-L models using the top-1 accuracy of an ImageNet-trained linear probe, MAE achieves 64.1% and SimCLR achieves 71.8%, while CAN achieves 75.4%. Our method masks 50% of patches in each view, making it significantly more scalable than prior contrastive methods that are based on two full image views. Our contributions are:

1. We present a simple model that scales in performance to JFT-300M dataset and does well on both linear probe and fine-tune regimes.
2. We show that the model is significantly (nearly 50% fewer FLOPs) more efficient than SimClr (see Figure 1(left)).
3. Our loss is more robust to distribution shifts and performs better on few-shot linear transfer than MAE or SimClr.

2 Related Work

Masked image models with Vision Transformers. The advent of the Vision Transformer (ViT) (Dosovitskiy et al., 2021b) provoked a focused effort to develop strong self-supervised learning frameworks that use ViT backbones. Works such as DINO (Caron et al., 2021) and MoCo-v3 (Chen et al., 2021b) demonstrated that techniques developed with ConvNet backbones in mind could also perform competitively using ViTs after proper tuning to suit the new architecture. ViT-specific methods have emerged since then, particularly masked image modelling (Bao et al., 2022; Chen et al., 2022; Xie et al., 2022), which takes inspiration from pre-training methods used in NLP (Devlin

et al., 2018). Notably MAE (He et al., 2022) showed that classical masked autoencoding approaches could be used to pre-train ViTs *without* passing masked tokens through the encoder. This provides a significant efficiency boost; our method similarly takes advantage of this.

Contrastive learning in computer vision. Self-supervision has received significant attention in computer vision as it offers a way to extract general purpose features without supervision. In particular, contrastive learning (Chen et al., 2020b; He et al., 2020; Oord et al., 2018; Tian et al., 2020; Chuang et al., 2020) has achieved state of the art performance by enforcing invariance to augmentations, whilst using negative samples (Robinson et al., 2021a; Ge et al., 2021) to avoid trivial solutions by spreading the embedding out uniformly on the sphere (Wang & Isola, 2020). The contrastive pre-training task is conceptually very different from masked image models such as MAE, which learn spatial statistical dependencies. Another distinction is that autoencoders encourage information preservation in latent representations, whilst contrastive learning could suppress features (Chen et al., 2021a; Robinson et al., 2021b). This leads us to hypothesize that the two approaches learn different data features, and may therefore be complementary learning mechanisms. This motivates us to study a synthesis of contrastive learning and masked image modelling, with the aim of developing a reinforced pre-training task that enjoys the merits of each.

Denoising diffusion models. Denoising autoencoders (DAE) (Vincent et al., 2010) learn to reconstruct clean data given a noisy input. By learning to map low-density data regions to high-density regions, DAE learns the shape of the data manifold. This connection was made precise by Vincent (2011), who showed that DAEs learn the score-function $s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$. This key observation underpins the significant recent advances in generative diffusion models, which use an estimate of the score-function to generate samples (Ho et al., 2020; Song et al., 2021). The recent success of DAEs in generative modelling has not yet translated to representation learning, with some exceptions (Asiedu et al., 2022; Zaidi et al., 2022). In this work we exploit a denoising autoencoder to eliminate the MAE inefficiency of reconstructing unmasked patches but never using them.

Concurrent work. A number of recent works propose approaches that combine ideas from masked image modelling and Siamese self-supervised learning. For instance, Huang et al. (2022) propose a combination of contrastive and masked reconstruction objectives using one masked view, and one full (unmasked) view. Other recent works (Tao et al., 2022; Chen et al., 2022; Assran et al., 2022) use similar asymmetric designs. The key distinction between CAN and concurrent work is that we strike a different balance between simplicity, efficiency, and performance: we focus on developing a *simple* combined method that is also efficient and symmetric: we use *two masked views* and no momentum encoder. We hope the simplicity and efficiency of CAN will make it easy to adapt and modify in future work.

3 A SIMPLE CONTRASTIVE MASKED AUTOENCODER FRAMEWORK

Our approach is a minimal synthesis of contrastive learning, the masked autoencoder (He et al., 2022), and the denoising loss used in the training of diffusion models. We focus on simplicity and scalability, aiming to design a hybrid with as few complex or costly components as possible. We also aim to minimize *wasted* computation: in particular, the MAE decoder requires reconstructions of all patches, but only those of masked patches are used in the loss. Below, first we detail the basic pipeline of generating views and passing masked inputs through the encoder and decoder. Then we explain the three different objectives we use: contrastive, reconstruction, and denoising. The penultimate section describes the combined objective, and the final section discusses scalability.

3.1 OVERVIEW OF METHOD

Given a batch of n images $\{\mathbf{x}\}_{i=1}^n$, we generate two views $\mathbf{x}_i^1, \mathbf{x}_i^2 \in \mathbb{R}^{h\times w\times 3}$ of each image without supervision using the same data augmentations as Chen et al. (2020b). Each image is then split into $T = (h/p) \times (w/p)$ patches of size $p \times p$: $\mathbf{x}_{i,\text{patch}}^1, \mathbf{x}_{i,\text{patch}}^2 \in \mathbb{R}^{T\times p\times p\times 3}$ in preparation for input to the ViT encoder. We always assume that p divides h and w. Two masks $\mathbf{M}_i^1, \mathbf{M}_i^2 \in \{0,1\}^T$ are independently generated, with a 1 in coordinate $t \in \{1, \dots T\}$ indicating that the tth patch is masked. Each patch is left unmasked independently with probability m, conditioned on always having exactly $T' = m \cdot T$ patches unmasked, which we assume is an integer. In experiments our default masking rate is $m = 50\%$ unless explicitly stated otherwise. Following He et al. (2022), only the T' *unmasked* patches are passed to the ViT encoder, which processes the two views in parallel.

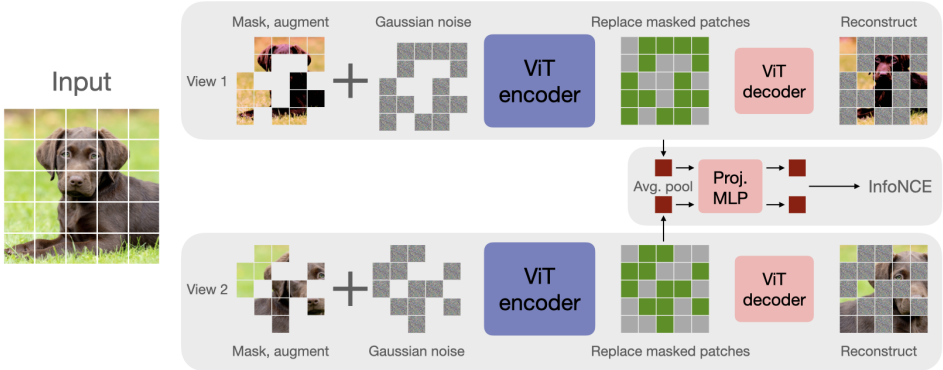

Figure 2: **The CAN framework:** Two views of an image are generated, 50% of patches randomly masked in each, and noise is added to patches. An encoder is trained to solve three tasks: 1) **Reconstruction:** encoded patches are passed to a decoder that reconstructs missing patches, 2) **Denoise:** reconstructs the noise added to unmasked patches, and 3) **Contrast:** pooled patches are passed to a contrastive loss, using in-batch samples as negatives (Chen et al., 2020b).

Masking a large fraction of patches from both views make our method much more efficient (see Table 1) than contrastive methods that use two full views, and recent works that use one full view and one masked view (Assran et al., 2022; Huang et al., 2022). Finally, we collect the embeddings of unmasked tokens $\mathbf{z}_i^1, \mathbf{z}_i^2 \in \mathbb{R}^{T' \times d}$ and reshape into $T \times d$ tensors by adding a learned `[M]` embedding to positions corresponding to masked tokens. The result is passed through a comparatively lightweight ViT decoder to produce outputs $\hat{\mathbf{x}}_i^1, \hat{\mathbf{x}}_i^2$ in image space $\mathbb{R}^{h \times w \times 3}$.

3.2 Contrastive Learning Objective

The embeddings $\mathbf{z}_i^1, \mathbf{z}_i^2 \in \mathbb{R}^{T' \times d}$ returned by the encoder are pooled via a simple mean along the first dimension to form d-dimensional embeddings, which are passed through a lightweight MLP projection head that maps into a lower dimension space $\mathbb{R}^r$, $r < d$, and normalized to unit length to produce embeddings $\mathbf{u}_i^1, \mathbf{u}_i^2 \in \mathbb{R}^r$ for $i = 1, \ldots n$. For the ith batch item we collect the other $2n - 2$ samples in-batch $\mathcal{N}_i = \{\mathbf{u}_j^1, \mathbf{u}_j^2\}_{j \neq i}$ to use as negatives, and compute the InfoNCE loss:

$$\mathcal{L}_{\text{InfoNCE}} = \frac{1}{2n} \sum_{v=1,2} \sum_{i=1}^{n} -\log \frac{e^{\mathbf{u}_i^{1\top} \mathbf{u}_i^2 / \tau}}{e^{\mathbf{u}_i^{1\top} \mathbf{u}_i^2 / \tau} + \sum_{\mathbf{u}^- \in \mathcal{N}_i} e^{\mathbf{u}_i^{v\top} \mathbf{u}^- / \tau}}$$

where $\tau > 0$ is a temperature parameter, which we set to $\tau = 0.1$ by default.

3.3 Patch Reconstruction Objective

The outputs $\hat{\mathbf{x}}_i^1, \hat{\mathbf{x}}_i^2$, $i = 1, \ldots, n$ of the ViT decoder are trained to reconstruct the missing patches of each image. Corroborating the findings of He et al. (2022), we find it best to only compute the reconstruction loss on masked patches:

$$\mathcal{L}_{\text{rec}} = \frac{1}{2n} \sum_{v=1,2} \sum_{i=1}^{n} \|\mathbf{M}_i^v \circ (\mathbf{x}_i^v - \hat{\mathbf{x}}_i^v)\|_2^2$$

where $\circ$ multiplies all pixels in the tth patch of the residual image $\mathbf{x}_i^v - \hat{\mathbf{x}}_i^v$ by $(\mathbf{M}_i^v)_t \in \{0, 1\}$. Whilst computing the loss only on masked patches gives better performance, it indicates wasted computation since the decoder also produces reconstructions for unmasked patches. To avoid waste we propose an alternative objective specifically for unmasked patches, which we discuss next.

3.4 Denoising Objective

Inspired by the significant advances in diffusion modelling using *denoising* training objectives (Ho et al., 2020; Kingma et al., 2021) and their equivalent score-based counterparts (Song et al., 2021;

Vincent, 2011) we revisit the suitability of denoising for self-supervised learning. We add independent isotropic Gaussian noise to each image $\mathbf{x}_i^v \leftarrow \mathbf{x}_i^v + \sigma_i^v \mathbf{e}_i^v$ with $\mathbf{e}_i^v \sim \mathcal{N}(\mathbf{0}, I)$ and σ_i^v uniformly sampled from an interval $[0, \sigma_{\text{max}}]$. This noisy input is masked and passed to the encoder as described in Section 3.1. When passing encoded patches to the decoder we make a small addition to the method in Section 3.1 to provide the decoder with information on the noise level σ_i^v to help it separate noise from the ground truth image. This is motivated by denoising diffusion methods, which pass both the noisy image and the noise level as inputs to the denoising model (Ho et al., 2020). We achieve this by using σ_i^v as a positional encoding in the decoder, similarly to Vaswani et al. (2017). First we produce a sinusoidal embedding of σ_i^v, which is passed through a light MLP to produce a (learnable) embedding $\mathbf{p}_i^v \in \mathbb{R}^d$, whose dimension matches the latent dimension of $\mathbf{z}_i^v \in \mathbb{R}^{T \times d}$. We add the result to each embedded token (including missing tokens `[M]`) to provide noise-level information: $(\mathbf{z}_i^v)_t \leftarrow (\mathbf{z}_i^v)_t + \mathbf{p}_i^v$ for $t = 1 \ldots, T$, and pass the result to the decoder producing $\mathbf{x}_i^v$. We define our denoising loss function, which is computed only on unmasked pixels:

$$\mathcal{L}_{\text{denoise}} = \frac{1}{2n} \sum_{v=1,2} \sum_{i=1}^{n} \|(1 - \mathbf{M}_i^v) \circ (\sigma_i^v \mathbf{e}_i^v - \hat{\mathbf{x}}_i^v)\|_2^2$$

where, $\circ$ is as defined in Section 3.2. Note that this denoising loss is extremely lightweight, introducing only a very small overhead due to the MLP. We emphasize that the reconstruction of noise patches comes at zero additional cost since the decoder produces reconstructions of all patches, both masked and unmasked, even though only reconstructions of masked patches are used in $\mathcal{L}_{\text{rec}}$. Finally, it has frequently been observed in the diffusion modelling literature that although it is equivalent to train a denoising model to estimate the noise, or to estimate the clean input itself (Vincent, 2011), there is a big empirical gap between the two, with noise prediction faring significantly better. While we do not pursue it further, we observed the same phenomenon in our testing.

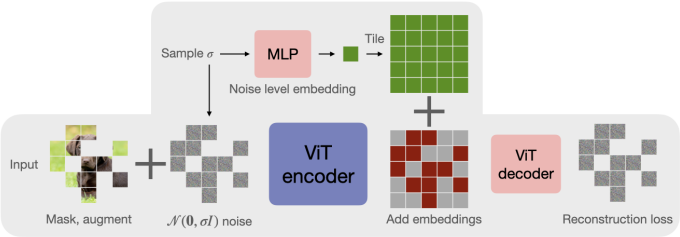

Figure 3: **Denoising:** As in denoising diffusion both the encoded patches and the noise level σ are passed to the decoder by passing σ through an MLP, and adding the result to each embedded token.

Ablation: Table 1 studies the effect of each of the components of the denoising method. We use ViT-B models trained for 100 epochs on ImageNet, and consider four settings, each adding in more parts of the method: 1) CAN with no denoising, 2) adding noise to the input only, 3) adding noise and using the denoising loss, and 4) the full method with all of the described components, including using σ_i^v as a positional encoding in the decoder. Results show that even simply adding noise as a data augmentation improves performance by 0.7%, which can be improved to 1% by also using the reconstruction loss with noise level passed as an argument. The noise level argument is necessary, as we find that the reconstruction loss without noise level argument performs worse (68.4%) noise with no reconstruction at all (68.6%).

None	+noise	+noise, +loss	Full
67.9	68.6	68.4	68.9

Table 1: Ablating components of the denoising objective. "Full" denotes the entire method as described in Section 3.4

We emphasize that the improvement from denoising comes at minimal cost to run time or memory at training (and no cost at inference), since it uses reconstructions produced by the decoder, which in the case of MAE are simply thrown away unused. Denoising prediction is efficiently scalable, and encourages the encoder to extract high-frequency features, which we hypothesize is complementary to and may be particularly useful for dense downstream tasks such as object detection and segmentation.

3.5 THE COMBINED OBJECTIVE FUNCTION

The overall CAN objective trains the encoder and decoder to optimize three losses combined:

$$\mathcal{L}_{\text{CAN}} = \lambda_{\text{InfoNCE}}\mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{denoise}}\mathcal{L}_{\text{denoise}}$$

where $0 \leq \lambda_{\text{InfoNCE}}, \lambda_{\text{rec}}, \lambda_{\text{denoise}}$, and $\lambda_{\text{InfoNCE}} + \lambda_{\text{rec}} + \lambda_{\text{denoise}} = 1$ weight the objectives. In practice we parameterize the weights by eliminating one variable using the equality constraint, taking: $\lambda_{\text{rec}} = (1 - \lambda_{\text{InfoNCE}}) \cdot \lambda$ and $\lambda_{\text{denoise}} = (1 - \lambda_{\text{InfoNCE}}) \cdot (1 - \lambda)$ where $0 \leq \lambda \leq 1$. This parameterization makes it easy to control the relative weighting between the two reconstruction losses $\mathcal{L}_{\text{rec}}, \mathcal{L}_{\text{denoise}}$ on the one hand, and the contrastive loss $\mathcal{L}_{\text{InfoNCE}}$ on the other. Empirically we find that performance is very robust to the choice of λ, and many choices of λ_{InfoNCE} also work well (see Section 5).

3.6 DISCUSSION ON EFFICIENCY

The goal of this work is to propose a conceptually minimal combined contrastive masked autoencoder approach, aiming to find better trade-offs between simplicity, efficiency, and performance. Consequently, we choose to omit a number of commonly used self-supervised learning design components. For instance, we do not use a momentum target network or multiple views (multi-crop), since they both increase memory requirements and run time. Even without these commonly used components, our minimal framework achieves very strong performance compared to prior work, and importantly improves performance over its contrastive and autoencoder constituent parts. We expect that a wide range of modifications, such as momentum target networks (He et al., 2020) and multi-crop (Caron et al., 2020), will improve performance further on top of the core method.

4 RESULTS

4.1 PRE-TRAINING ON UNCURATED DATA: JFT-300M

A key promise of self-supervised learning is to allow models to be trained on extremely large scale image datasets collected from the Web. Not only is such data likely to be *unannotated*, but also *uncurated*: images containing many objects, variable lighting, artifacts (e.g., watermarks) and so on. The large variation in images found online presents a major challenge to self-supervised learning, and it is not guaranteed that methods that work well on curated (and comparatively smaller) datasets such as ImageNet will work equally well on less curated data. To study how CAN scales to large datasets we use JFT-300M (Sun et al., 2017), a dataset of around 300 million images.

	Architecture	Epochs	IN-1K top-1
MoCLR (Tian et al., 2021)	R50	5000	67.6
BYOL (Grill et al., 2020)	R50	5000	67.9
DnC (Tian et al., 2021)	R50	1000	67.9
DnC (Tian et al., 2021)	R50	4500	70.7
MoCLR (Tian et al., 2021)	R200×2	5000	74.2
DnC (Tian et al., 2021)	R200×2	3000	**77.3**
MAE† (He et al., 2022)	ViT-L	1600	50.5
MAE† (He et al., 2022)	ViT-L	5000	64.1
SimCLR† (Chen et al., 2020b)	ViT-B	800	65.8
SimCLR† (Chen et al., 2020b)	ViT-L	800	72.6
SimCLR† (Chen et al., 2020b)	ViT-L	1600	73.1
CAN (ours)	ViT-B	800	67.1
CAN (ours)	ViT-L	800	72.8
CAN (ours)	ViT-L	1600	74.3
CAN (ours)	ViT-L	3000	75.3
CAN (ours)	ViT-L	5000	75.4

Table 2: **JFT-300M pre-training:** Comparison to the state of the art on ImageNet linear probe. Finetune results are in Figures 5, 8, and 9. CAN outperforms all methods except DnC, which uses a complicated multi-stage training process. Computation is measured as ImageNet-equivalent epochs. †Our implementation.

Setup. Training time is measured in ImageNet-equivalent epochs: 1 epoch equals $1281167/[\text{batch size}]$ steps, the number of steps in one ImageNet (IN-1K) epoch. Models are evaluated using linear probe and finetuning on IN-1K. See Appendix C for further training details and Section 5 for details on loss weights and noise level hyperparmeters.

Results. Figure 1 compares CAN to SimCLR and MAE baselines using ViT-L models. CAN achieves a much better trade-off between efficiency (measured in FLOPs) and performance using ViT-L models for all three methods: CAN uses 41% fewer FLOPs than SimCLR and consistently outperforms SimCLR and MAE: for training ViT-L models for 5000 epochs, CAN achieves an IN-1K linear probe performance of 75.4%, compared to 71.8% for SimCLR and 64.1% for MAE. The relatively poorer linear probe and transfer learning performance of MAE on JFT-300M highlights

Method	Pre-training epochs	Encoder	Additional params.	Full image	Finetune	Linear probe
from scratch†	100	ViT-B	✗	✓	79.1	—
MoCo-v3 (Chen et al., 2021b)	300	ViT-B	✓	✓	83.0	76.7
DINO (Caron et al., 2021)	1600	ViT-B	✓	✓	82.8	78.2
CIM (Fang et al., 2022)	300	ViT-B	✓	✓	83.1	—
CAE (Chen et al., 2022)	800	ViT-B	✓	✓	83.8	68.6
CAE (Chen et al., 2022)	1600	ViT-B	✓	✓	83.9	70.4
BEiT (Bao et al., 2022)	800	ViT-B	✓	✓	83.2	37.6*
SimMIM (Xie et al., 2022)	800	ViT-B	✗	✓	83.8	56.7
MAE (He et al., 2022)	800	ViT-B	✗	✗	83.1	—
MAE (He et al., 2022)	1600	ViT-B	✗	✗	83.6	68.0
CAN (ours)	800	ViT-B	✗	✗	83.4	74.0
CAN (ours)	1600	ViT-B	✗	✗	83.6	74.8
SimCLR† (Chen et al., 2020b)	800	ViT-L	✗	✓	83.4	73.9
MAE (He et al., 2022)	800	ViT-L	✗	✗	84.9	73.5
MAE† (He et al., 2022)	800	ViT-L	✗	✗	83.7	71.4
CAN (ours)	800	ViT-L	✗	✗	84.7	76.2

Table 3: **Pre-training on ImageNet-1K.** †Our implementation. *Quoted from Chen et al. (2022).

the non-triviality of scaling from IN-1K to larger datasets and suggests that while MAE is scalable for *model size*, scalability to larger *datasets* requires further study.

We also compare CAN to the current state of the art on JFT-300M pre-training in Table 2. Our best performance, 75.4% with ViT-L outperforms all methods besides DnC, with 77.3% (Tian et al., 2021) with R200$\times$2. However we note that CAN is *considerably* simpler than DnC, which involves multiple training steps including training 10 separate "expert" models (each as large as the final model), and then using MoCLR (an improvement of SimCLR that adds a momentum encoder and more), and finally using distillation to produce a single model. Our calculations suggest that training a ViT-L with CAN is about 3$\times$ faster than training the considerably smaller ResNet50 with DnC in terms of wall clock time (see Appendix B for explanation). CAN on ViT-L beats MoCLR with R200$\times$2 backbone (similar parameter counts), where we note that MoCLR performs as well or better than BYOL and MoCo-v3 on IN-1K (Tian et al., 2021).

4.2 Pre-training on ImageNet

We also evaluate our method using ImageNet (IN-1K) pre-training to verify that it remains competitive in this setting. Results in Table 3 record the top-1 accuracy on IN-1K classification of finetuned models, and linear probes. On finetuning CAN achieves 83.6% with ViT-B, outperforming other contrastive approaches such as MoCo-v3 (83.0%), and is competitive with other state-of-the-art approaches such as CAE (83.9%). The linear probe performance of CAN is 74.8% using ViT-B, which is higher than all masked image modelling methods, the best of which is CAE with 70.4% (Chen et al., 2022). CAE is only outperformed by MoCo-v3 and DINO, both of which use momentum encoders and two full image views, and in the case of DINO a further 10 multi-crop views. Additionally we report results for our implementation of MAE on ViT-L, which approximately matches the original numbers reported by He et al. (2022). We use our MAE implementation when pre-training on JFT-300M.

4.3 Few-shot learning

We evaluate suitability of CAN for few-shot learning by training linear probes, following the evaluation protocol of Dosovitskiy et al. (2021a). We use the models pre-trained on JFT-300M for 5000 epochs whose ImageNet performance is recorded in Figure 1. Results in Figure 4 for few-shot transfer learning on 9 other datasets show that the superior performance on IN-1K translates to improved performance on all other datasets. We also note that our 25-shot ViT-L models beat *full-shot* both DnC and BYOL ResNet50 models (also trained for 5000 epochs on JFT-300M) on 6 out of 8 datasets (Tian et al., 2021). See Appendix A for additional 1-shot and 5-shot results, and results on additional models, including IN-1K pre-training.

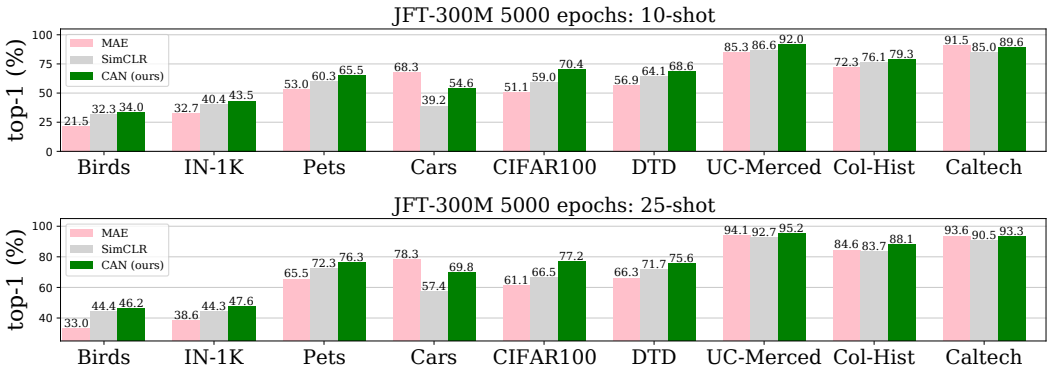

Figure 4: **Few-shot:** ViT-L models pre-trained on JFT-300M for 5000 epochs are evaluated on 9 datasets in few-shot setting (10-shot and 25-shot). CAN outperforms MAE and SimCLR on all tasks.

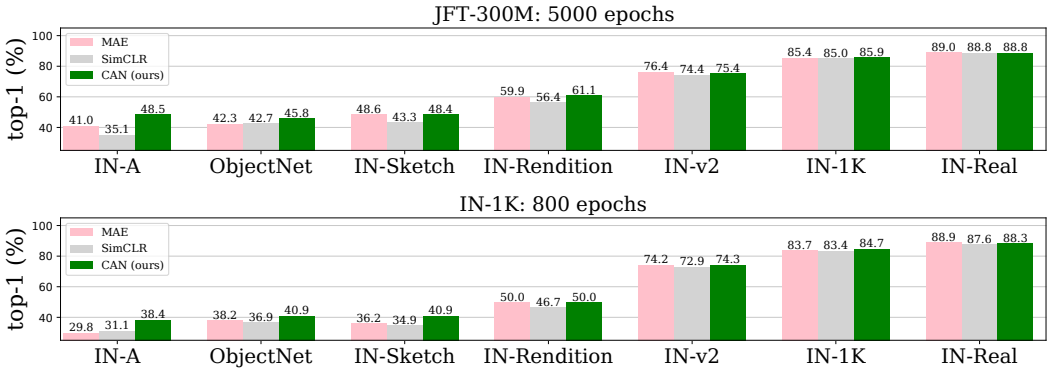

Figure 5: **Robustness:** Evaluating performance under distribution shifts with respect to models finetuned on IM-1K. Validation performance of ViT-L models is reported on 7 different datasets with the same 1000 classes as ImageNet (IN-1K). The shown IN-v2 results are for no distribution shift, i.e., both finetuned and evaluated on IN1K.

4.4 Robustness to distribution shift

Finally, we consider the robustness of CAN to distribution shifts. We use ViT-L backbones trained for 5000 epochs on JFT-300M, which have been finetuned on IN-1K. Model performance is evaluated on a number of different validation sets with the same 1000 classes as IN-1K Mao et al. (2022). Figure 5 reports results on the following 7 validation sets, which cover a large variety of distribution shifts: original IN-1K (Deng et al., 2009), IN-v2 (Recht et al., 2019), IN-ReaL (Beyer et al., 2020), IN-Adversarial (Hendrycks et al., 2021b), IN-Rendition (Hendrycks et al., 2021a), ObjectNet (Barbu et al., 2019). CAN performs particularly well under JFT-300M pre-training: it improves over SimCLR and MAE baselines in all 7 cases, often by significant margins. See Appendix A for additional results.

5 Hyperparameter analysis

We study the different components of CAN to better understand the effect of the different mechanisms, and to determine optimal parameter configurations. All ablations use ViT-B models trained for 100 epochs on IN-1K, unless explicitly said otherwise. We use the best loss weights and noise level in these experiments for all longer runs.

Complementarity of contrastive and reconstruction losses. A key hypothesis motivating our work is that contrastive learning and masked autoencoder reconstruction may not only be compatible training objectives, but are *complementary* ones. Table 4 compares the final training value of the contrastive $\mathcal{L}_{\text{InfoNCE}}$ and reconstruction $\mathcal{L}_{\text{rec}}$ when jointly trained (i.e., CAN) compared to only op-

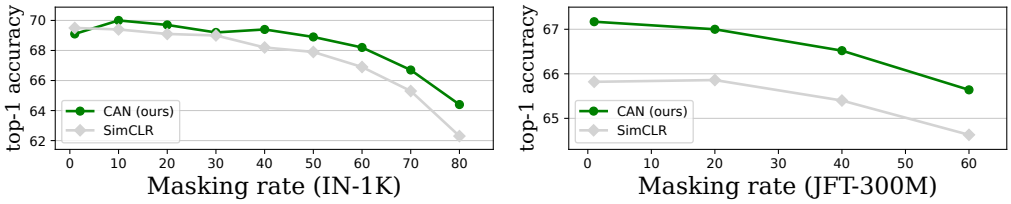

Figure 6: CAN and SimCLR with different masking rates. ViT-B models are pre-trained for 100 epochs on IN-1K, and 800 epochs on JFT-300M.

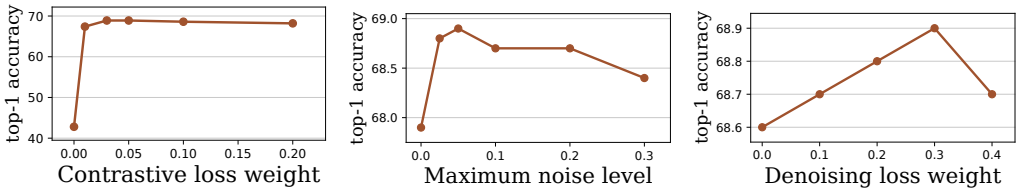

Figure 7: ViT-B models pre-trained on IN-1K for 100 epochs. **Left:** The best contrastive loss weight is small but non-negative. **Middle:** A wide range of σ_{max} values improve over no-noise. **Right:** Performance is not sensitive to the denoising loss weight.

timizing $\mathcal{L}_{\text{InfoNCE}}$ (SimCLR) or only $\mathcal{L}_{\text{rec}}$ (MAE). The results support the hypothesis: joint training achieves a lower loss on *both* objectives compared to individual training.

Masking rate. Figure 6 reports the behavior of CAN and SimCLR under different masking rates on IN-1K and JFT-300M pre-training (for JFT-300M we use 800 epochs). The performance of SimCLR decreases as the masking rate increases, suggesting that masking is not an effective data augmentation. In contrast, performance of CAN peaks at a non-zero masking rate, but at a much lower rate than the 75% used by MAE on IN-1K. This occurs since very low masking rates are preferred by the contrastive part of CAN, but severely damage the autoencoder part, which can learn trivial solutions. The considerable efficiency improvement from masking 50% of patches more than makes up for the small drop in performance for a fixed number of epochs, since many more epochs can be completed in a given wall-clock time.

Method	Contrastive loss ↓	Reconstruction loss ↓
SimCLR	9.157	—
MAE	—	0.1658
CAN (ours)	**9.143**	**0.1633**

Table 4: **Complementary training:** All methods use 50% masking for fair comparison. CAN training achieves *lower* training loss for both contrastive and reconstruction than individual training.

Contrastive loss weight. We vary the weighting λ_{InfoNCE} used to weight the contribution of the contrastive and reconstruction losses. Recall that larger λ_{InfoNCE} places higher weight on the contrastive loss. Results in Figure 7 show that the best weight is $\lambda_{\text{InfoNCE}} = 0.03$, which approximately balances the magnitudes of the two terms (see Table 4).

Denoising loss weight and noise level. We study the noise level interval $[0, \sigma_{\text{max}}]$ from which to sample input noise, and the weight λ balancing the denoising and reconstruction losses. Results in Fig 7 show that the best maximum noise level is $\sigma_{\text{max}} = 0.05$, and that similar performance is attained for a number of different weights on the denoising loss.

6 Discussion

We present CAN, a simple, efficient and scalable self-supervised method for visual representation learning. CAN combines ideas from contrastive learning, masked autoencoding, and diffusion denoising into a single high-performing method. Extensive empirical results show that CAN scales with minimal changes to the large uncurated datasets, providing a significant boost over SimCLR and MAE methods on a wide range of downstream tasks and evaluations, including linear probes, few-shot, robustness, and finetuning. Our results suggests that contrasting and reconstruction are complementary principles that can mutually reinforce one another.

REFERENCES

Emmanuel Brempong Asiedu, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Decoder denoising pretraining for semantic segmentation. *preprint arXiv:2205.11423*, 2022.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *preprint arXiv:2204.07141*, 2022.

Dana H Ballard. Modular learning in neural networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 647, pp. 279–284, 1987.

Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *Int. Conf. on Learning Representations (ICLR)*, 2022.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

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

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Mark Chen, Alec Radford, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Int. Conference on Machine Learning (ICML)*, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020b.

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 11834–11845, 2021a.

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. In *preprint arXiv:2202.03026*, 2022.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Int. Conference on Computer Vision (ICCV)*, pp. 9640–9649, 2021b.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 8765–8775, 2020.

Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14755–14764, 2022.

Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3008–3017, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. on Learning Representations (ICLR)*, 2021a.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. on Learning Representations (ICLR)*, 2021b.

Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. *preprint arXiv:2202.03382*, 2022.

Songwei Ge, Shlok Kumar Mishra, Haohan Wang, Chun-Liang Li, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume abs/2110.14189, 2021.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *preprint arXiv:1706.0267*, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022.

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

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15262–15271, 2021b.

Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. Pyramid adversarial training improves ViT performance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13419–13429, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 1133–1141. IEEE, 2017.

Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv:2207.13532v1*, 2022.

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

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21696–21707, 2021.

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

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in Adam. *preprint arXiv:1711.05101*, 2017a.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Int. Conf. on Learning Representations (ICLR)*, 2017b.

Chengzhi Mao, Lu Jiang, Mostafa Dehghani, Carl Vondrick, Rahul Sukthankar, and Irfan Essa. Discrete representations strengthen vision transformer robustness. 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *preprint arXiv:1807.03748*, 2018.

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

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Int. Conference on Machine Learning (ICML)*, pp. 5389–5400. PMLR, 2019.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *Int. Conf. on Learning Representations (ICLR)*, 2021a.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 4974–4986, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. on Learning Representations (ICLR)*, 2021.

Andreas Steiner, Alexander Kolesnikov, , Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. In *Transactions on Machine Learning Research (TMLR)*, 2021.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Int. Conference on Computer Vision (ICCV)*, pp. 843–852, 2017.

Chenxin Tao, Xizhou Zhu, Gao Huang, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. In *preprint arXiv:2206.01204*, 2022.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Europ. Conference on Computer Vision (ECCV)*, pp. 776–794, 2020.

Yonglong Tian, Olivier J Henaff, and Aäron van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *Int. Conference on Computer Vision (ICCV)*, pp. 10063–10074, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Int. Conference on Machine Learning (ICML)*, pp. 9929–9939. PMLR, 2020.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9653–9663, 2022.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. In *ECCV*, 2017.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *preprint arXiv:2205.01917*, 2022.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. *preprint arXiv:2206.00133*, 2022.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

A ADDITIONAL TRANSFER LEARNING RESULTS

We report additional results for few-shot learning and robustness.

Robustness: Sections 4.4 reports robustness results for ViT-L models pre-trained on JFT-300M for 5000 epochs, and ViT-L models pre-trained on IN-1K for 800 epochs. In both cases we report the performance of the models after finetuning on IN-1K.

Here we report the same robustness results for ViT-L models trained on JFT-300M for 1600 and 800 epochs (Figure 8), and ViT-B models pre-trained for 800 epochs (Figure 9). Figure 9 also compares our ViT-B model to ViT-B models trained from scratch on ImageNet. We find that our model is considerably more robust than training with cross-entropy and Mixup from scratch, and also outperforms PyramidAT (Herrmann et al., 2022), an adversarial training method that introduces significant overheads compared to standard cross-entropy training. We emphasize that here there are two differences in the training: a) the training algorithm itself, and b) the data seen by the model. Our model sees extra JFT-300M data not seen by the other two approaches. This means that the methods are not exactly comparable. It is, however, a realistic setting showing the benefits to robustness of pre-training on large datasets.

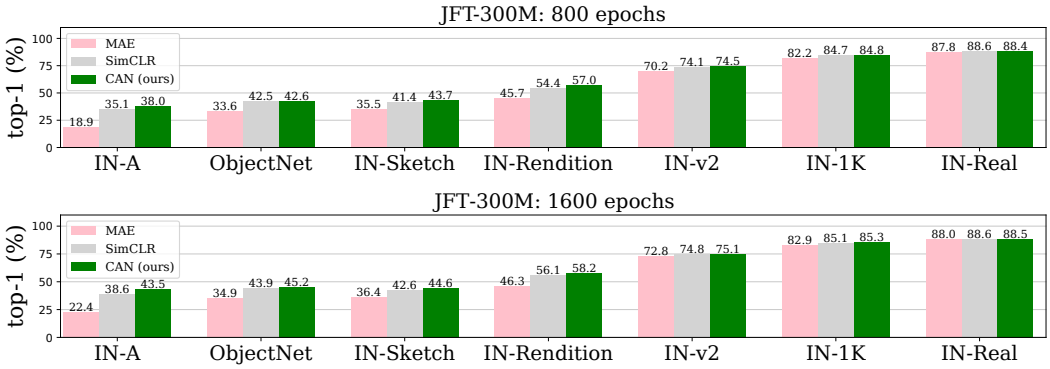

Figure 8: ViT-L models pre-trained on JFT-300M for 800 and 1600 epochs respectively, evaluated on 7 datasets with distribution shifts from IN-1K.

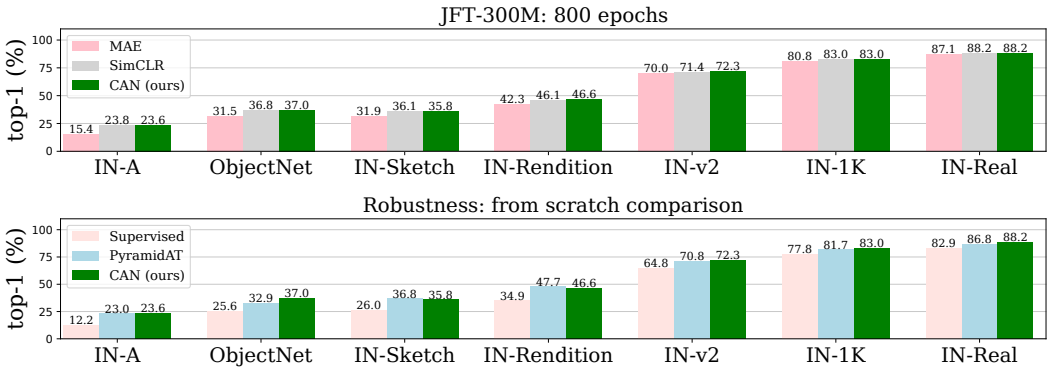

Figure 9: **Top:** ViT-B models pre-trained on JFT-300M for 800 epochs, evaluated on 7 datasets with distribution shifts from IN-1K. **Bottom:** Comparison of our JFT-300M pre-trained ViT-B model to training ViT-B from scratch on IN-1K. We compare to standard supervised cross-entropy training with Mixup, and to PyramidAT (Herrmann et al., 2022), which uses an adversarial training method. CAN considerably outperforms supervised training, and beats PyramidAT in 6 out of 7 cases without requiring adversarial training.

Few shot: Section 4.3 reports 10- and 25-shot results for ViT-L models pre-trained on JFT-300M for 5000 epochs. Here we report 1- and 5-shot results for the same models in Figure 10. We additionally

show the full set of $\{1, 5, 10, 25\}$-shot results for ViT-L models pre-trained on JFT-300M for 800 and 1600 epochs (Figures 11 and 12 respectively), ViT-B models pre-trained on JFT-300M for 800 epochs (Figure 13), and ViT-L models pre-trained on IN-1K for 800 epochs (Figure 14).

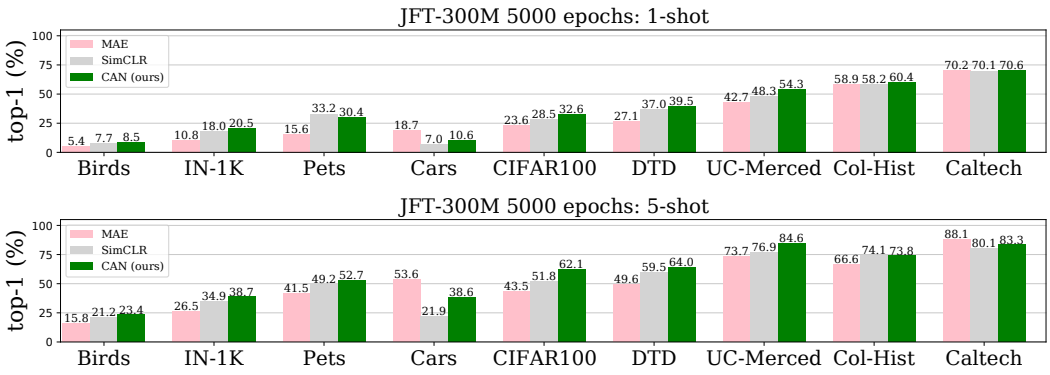

Figure 10: **Few shot:** ViT-L models pre-trained on JFT-300M for 5000 epochs evaluated on 9 few-shot learning tasks. Results accompany the 10- and 25-shot results in Figure 4.

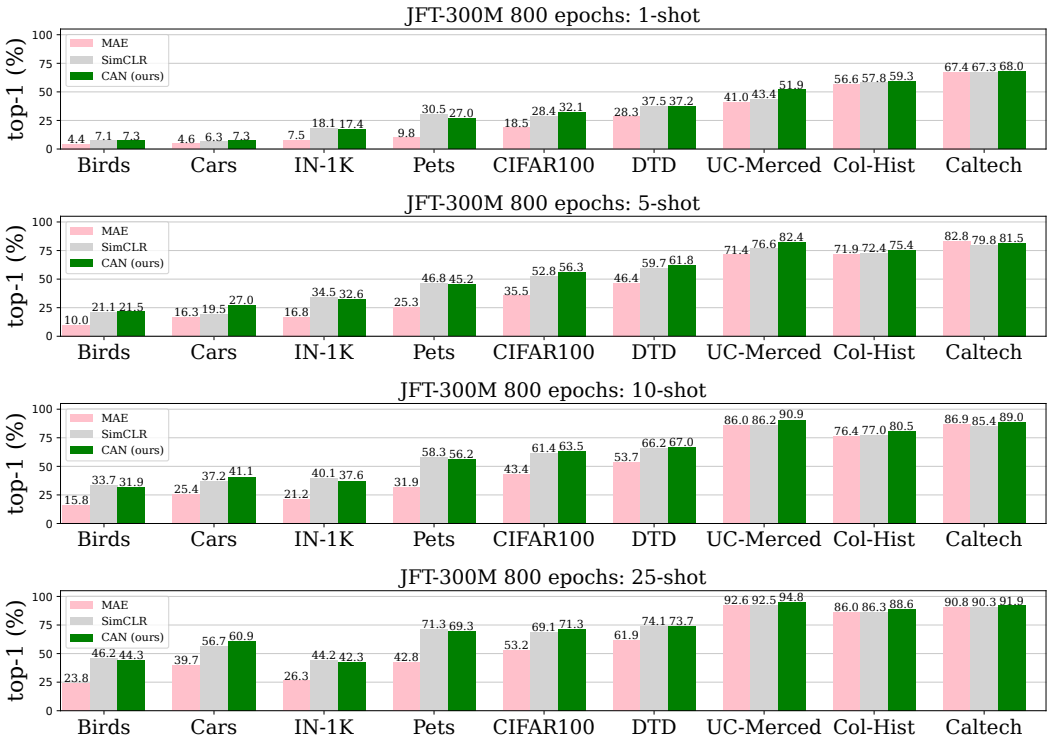

Figure 11: **Few shot:** ViT-L models pre-trained on JFT-300M for 800 epochs are evaluated on 9 few-shot learning tasks.

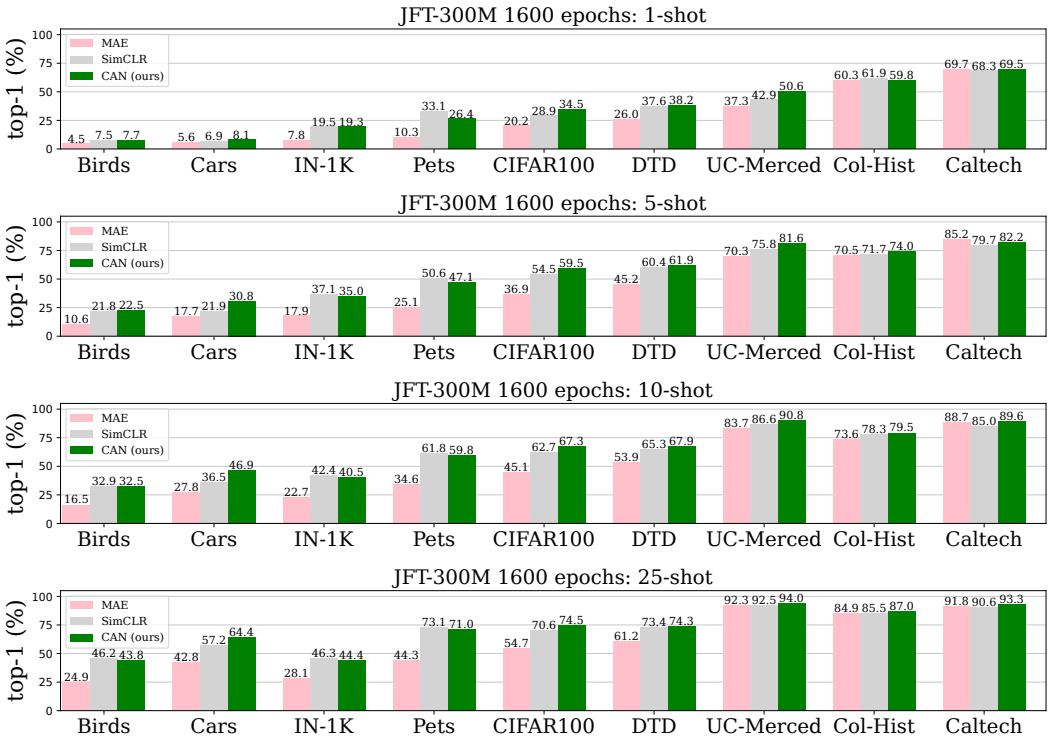

Figure 12: **Few shot:** ViT-L models pre-trained on JFT-300M for 1600 epochs are evaluated on 9 few-shot learning tasks.

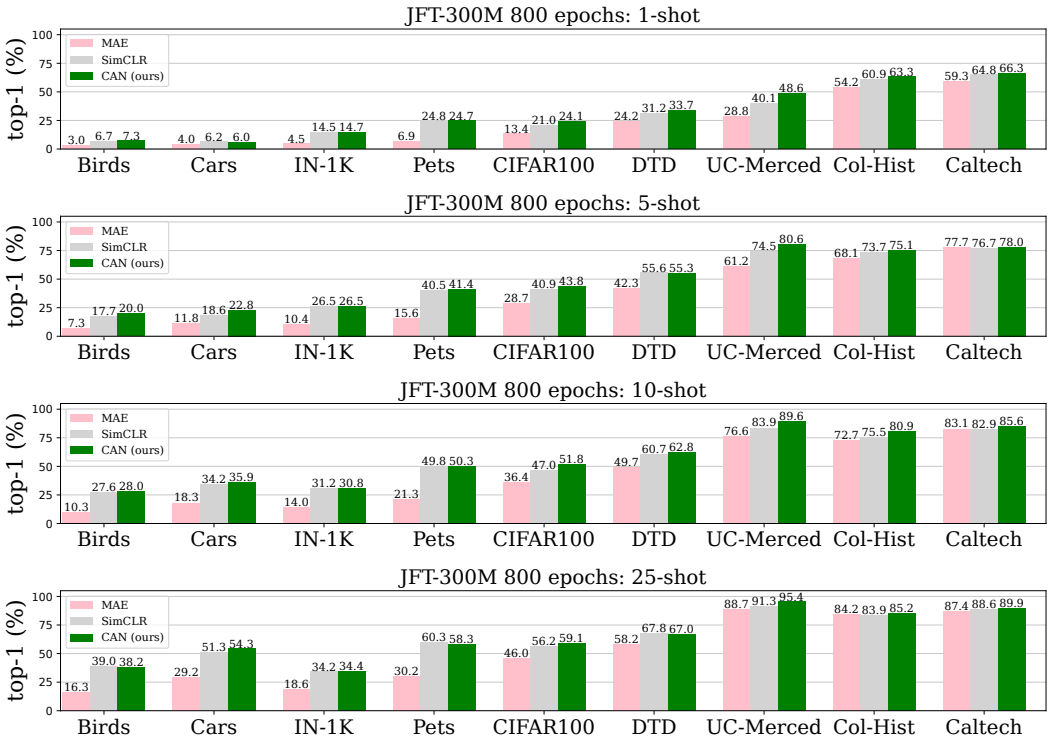

Figure 13: **Few shot:** ViT-B models pre-trained on JFT-300M for 800 epochs are evaluated on 9 few-shot learning tasks.

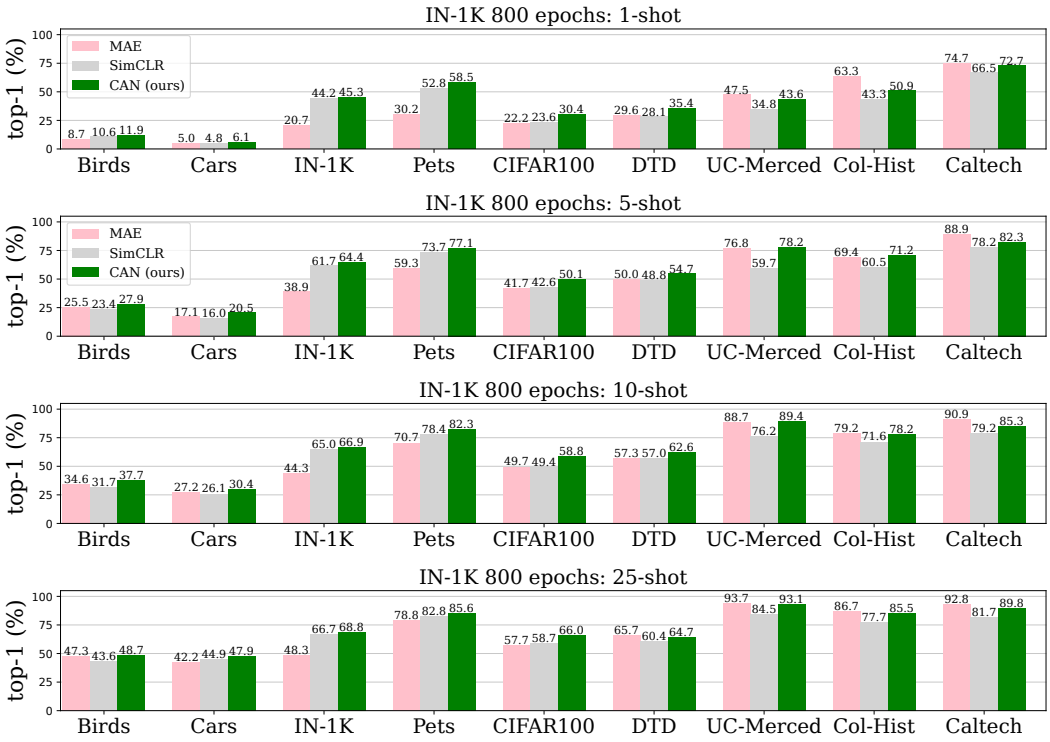

Figure 14: **Few shot:** ViT-L models pre-trained on IN-1K for 800 epochs are evaluated on 9 few-shot learning tasks.

We make a number of observations.

1. Across all settings CAN performs the best on JFT-300M pre-training.

2. The situation is less consistent on IN-1K pre-training. For instance, although MAE has comparatively poor few-shot performance on IN-1K, it is competitive on others for 25-shot evaluation: in this setting CAN only beats MAE on 5 out of 9 datasets. However, on 10-shot CAN outperforms MAE and SimCLR in 8 out of 9 cases, showing a subtle picture.

3. JFT-300M pre-training often outperforms IN-1K pre-training. Comparing Figures 11 and 14, JFT-300M yields better 25-shot CAN performance on 6 out of 9 datasets.

4. Model scale helps. Comparing Figures 13 and 11, ViT-L models perform best in nearly all cases.

B RUNTIME OF CAN COMPARED TO DNC

In the main paper we estimate our method is significantly faster than DnC (Tian et al., 2021). We determined this approximate comparison from the following two pieces of information: 1) DnC reports that 3000 ImageNet epochs takes 29 hours on 512 TPUs for a ResNet-50 model (25M parameters), and 2) 3000 ImageNet epochs of CAN take 78 hours on 64 TPUs for a ViT-L model (300M parameters). We assume a linear relationship between number of TPUs and runtime. Under this assumption, we estimate that CAN would take approximately 10 hours to train with 512 TPUs, compared to the 29 hours reported by Tian et al. (2021) for a model with 1/10th the number of parameters. We emphasize that this is far from an exact comparison and is only intended as a very approximate guide.

C HYPERPARAMETER SETTINGS

We list hyperparameters used for CAN pre-training in Table 5 and Table 6. For preprocessing we closely follow SimCLR Chen et al. (2020b). We use the same hyperparameters for SimCLR pre-training. For MAE pre-training, we use the same hyperparameters as listed in He et al. (2022), except for the use of Glorot uniform initialization instead of LeCun initialization as done in He et al. (2022). We found that this provided better performance for our JAX-based MAE implementation. Table 7 lists the hyperparameters for finetuning evaluations. We use the same set of hyperparameters for each finetuning each pre-training method, and for both ViT-B and ViT-L model sizes. For linear probing we list the hyperparameters in Table 8 for which we followed the settings in He et al. (2022). We use global average pool of the final representation instead of the cls token.

MAE longer training: MAE pre-training for longer training (5000 epochs) on JFT becomes unstable after about 500k steps (training loss oscillates); this results in poorer fine-tuning performance. To overcome this, we decrease the base learning rate by 75% as shown in Table 9. However our model CAN is more stable and we use the same hyperparameters across different numbers of epochs.

Few shot training: For few-shot learning we use the same hyperparameters and pipeline as Dosovitskiy et al. (2021a). We use the same pre-processing as was done in (Kolesnikov et al., 2020). We use a base learning rate of 0.01 and train for 2500 steps, using an input resolution of 384×384.

Hardware details: We use TPU-v4 for all of our experiments. CAN on ViT-B uses 64 TPUs for a batch size of 4096. SimCLR, on the other hand, uses 128 TPUs for the same batch size, and is more compute intensive than CAN.

Decoder architecture: Our decoder architecture is the same as He et al. (2022). We use standard ViT with a decoder depth of 8 and decoder width of 512. We use 16 heads and 2048 as the dimension of the MLP.

Projection head architecture: We use 2 hidden layers in our projection heads. Each layer has a Fully-Connected (FC) layer (dim 4096) followed by BatchNorm (momentum=0.9) followed by ReLU. After these 2 layers we have a FC layer which transforms the features to 128 dimensions. We apply contrastive learning on top of these 128 dimensional features.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2017a)
base learning rate(ViT-B)	2.5e-4
base learning rate (ViT-L)	1.25e-5
weight decay (ViT-B)	0.05
weight decay (ViT-L)	0.1
optimizer momentum	$\beta_1, \beta_2{=}0.9, 0.95$ (Chen et al., 2020a)
batch size	4096
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017b)
warmup epochs (Goyal et al., 2017)	40
augmentation	RandomResizedCrop, Color Jittering(strength=1.0), GrayScale(probability=0.2), Gaussian Blurring (probability=0.5)

Table 5: Hyperparameters for CAN pre-training on ImageNet. Note that we use lower learning rate for ViT-L as compared to ViT-B, following Steiner et al. (2021). We use the same hyper-parameters for SimCLR training.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2017a)
base learning rate (ViT-B)	2.5e-4
base learning rate (ViT-L)	3.125e-6
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2{=}0.9, 0.95$ (Chen et al., 2020a)
batch size	4096
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017b)
warmup epochs (Goyal et al., 2017)	40
augmentation	RandomResizedCrop, Color Jittering(strength=1.0), GrayScale(probability=0.2), Gaussian Blurring (probability=0.5)

Table 6: Hyperparameters for CAN pre-training on JFT. Note that we use lower learning rate for ViT-L as compared to ViT-B, following Steiner et al. (2021). We use the same hyper-parameters for SimCLR pre-training.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2017a)
base learning rate	5e-4
weight decay	0.005
optimizer momentum	$\beta_1, \beta_2{=}0.9, 0.999$ (Chen et al., 2020a)
batch size	1024
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017b)
warmup epochs (Goyal et al., 2017)	5
training epochs	100
label smoothing	0.1
drop path	0.1
layer-wise lr decay	0.65
augmentation	RandomResizedCrop, Flip, RandAug(layers=2, magnitude=9) (Cubuk et al., 2020), Random Erase (Zhong et al., 2020)(probability=0.25)

Table 7: Hyperparameters for finetuning CAN pre-trained model on ImageNet. We use the same hyperparameters for ViT-B and ViT-L, for both JFT and ImageNet pre-trainined models.

config	value
optimizer	LARS (You et al., 2017)
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	16384
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017b)
warmup epochs (Goyal et al., 2017)	10
training epochs	100
batch norm momentum	0.9
label smoothing	0
augmentation	RandomResizedCrop

Table 8: Hyperparameters for linear probing for CAN pre-trained model on ImageNet. We use the same hyperparameters for ViT-B and ViT-L, for both JFT and ImageNet pre-trainined models. Note that these hyperparamters are same as reported in He et al. (2022).

config	value
optimizer	AdamW (Loshchilov & Hutter, 2017a)
base learning rate (ViT-L)	3.75e-6
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2{=}0.9, 0.95$ (Chen et al., 2020a)
batch size	4096
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017b)
warmup epochs (Goyal et al., 2017)	40
augmentation	RandomResizedCrop, Color Jittering(strength=1.0), GrayScale(probability=0.2), Gaussian Blurring (probability=0.5)

Table 9: Hyperparameters for MAE pre-training on JFT for 5000 epochs on ViT-L. The default hyperparameters for MAE are unstable for longer epoch pre-training: we see instability in the loss. To overcome this, we decreased the base learning rate for ViT-L, which helps in stabilizing the loss. Note that our proposed model CAN is more stable and uses the same hyperparameters across different numbers of epochs.444