A Additional Related Work

There are additional works, orthogonal to our contribution, exploring strategies for improving self-supervised learning from images with transformer architectures. For example, in the context of joint-embedding architectures, [36] propose a multi-scale transformer architecture (with a region matching regularization penalty) to reduce the computational complexity of DINO pre-training. In the context of decoder-based architectures, [2] improves the representations obtained through pixel reconstruction by adding a rotation classification loss and a contrastive loss; alternatively [37] directly improves the masked pixel reconstruction loss by constraining the set of pixels that can be masked so as to avoid destroying global semantic structure.

There are also more general approaches for representation learning using multi-modal losses, adversarial training, referential games, latent-space data augmentations, or hand-crafted pretext tasks. For example, [20] learns representations by training a generative network with an adversarial loss; [26] casts view-invariance learning as a contrastive speaker-listener signaling game; [48] performs masked prediction with CovNets using a discriminative loss that requires the network to correctly identify masked patches out of a pool of candidates; [46] learns representations by contrastively enforcing invariance to multi-channel image views (e.g., luminance and chrominance); [62] also learns representations using a contrastive view-invariance criteria, but where views are generated using transformations (interpolations and extrapolations) in feature space; and then there are approaches for representation learning based on solving pretext tasks such as jigsaws [42] or context prediction [19].

B Additional Extreme Low-Shot Comparisons

In additional to the extreme low-shot evaluations in Table 2, here we provide additional comparisons to popular methods for self-supervised learning with Convolutional Networks, using a linear probe and the best publicly available model for each method; cf. Table 10. MSN consistently outperforms the best BYOL/MoCov3/SimCLR models.

Table 10: **Extreme low-shot.** We evaluate the label-efficiency of self-supervised models pretrained on the ImageNet-1K dataset using an extremely small number of the ImageNet-1K labeled images.

		Images per Class		
Method	**Architecture**	1	2	5
SimCLR [13]	RN101	33.5	42.7	52.0
MoCov3 [16]	ViT-B/16	37.4	48.3	58.0
BYOL [25]	RN200 (2×)	40.7	52.7	62.9
MSN (Ours)	**ViT-L/7**	**57.1**	**66.4**	**72.1**

C Implementations Details

In this appendix section we provide the implementation details for MSN pre-training and evaluation.

C.1 MSN Pre-training

We adopt similar hyper-parameter settings that have previously been reported in the self-supervised literature for training Vision Transformers [11, 15]. Specifically, for pre-training, we use the AdamW optimizer [38] with a batch-size of 1024. We linearly warm up the learning-rate from 0.0002 to 0.001 during the first 15 epochs, and decay it following a cosine schedule thereafter. To construct the different image views, we apply the SimCLR data augmentations of [13] to each sampled image; namely random crop, horizontal flip, color distortion, and Gaussian blur. For each sampled image, we generate one large anchor view of size 224×224 pixels, and apply a random mask with a pre-specified masking ratio (0.15 for the ViT-S/16, 0.3 for the ViT-B/16 and ViT-B/8, and 0.7 for the ViT-L/7 and the ViT-B/4). For each sampled image, we also generate 10 small focal anchor views of size 96×96 pixels. We use a temperature of 0.1 for the anchor network, and a temperature of 0.025 for the target network. Following the DINO method of [11], we update the target network via an exponential moving average of the anchor network with a momentum value of 0.996, and linearly increase this value to 1.0 by the end of training. Similarly, following [11], weight decay is set to 0.04 and increased to 0.4 throughout training via a cosine schedule. By default, we set the ME-MAX regularization weight λ to 1.0 and apply Sinkhorn normalization to the targets [10] to avoid having to tune the ME-MAX regularization weight; however, in general, we observe stronger MSN performance when omitting Sinkhorn normalization (see Appendix E). We train with a 3-layer projection head with output dimension 256 and batch-normalization at the input and hidden layers, and use 1024 prototypes of dimension 256. We observe that using more prototypes has little effect on training, but using too few prototypes can hurt performance (see Appendix E). We discard the projection head during evaluation, and always use the representations computed from the output of the target encoder trunk for evaluation.

C.2 Low-Shot Evaluation

To avoid overfitting, we freeze the weights of the pre-trained model and train a linear classifier on top using 1, 2 or 5 labeled samples per class. Specifically, we take a single center crop of each labeled image, extract its representation using the pre-trained model, and then train a classifier on these representations using L_2-regularized logistic regression. Following [11], we use the `cyanure` package [40] to run logistic regression on the extracted representations. This objective is smooth and strongly-convex (i.e., has a unique minimizer) and can therefore be efficiently solved for using the `cyanure` python numerical solver on a single CPU core. All low-shot evaluations (including the 1% ImageNet-1K evaluation) are

computed with this procedure, except for models pre-trained using MAE [27], which benefit from using partial fine-tuning [27].

Partial fine-tuning corresponds to fine-tuning the last block of the pre-trained model along with a linear head. MAE benefits from partial fine-tuning, but for sufficiently large models, such as the ViT-H/14, this leads to significant overfitting in the low-shot regime. Our results in Table 2 and Figure 2 report the best performance across evaluation methods for MAE. In particular, all the MAE results are obtained via partial fine-tuning, except for the 1 image per class setting, and all results with the ViT-H/14 architecture, which use a linear head. We compare both protocols in more detail in Appendix E.

C.3 Linear Evaluation

For linear evaluation, we use a similar procedure as [27]. Specifically, we use a large batch-size of 16,384 images and train a linear classifier for 100 epochs using a learning rate of 6.4, and decay it following a cosine schedule. We only apply basic data augmentations; namely, random resized crops to a resolution of 224×224 pixels, and random horizontal flips. We also L_2-normalize the representations before feeding them into the linear classifier, and optimize the classifier weights using SGD with Nesterov momentum. We do not apply any weight-decay and do not use any warmup.

C.4 Fine-Tuning Evaluation

We follow the common practice for fine-tuning SSL pre-trained ViT models. Specifically, we follow the setup of [47, 5, 27]. We fine-tune a pre-trained ViT model for 100 epochs on the full supervised ImageNet-1K training data set using the AdamW [38] optimizer. We use a batch size of 1024 with a learning rate of 0.002. The learning rate is linearly warmed-up during the first 5 epochs and decayed with a cosine schedule thereafter. A layer-wise decay of 0.65 is also applied, along with the data augmentations defined by RandAugment(9, 0.5) [17]. We additionally use label smoothing set to 0.1, mixup [58] set to 0.8, cutmix [56] set to 1.0, and drop path set to 0.2.

C.5 Transfer Learning

Linear Evaluation When performing linear evaluation for transfer learning, we freeze the weights of the ImageNet-1K pre-trained model and optimize a linear classifier on top. We resize each downstream image to 256×256 pixels, and take a single center crop of size 224×224 pixels. Next, we extract a representation of each image using the pre-trained model, and subsequently train a classifier on top using L_2-regularized logistic regression.

Fine Tuning When performing end-to-end fine-tuning for transfer learning, we follow the protocol of DeiT and DINO [47, 11]. Models transferred to CIFAR10 and CIFAR100 are fine-tuned for 1000 epochs using a batch size of 768 and a learning rate of 0.000075. Models transferred to iNat18 and iNat19 models are fine-tuned for 300 epochs using a batch size of 1024 and a learning of 0.0001. All transfer fine-tuning experiments use the data augmentations defined by RandAugment(9, 0.5) [17]. We also use label smoothing set to 0.1, mixup [58] set to 0.8, cutmix [56] set to 1.0, and drop path set to 0.1. The learning rate is linearly warmed-up during the 5 first epochs and decayed with a cosine schedule thereafter.

D Theoretical Guarantees

In this section we describe how MSN pre-training provably avoids representation collapse.

Recall that in each iteration of pre-training, we sample a mini-batch of $B \geq 1$ images, and generate $M \geq 1$ anchor views of each image. Here we show that MSN is guaranteed to avoid the trivial collapse of representations under the following assumption.

Assumption 1 (Target Sharpening) *The target p^+ is sharpened, such that it is not equal to the uniform distribution.*

Proposition 1 (Non-Collapsing Representations). *Suppose Assumption 1 holds. If $f_\theta(\cdot)$ is such that the representations collapse, i.e., $z_{i,m} = z_{j,k}$ for all $i, j \in [B]$ and $m, k \in [M]$, then $\left\|\nabla_\theta H(p_i^+, p_{i,m})\right\| + \|\nabla_\theta H(\overline{p})\| > 0$ for all i, m.*

Proof. For L_2-normalized representations and prototypes, the prediction $p_{i,m} \in \Delta_K$ corresponding to the m^{th} view of the i^{th} image in the mini-batch is given by

$$p_{i,m} \coloneqq \text{softmax}\left(\frac{z_{i,m} \cdot \mathbf{q}}{\tau}\right),$$

where $\mathbf{q} \in \mathbb{R}^{K \times d}$ is the prototype matrix with $K > 1$ learnable prototypes, each of dimension d, and $\tau > 0$ is a scaler temperature. Since $z_{i,m} = z_{j,k}$ for all $i, j \in [B]$ and $m, k \in [M]$, it holds that $z_{i,m} \cdot \mathbf{q} = z_{j,k} \cdot \mathbf{q}$, and therefore $p_{i,m} = p_{j,k}$. Now consider two separate cases.

Case 1: The predictions are equal to the uniform distribution, i.e., $p_{i,m} = \frac{1}{K}\mathbf{1}_K$, where $\mathbf{1_K} \in \mathbb{R}^K$ is the K-dimensional vector with each entry equal to 1. In that case, since, by Assumption 1, the targets p_i^+ are sharpened such that they are not equal to the uniform distribution, it follows that $p_{i,m} \neq p_i^+$, and hence $\left\|\nabla_\theta H(p_i^+, p_{i,m})\right\| > 0$.

Case 2: The predictions are not equal to the uniform distribution, i.e., $p_{i,m} \neq \frac{1}{K}\mathbf{1}_K$. In that case, we have that the average prediction across all the anchor views $\overline{p} \coloneqq \frac{1}{MB}\sum_{i=1}^{B}\sum_{m=1}^{M} p_{i,m}$ is also not equal to the uniform distribution; i.e., $\overline{p} \neq \frac{1}{K}\mathbf{1}_K$, and hence $\|\nabla_\theta H(\overline{p})\| > 0$.

Proposition 1 provides a theoretical guarantee that MSN is immune to the trivial collapse of representations. In short, the underlying principle is that entropy maximization encourages the anchor predictions to utilize the full set of prototypes, thereby preventing collapse to a non-uniform distribution, while target sharpening encourages the anchor predictions to be confident, thereby preventing collapse to the uniform distribution.

Note that the sharpening mechanism defined in Section 3 (i.e., applying a temperature τ^+ in the target network softmax) may not always satisfy Assumption 1, unless one introduces a simple tie-breaking rule. In practice, such a rule is not necessary as the targets never become uniform (since we apply sharpening from the start of the training), although, it is important to use a sufficiently small temperature value in this case.

E Additional Ablations

E.1 Sinkhorn Normalization

By default, we set the ME-MAX regularization weight λ to 1.0 and apply Sinkhorn normalization on the targets to avoid having to tune the ME-MAX regularization weight. However, we find that tuning the ME-MAX regularization weight and omitting Sinkhorn normalization can result in better performance; cf. Table 11.

Table 11: **Effect of Sinkhorn normalization.** We train a ViT-S/16 with a masking ratio of 0.15, and explore the impact of Sinkhorn normalization during pre-training on low-shot performance with 1% of ImageNet-1K. Tuning the ME-MAX regularization weight and omitting Sinkhorn normalization gives better performance.

Architecture	**Target Normalization**	ME-MAX **weight** λ	**Top 1**
ViT-S/16	Sinkhorn	1.0	66.4
	None	1.0	60.8
	None	5.0	**67.2**

E.2 Number of Prototypes

By default we train with 1024 prototypes of dimension 256. In this section we explore the effect of the number of prototypes on low-shot performance. We observe that using more prototypes has little effect on training, but using too few prototypes can hurt performance; cf. Table 12.

E.3 Masked Auto-Encoder Partial Fine-Tuning

Here we explore the low-shot performance of MAE when relying on alternative evaluation strategies. [27] conjecture that using pixel reconstruction in their MAE

Table 12: **Effect of number of prototypes.** We train a ViT-B/16 with a masking ratio of 0.3, and explore the impact of the number of prototypes during pre-training on low-shot performance with 1% of ImageNet-1K. Using more prototypes has little effect on training, but using fewer prototypes can degrade performance.

Architecture	Prototypes	Top 1
ViT-B/16	512	67.6
	1024	**69.5**
	2048	**69.5**

objective results in encoder representations of a lower semantic level than other methods, which may explain their difficulty in training a linear classifier on the frozen features. In Table 13 we explore the effect of partial fine-tuning on the low-shot performance of pre-trained MAE models. Partial fine-tuning corresponds to fine-tuning the last block of the pre-trained model along with a linear head on the available labeled samples. As observed in [27], MAE benefits from partial fine-tuning. However, for sufficiently large models, such as the ViT-H/14, this leads to significant overfitting in the low-shot regime, where one must instead resort to linear evaluation. We report the best numbers for MAE across the two low-shot adaptation strategies in Figure 2.

Table 13: **MAE low-shot evaluations.** Top-1 low-shot validation accuracy for different training strategies with MAE pre-trained models. Partial fine-tuning corresponds to fine-tuning the last block of the pre-trained model along with a linear head on the available labeled samples. Linear evaluation corresponds to training a linear classifier on top of the frozen pre-trained encoder. MAE benefits from partial fine-tuning, but for sufficiently large models, such as the ViT-H/14, this leads to significant overfitting in the low-shot regime, where one must instead one must resort to linear evaluation.

		Top 1		
		Images per Class		
Architecture	Adaptation Strategy	2	5	∼13
ViT-B/16	Partial Fine-Tuning	**25.0**	**40.5**	**51.1**
	Linear Eval.	14.5	25.2	36.6
ViT-L/16	Partial Fine-Tuning	19.3	**42.3**	**59.4**
	Linear Eval.	**22.1**	35.7	48.6
ViT-H/14	Partial Fine-Tuning	`rand`	`rand`	`rand`
	Linear Eval.	**18.6**	**32.8**	**46.7**

F MSN Representation Robustness

Next we report the performance of MSN-pre-trained models on datasets that have been developed to evaluate the robustness of models trained on the standard ImageNet training set. We consider four datasets: ImageNet-A ([32])[2], ImageNet-R ([30])[3], ImageNet-Sketch ([51])[4], and ImageNet-C ([31])[5].

Table 14 shows results for a ViT-B/16 pre-trained using MSN and fine-tuned using the protocol described in Appendix C. For comparison, we also report the performance of a fine-tuned ViT-B/16 pre-trained using MAE [27], along with a supervised ResNet50 baseline, which is available in the PyTorch Torchvision package[6]. For ImageNet-A, -R, and -Sketch, we report top-1 accuracy on each provided validation set. For ImageNet-C, we use the mean Corruption Error metric proposed in [31], where values are normalized by AlexNet performance on the same validation set.

Table 14: **Evaluation on alternative ImageNet validation sets.** We evaluate the performance of a fine-tuned ViT-B/16 model on four alternative ImageNet validation sets: ImageNet-A, ImageNet-R, ImageNet-Sketch, and ImageNet-C. The metric used for the first three (-A, -R, and -Sketch) is top-1 accuracy on the validation set. On ImageNet-C, performance is measured in terms of mean Corruption Error (mCE) [31].

	IN-A (top-1 ↑)	**IN-R** (top-1 ↑)	**IN-Sketch** (top-1 ↑)	**IN-C** (mCE ↓)
Supervised ResNet50	0.04	36.11	24.2	76.7
MAE ViT-B/16 [27]	35.9	48.3	34.5	51.7
MSN ViT-B/16	**37.5**	**50.0**	**36.3**	**46.6**

In each case we find that the performance of an MSN-pretrained ViT-B/16 is comparable or better than that of an MAE-pretrained ViT-B/16. Note also, that larger MAE-pretrained models achieve stronger performance on all four datasets [27].

G MSN Invariance to Masking

The goal of MSN pretraining is to denoise the input images at the representation level by ensuring that the representation of a masked input matches the representation of the unmasked one. Here, we shows that MSN pretraining learns representations that are robust to patch masking.

[2] https://github.com/hendrycks/natural-adv-examples
[3] https://github.com/hendrycks/imagenet-r
[4] https://github.com/HaohanWang/ImageNet-Sketch
[5] https://github.com/hendrycks/robustness
[6] https://github.com/pytorch/vision

In Table 15, we evaluate the performance of MSN and DINO when masking parts of an image during evaluation. Models are evaluated on 1% of ImageNet-1K using logistic regression on top of frozen features. The logistic regression classifier is trained using masked images, and then evaluated on the standard ImageNet-1K validation set using unmasked images.

If the MSN representations are robust to missing image patches, then a linear classifier should be able to identify generalizable features when training on the representations of masked images. On the other hand, if the representations output by the learned encoder are not robust to missing image patches, then a linear classifier would have difficulty finding generalizable features when training on the representations of masked images.

We observe that masked pre-training results in representations that are more robust to patch removal, suggesting that MSN is performing an image denoising at the representation level. Furthermore, models pre-trained with more aggressive masking exhibit this quality to a higher degree. For example, the low-shot accuracy of ViT-L/7 pre-trained with aggressive masking is almost unaffected when we remove 70% of the patches at test time; 75.1% top-1 without dropping patches during evaluation versus 74.9% top-1 when dropping 70% of the patches during evaluation.

Table 15: **Robustness to missing patches (low-shot).** Evaluating the low-shot accuracy of pre-trained models on 1% of ImageNet-1K when corrupting the annotated images by dropping patches. We train a linear classifier using masked images, and then evaluate on the standard ImageNet-1K validation set using unmasked images. We observe that MSN pre-training leads to representations that are more robust to masking. Moreover, models pre-trained with more aggressive masking exhibit this behaviour to a higher degree.

			Top 1		
			Eval. Masking Ratio		
Alg.	**Arch.**	**Pre-train Masking Ratio**	0.0	0.7	Δ
DINO	ViT-B/16	0.0	67.0	63.1	-3.9
MSN	ViT-B/16	0.3	69.5	67.1	-2.4
	ViT-L/7	0.7	75.1	74.9	**-0.2**

We also report the average cosine distance between masked and unmasked representations of the same image in Table 16. As expected, the cosine similarity between masked and unmasked representations of the same image is higher when pre-training with MSN, supporting the observation that masked-pretraining results in representations that are more robust to patch-removal.

Table 16: **Robustness to missing patches (cosine-similarity).** Average Cosine Distance between masked and unmasked representations of the same image. We compare the representations learned with MSN masked pre-training to those learned with DINO when using a ViT-B/16 encoder. The MSN ViT-B/16 is pre-trained with a masking ratio of 0.3. The cosine distances are computed and averaged over the ImageNet-1k validation set. The cosine similarity between masked and unmasked representations of the same image is higher when pre-training with MSN, supporting the observation that masked-pretraining results in representations that are more robust to patch-removal.

	Cosine Similarity				
	Eval. Masking Ratio				
Alg.	0.15	0.3	0.5	0.7	0.9
DINO	0.98	0.97	0.92	0.81	0.56
MSN	**0.99**	**0.99**	**0.99**	**0.98**	**0.97**

H Qualitative Analysis

We qualitatively investigate the properties of the MSN pre-trained representations. We follow the RCDM framework [8] and train a conditional generative diffusion model, which maps a learned image representation back to pixel space. Specifically, RCDM takes as input random noise and the representation vector of an image computed by an SSL model (either an MSN pre-trained model or a DINO pre-trained model in this analysis), and aims to reconstruct the image as close as possible to the original one through a diffusion process.

By using RCDM to sample an image based on its SSL representation, we can visualize how different pre-training strategies affect the degree of information contained in the representation. Qualities that vary across RCDM samples represent information that is not contained in the pre-trained representation. Qualities that are semantically common across samples represent information contained in the representation.

H.1 Comparison with DINO

We apply RCDM on top of either a DINO or MSN pre-trained ViT-B/8 encoder to generate images of resolution 128×128 pixels. RCDM is trained using unmasked images processed with the ViT-B/8 encoder. We then use masked images from the validation set at sampling time.

In Figure 4, we generate samples for RCDM when masking 50% of the conditioning images. The first column depicts images from the ImageNet validation set. The second column depicts the same image, but with 50% of the patches masked. The representation of the masked image is used as conditioning for the RCDM diffusion model. The subsequent columns in Figure 4 show various images sampled from the conditioned RCDM diffusion model. We observe that the RCDM samples conditioned on the MSN representations (cf. Figure 4a) preserve the

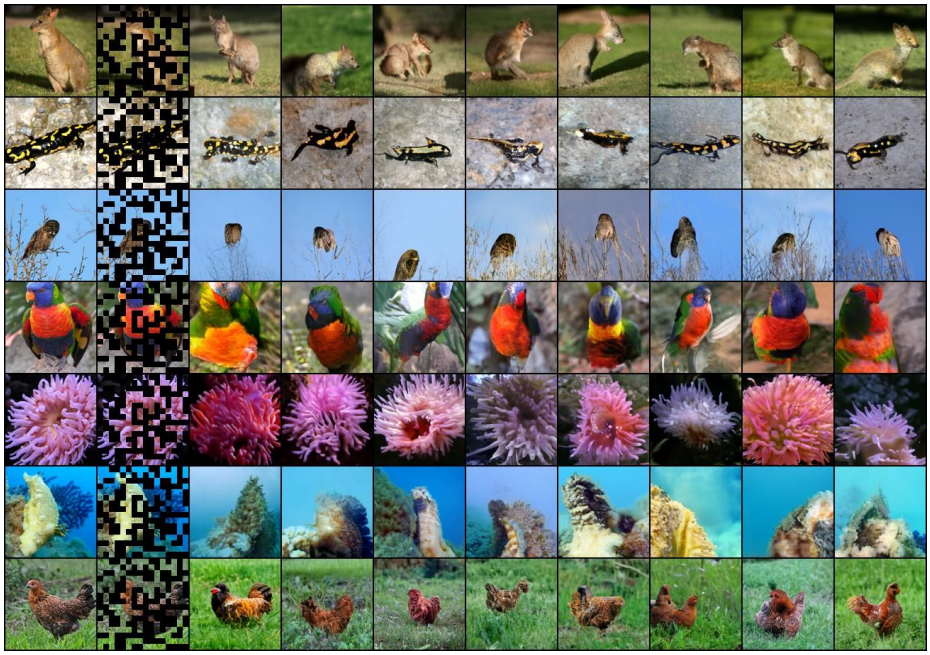
semantic category of the masked images, and remain visually close to the original image, despite the missing patches. By contrast, the samples generated by the RCDM diffusion model conditioned on the DINO representations (cf. Figure 4b) are more blurry and do not preserve as well the semantic category of the masked images.

Figure 5 depicts similar visualizations, but with 80% of the patches masked. In this case, even with 80% of the patches missing, samples generated by RCDM conditioned on MSN representations preserve some of the structure in original images (cf. Figure 5a). On the other hand, conditioning on DINO representations leads to almost uniform background generation (cf. Figure 5b).

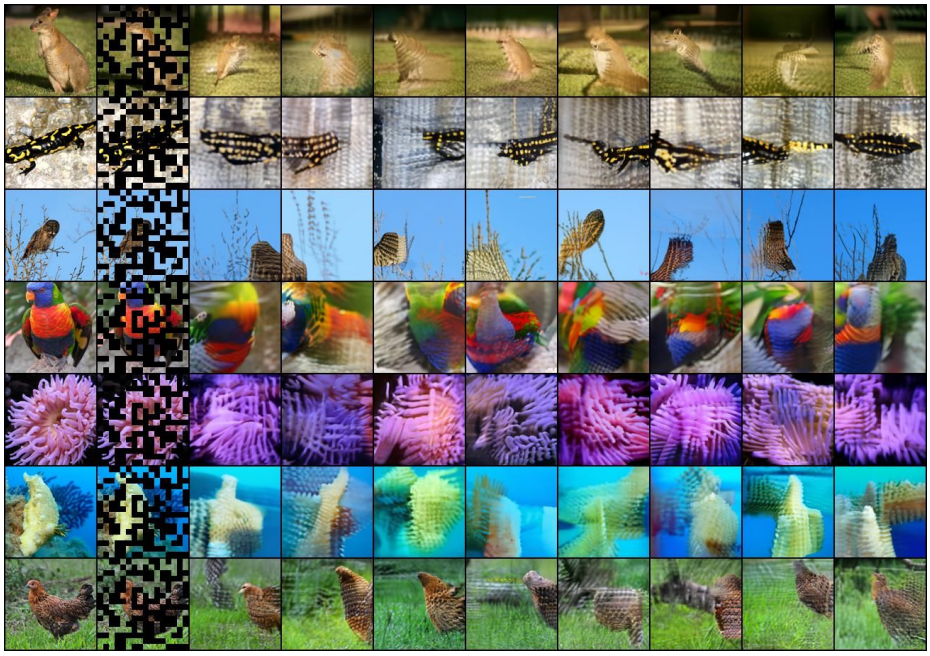
H.2 MSN ViT-L/7 Visualizations

We apply RCDM on top of the MSN pre-trained ViT-L/7 encoder to generate images with a resolution of 256×256 pixels. RCDM is trained using images with 70% of patches masked. We then use masked images from the validation set (with various masking ratios) at sampling time, see Figures 6, 7, and 8.

Visualizations show that MSN discards instance-specific information such as background, pose, and lighting, while retaining semantic information about the images, even when a large fraction of the patches are masked.

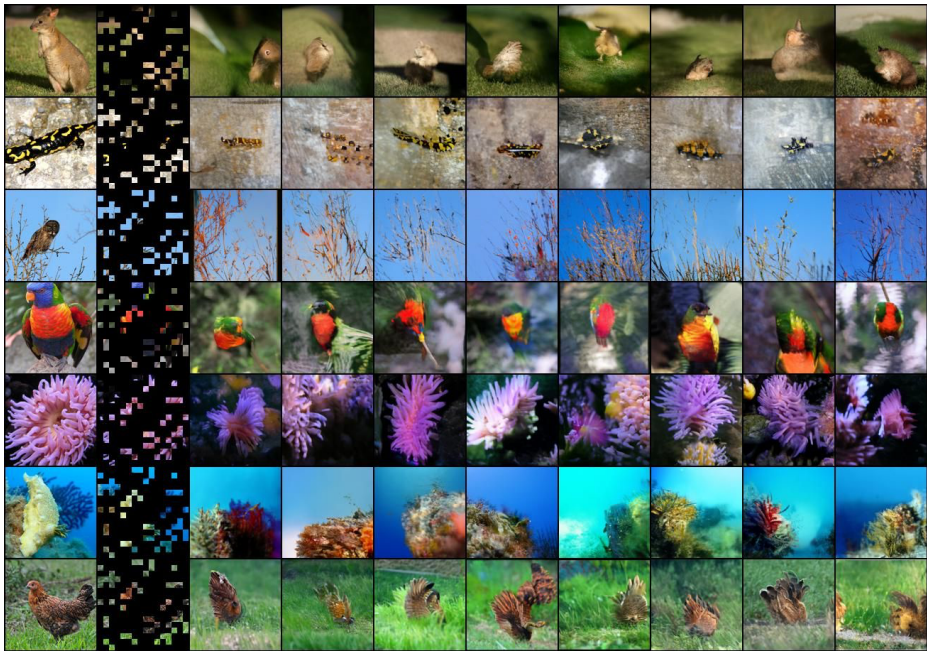


(a) MSN Representations visualized on ImageNet validation set.

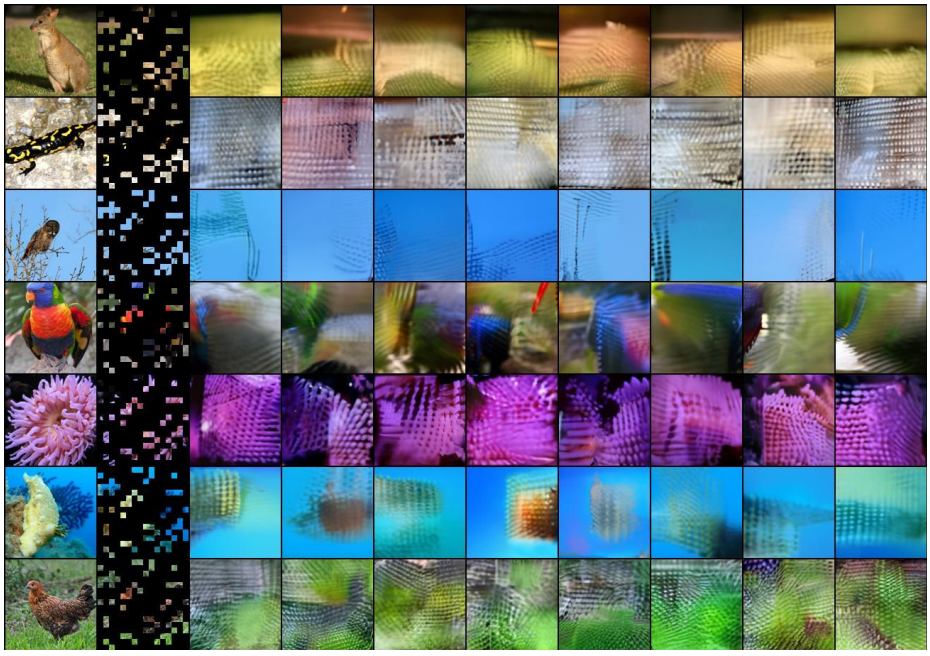


(b) DINO Representations visualized on ImageNet validation set.

Fig. 4: **Visualizations of ViT-B/8 pre-trained representations computed from images with 50% of patches masked.** First column: original image. Second column: image with 50% of patches masked used to compute representations of an SSL pre-trained ViT-B/8 encoder. Other columns: RCDM sampling from generative model conditioned on SSL representation of masked image.



(a) MSN representations visualized on ImageNet validation set.



(b) DINO representations visualized on ImageNet validation set.

Fig. 5: **Visualizations of ViT-B/8 pre-trained representations computed from images with 80% of patches masked.** First column: original image. Second column: image with 80% of patches masked used to compute representations of an SSL pre-trained ViT-B/8 encoder. Other columns: RCDM sampling from generative model conditioned on SSL representation of masked image.

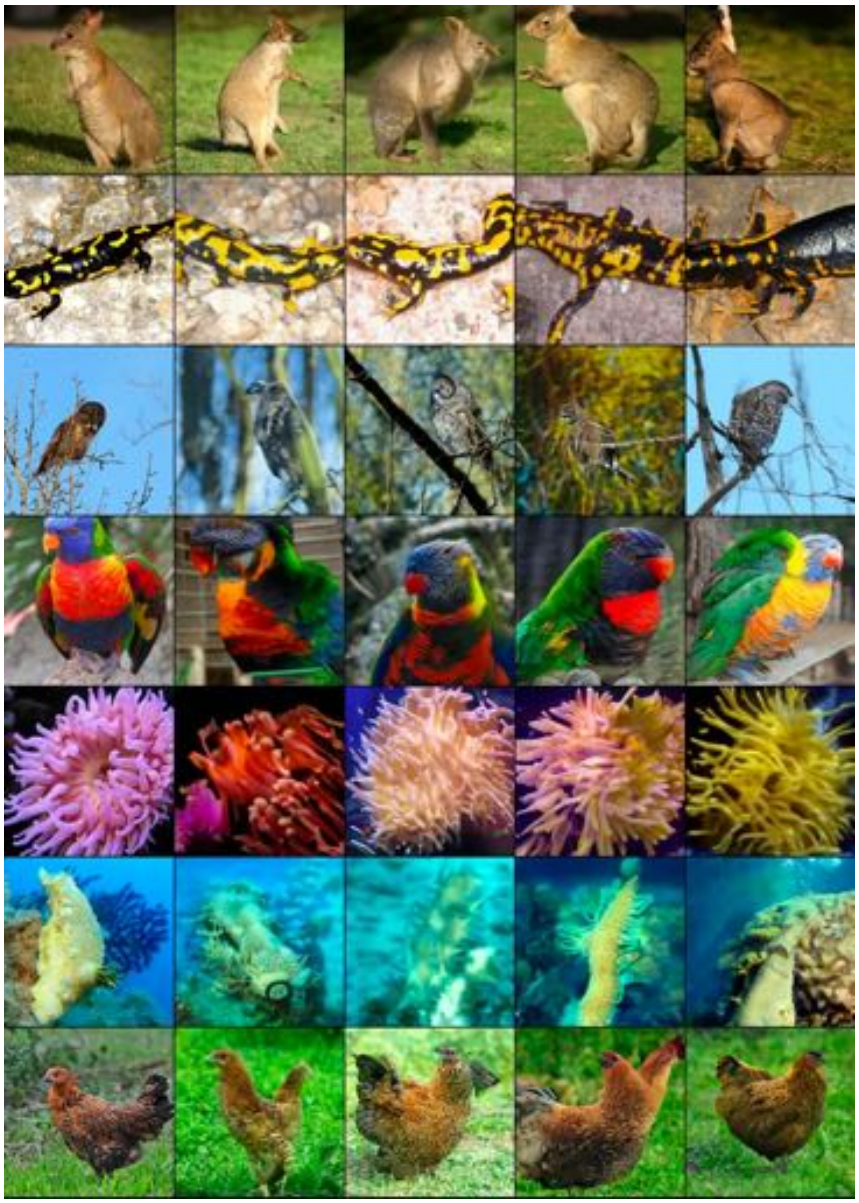


Fig. 6: **Visualizations of MSN pre-trained ViT-L/7 representations computed from unmasked images.** First column: original image. Other columns: RCDM sampling from generative model conditioned on MSN representation using a ViT-L/7 encoder. MSN representations are computed from unmasked images. Qualities that vary across samples represent information that the representation is invariant to; e.g., in this case, MSN discards background, pose, and lighting information. Qualities that are common across samples represent information contained in the pre-trained representation.

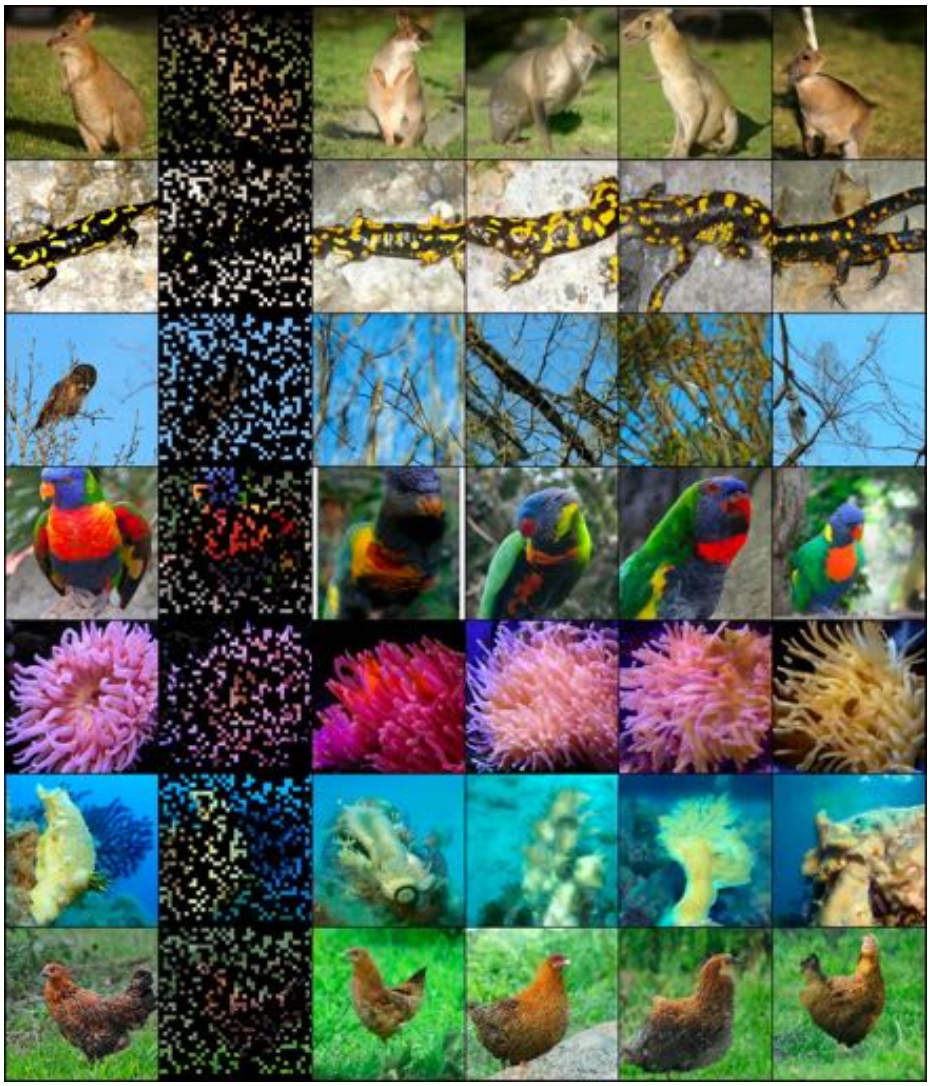


Fig. 7: **Visualizations of MSN pre-trained ViT-L/7 representations computed from images with 70% of patches masked.** First column: original image. Second column: image with 70% of patches masked used to compute representations of an SSL pre-trained ViT-L/7 encoder. Other columns: RCDM sampling from generative model conditioned on SSL representation of masked image. Qualities that vary across samples represent information that the representation is invariant to; e.g., in this case, MSN discards background, pose, and lighting information. Qualities that are common across samples represent information contained in the pre-trained representation.

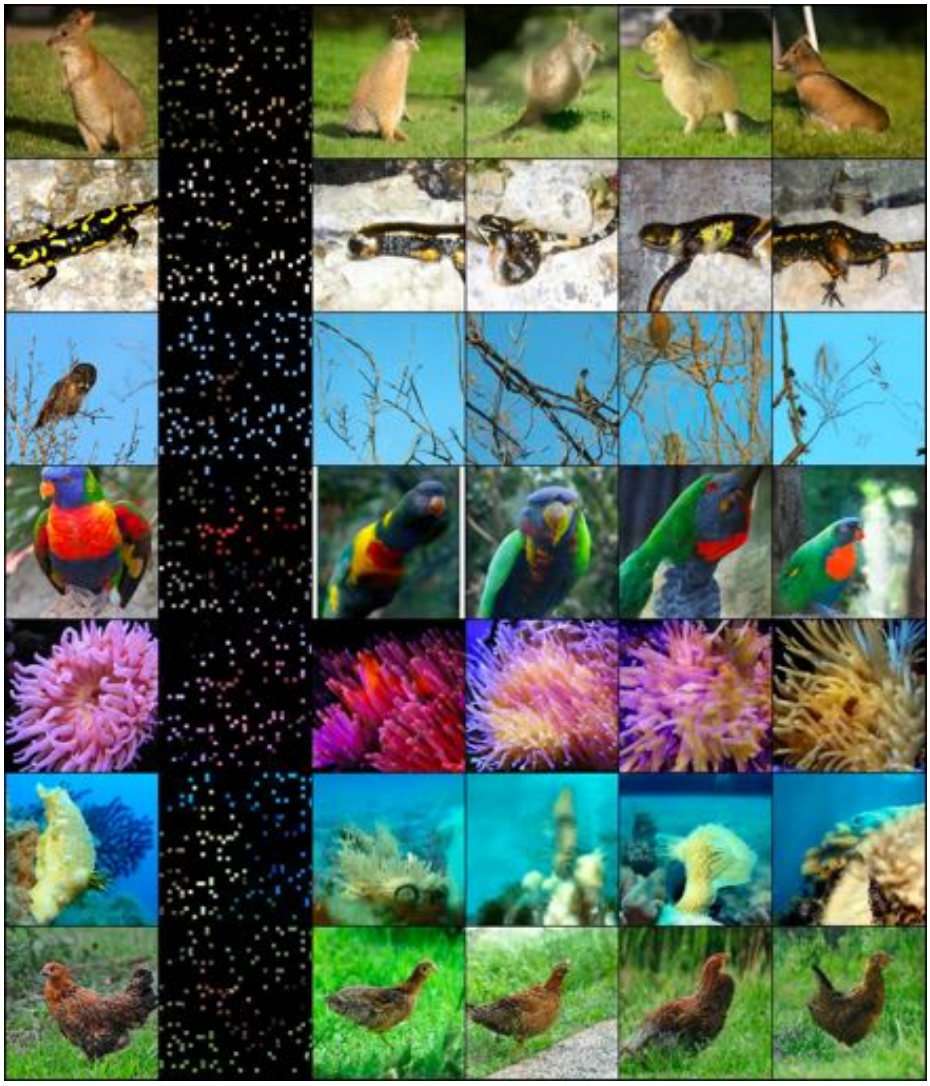


Fig. 8: **Visualizations of MSN pre-trained ViT-L/7 representations computed from images with 90% of patches masked.** First column: original image. Second column: image with 90% of patches masked used to compute representations of an SSL pre-trained ViT-L/7 encoder. Other columns: RCDM sampling from generative model conditioned on SSL representation of masked image. Qualities that vary across samples represent information that the representation is invariant to; e.g., in this case, MSN discards background, pose, and lighting information. Qualities that are common across samples represent information contained in the pre-trained representation. Even with high-masking ratio, MSN retains semantic information about the images.