MAGE: MAsked Generative Encoder to Unify Representation Learning and Image Synthesis

Tianhong Li[1*] Huiwen Chang[3] Shlok Kumar Mishra[2] Han Zhang[3]
Dina Katabi[1] Dilip Krishnan[3]
[1]MIT CSAIL [2]University of Maryland [3]Google Research

Abstract

*Generative modeling and representation learning are two key tasks in computer vision. However, these models are typically trained independently, which ignores the potential for each task to help the other, and leads to training and model maintenance overheads. In this work, we propose MAsked Generative Encoder (MAGE), the first framework to unify SOTA image generation and self-supervised representation learning. Our key insight is that using variable masking ratios in masked image modeling pre-training can allow generative training (very high masking ratio) and representation learning (lower masking ratio) under the same training framework. Inspired by previous generative models, MAGE uses semantic tokens learned by a vector-quantized GAN at inputs and outputs, combining this with masking. We can further improve the representation by adding a contrastive loss to the encoder output. We extensively evaluate the generation and representation learning capabilities of MAGE. On ImageNet-1K, a single MAGE ViT-L model obtains **9.10** FID in the task of class-unconditional image generation and **78.9%** top-1 accuracy for linear probing, achieving state-of-the-art performance in both image generation and representation learning. Code is available at https://github.com/LTH14/mage.*

1. Introduction

In recent years, we have seen rapid progress in both generative models and representation learning of visual data. Generative models have demonstrated increasingly spectacular performance in generating realistic images [3,7,16,50], while state-of-the-art self-supervised representation learning methods can extract representations at a high semantic level to achieve excellent performance on a number of downstream tasks such as linear probing and few-shot transfer [2,6,8,13,24,26].

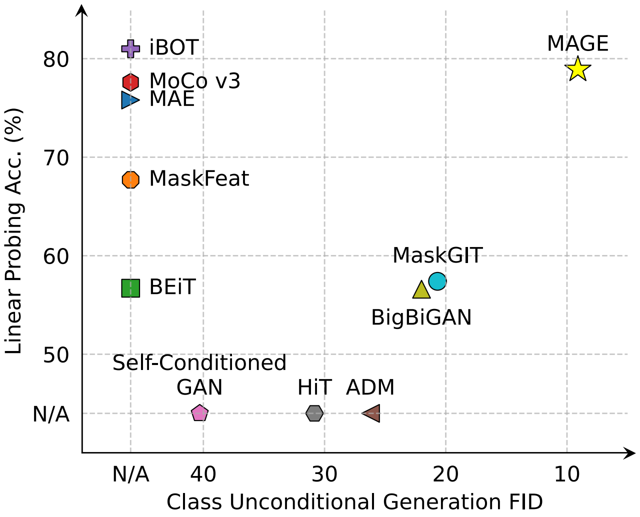


Figure 1. Linear probing and class unconditional generation performance of different methods trained and evaluated on ImageNet-1K. MAGE achieves SOTA performance in linear probing and establishes a new SOTA in class unconditional generation.

Currently, these two families of models are typically trained independently. Intuitively, since generation and recognition tasks require both visual and semantic understanding of data, they should be complementary when combined in a single framework. Generation benefits representation by ensuring that both high-level semantics and low-level visual details are captured; conversely, representation benefits generation by providing rich semantic guidance. Researchers in natural language processing have observed this synergy: frameworks such as BERT [15] have both high-quality text generation and feature extraction. Another example is DALLE-2 [47], where latents conditioned on a *pre-trained* CLIP representation are used to create high-quality text-to-image generations.

However, in computer vision, there are currently no widely adopted models that unify image generation and representation learning in the same framework. Such unification is non-trivial due to the structural difference be-

[*] This work was done when Tianhong Li was intern at Google Research. Correspondence to: Tianhong Li <tianhong@mit.edu>, Huiwen Chang <huiwenchang@google.com>.

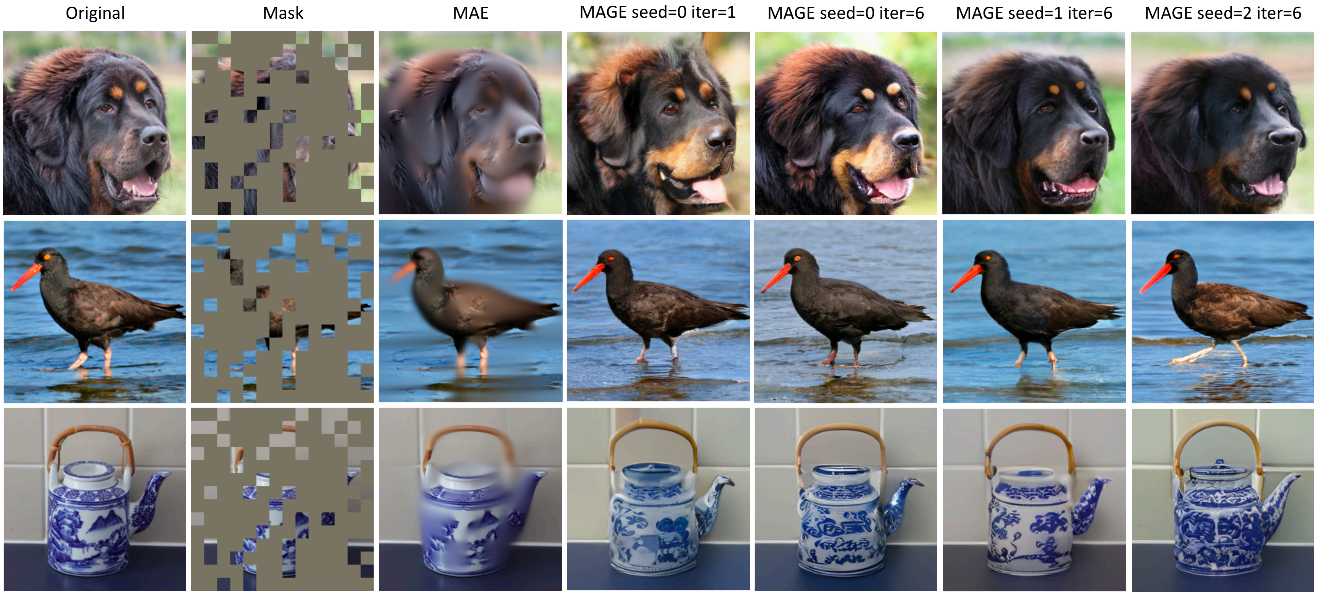


Figure 2. Reconstruction results using MAE and MAGE with 75% masking ratio. MAE reconstructs blurry images with low quality, while MAGE can reconstruct high-quality images with detail, and further improves quality through iterative decoding (see subsection 3.2 for details). With the same mask, MAGE generates diverse reconstruction results with different random seeds. Note that the mask for MAGE is on semantic tokens whereas that of MAE is on patches in the input image.

tween these tasks: in generative modeling, we **output** high-dimensional data, conditioned on low-dimension inputs such as class labels, text embeddings, or random noise. In representation learning, we **input** a high-dimensional image and create a low-dimensional compact embedding useful for downstream tasks.

Recently, a number of papers have shown that representation learning frameworks based on masked image modeling (MIM) can obtain high-quality representations [2, 18, 26, 32], often with very high masking ratios (e.g. 75%) [26]. Inspired by NLP, these methods mask some patches at the input, and the pre-training task is to reconstruct the original image by predicting these masked patches. After pre-training, task-specific heads can be added to the encoder to perform linear probe or fine-tuning.

These works inspire us to revisit the unification question. Our key insight in this work is that generation is viewed as "reconstructing" images that are 100% masked, while representation learning is viewed as "encoding" images that are 0% masked. We can therefore enable a unified architecture by using a *variable masking ratio* during MIM pre-training. The model is trained to reconstruct over a *wide range* of masking ratios covering high masking ratios that enable generation capabilities, and lower masking ratios that enable representation learning. This simple but very effective approach allows a smooth combination of generative training and representation learning in the *same framework*: same architecture, training scheme, and loss function.

However, directly combining existing MIM methods with a variable masking ratio is insufficient for high quality generation because such methods typically use a simple reconstruction loss on pixels, leading to blurry outputs. For example, as a representative of such methods, the reconstruction quality of MAE [27] is poor: fine details and textures are missing (Figure 2). A similar issue exists in many other MIM methods [11, 36].

This paper focuses on bridging this gap. We propose MAGE, a framework that can both generate realistic images and extract high-quality representations from images. Besides using variable masking ratio during pre-training, unlike previous MIM methods whose inputs are pixels, both the inputs and the reconstruction targets of MAGE are *semantic tokens*. This design improves both generation and representation learning, overcoming the issue described above. For generation, as shown in Figure 2, operating in token space not only allows MAGE to perform image generation tasks iteratively (subsection 3.2), but also enables MAGE to learn a probability distribution of the masked tokens instead of an average of all possible masked pixels, leading to diverse generation results. For representation learning, using tokens as inputs and outputs allows the network to operate at a high semantic level without losing low-level details, leading to significantly higher linear probing performances than existing MIM methods.

We evaluate MAGE on multiple generative and representation downstream tasks. As shown in Figure 1, for class-**un**conditional image generation on ImageNet-1K, our method surpasses state of the art with both ViT-B and ViT-

L (ViT-B achieves 11.11 FID [29] and ViT-L achieves 9.10 FID), outperforming the previous state-of-the-art result by a large margin (MaskGIT [7] with 20.68 FID). This significantly push the limit of class-unconditional generation to a level even close to the state-of-the-art of class-conditional image generation (∼6 FID [7, 50]), which is regarded as a much easier task in the literature [40]. For linear probing on ImageNet-1K, our method with ViT-L achieves 78.9% top-1 accuracy, surpassing all previous MIM-based representation learning methods and many strong contrastive baselines such as MoCo-v3 [13]. Moreover, when combined with a simple contrastive loss [9], MAGE-C with ViT-L can further get 80.9% accuracy, achieving state-of-the-art performance in self-supervised representation learning. We summarize our contributions:

- We introduce MAGE, a novel method that unifies generative model and representation learning by a single token-based MIM framework with variable masking ratios, introducing new insights to resolve the unification problem.
- MAGE establishes a new state of the art on the task of class-unconditional image generation on ImageNet-1K.
- MAGE further achieves state of the art in different downstream tasks, such as linear probing, few-shot learning, transfer learning, and class-conditional image generation.

2. Related Work

Self-supervised Learning in Computer Vision. Early work on unsupervised representation learning has focused on designing pretext tasks and training the network to predict their pseudo labels. Such tasks include solving jigsaw puzzles [43], restoring a missing patch [45], or predicting image rotation [22]. These pretext tasks result in representations that significantly trailed supervised training.

Contrastive learning [8, 10, 44] has proven to be a competitive and systematic method to learn effective representations without human supervision, getting performance very close to that of supervised pre-training. A number of variants of contrastive learning have been proposed: SimCLR [8] uses a large batch size, and samples negative pairs within each batch; momentum-contrastive approach (MoCo) [28] leverages a moving-average encoder and a queue to store negative samples during training; Contrastive-Multiview-Coding [55] maintains a memory-bank to store features and generate negative samples. Some recent methods, like BYOL, do not rely on negative pairs [12, 24]. Instead, they use two neural networks that learn from each other to boost performance.

Recently, vision researchers have found that masked image modeling (MIM), modeled after techniques in NLP e.g. [15], is a very effective task for self-supervised learning. BEiT [2] recovers discrete visual tokens from masked inputs. PeCo [18] further regards MoCo-v3 [13] as the perceptual model in VQGAN training to get a better tokenizer. MAE [26] considers MIM as a denoising pixel-level reconstruction task, and CMAE [32] further combines MAE with a contrastive loss. Some other methods such as MaskFeat [58] and MVP [59] predict features generated from teacher models.

However, current self-supervised learning methods based on MIM favor the performance of the representations on downstream tasks instead of the quality of the reconstructed images, leading to poor reconstructive results [2, 26]. Our paper for the first time shows that a single model can not only learn high-level fine-grained representations, but also be used to generate images of high visual fidelity.

Generative Models for Image Synthesis. Recent years have witnessed tremendous progress in deep generative models for image synthesis. One major stream of generative models is built on top of generative adversarial networks (GANs) [4, 23, 33, 64, 65]. GAN-based models can generate realistic images in various domains, but suffer from training instabilities and mode collapse issues. Another stream is based on a two-stage scheme [7, 34, 48, 56, 61]: first tokenize the image into a latent space and then apply maximum likelihood estimation and sampling in the latent space. VQVAE-2 [48] first shows this two-stage scheme can generate more diverse samples than GANs. ViT-VQGAN [61] uses ViT-based [19] encoder and decoder to get the latent code and then apply autoregressive generation in the latent space. MaskGIT [7] explores using a bidirectional transformer for token modeling and proposes parallel decoding for faster inference speed. Very recently, diffusion models [16, 30, 50, 53] have also achieved superior results on image synthesis.

However, the above generative models lack the ability to extract high-quality semantic representations from images. Works such as [17] and [61] explore the possibility of using latent features as representations, but their performance is sub-optimal. Our method surpasses previous generative models on both class unconditional generation and representation learning by a large margin, showing that a unified, high-performance framework is feasible.

3. Method

MAGE is a unified framework for both generative tasks and representation learning. To enable such unification, we first use a pre-trained VQGAN model [21] to quantize input images into semantic tokens. Then we randomly mask out some input tokens using a variable masking ratio ranging from 0.5 to 1 (see Figure 3), and apply an encoder-decoder transformer architecture on the remaining (unmasked) tokens to predict the masked tokens. We can further improve the separability of the learned representation by adding a simple yet effective contrastive loss similar to SimCLR [9] on the output of the encoder (MAGE-C). Below, we de-

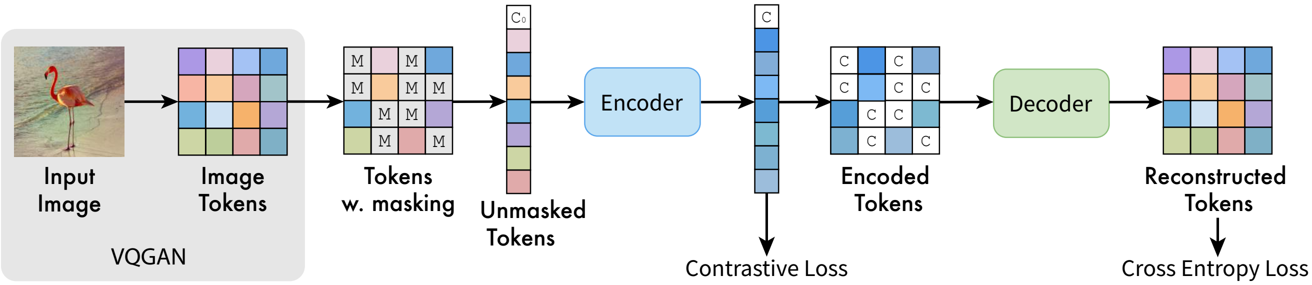


Figure 3. MAGE Framework: we first use a VQGAN tokenizer to tokenize the input image into a sequence of semantic tokens. We then sample a masking ratio (see text for details on the sampling strategy) and randomly mask out tokens according to this sampled ratio. A ViT encoder-decoder structure processes the unmasked tokens. A reconstructive cross-entropy loss encourages the model to reconstruct masked tokens. We can also add an optional contrastive loss at the output of the encoder to further improve the linear separability of the learned latent feature space.

scribe our design in detail.

3.1. Pre-training

Tokenization. We first tokenize the input image into a sequence of semantic tokens using a tokenizer. The tokenizer employs the same setup as the first stage in the VQGAN model [21]. This tokenization step allows our model to operate on semantic tokens instead of raw pixels, which is beneficial for both generation and representation learning as shown in Figure 2 and Figure 6.

Masking Strategy. To further bridge the gap between generative modeling and representation learning, we adopt a masking strategy with variable masking ratios. Specifically, we first randomly sample the masking ratio m_r from a truncated Gaussian distribution centered at 0.55, left truncated by 0.5, and right truncated by 1. If the length of the input sequence of tokens is l, we randomly mask out $m_r \cdot l$ tokens and replace them with a learnable mask token `[M]` (Figure 3). Since $m_r \geq 0.5$, we further randomly drop out $0.5 \cdot l$ tokens from those masked tokens. Dropping a large fraction of masked tokens significantly reduces overall pre-training time and memory consumption, while helping both generation and representation performance. This is consistent with the findings in MAE [26] for representation performance.

Encoder-Decoder Design. After masking and dropping input tokens, following [20], we concatenate a learnable "fake" class token `[C`$_0$`]` to the input sequence. The concatenated sequence is then fed into a Vision Transformer (ViT) [20] encoder-decoder structure. Specifically, the ViT encoder takes the sequence of tokens after masking and dropping as input and encodes them into latent feature space. Before decoding, the output of the encoder is first padded to the full input length using the class token feature `[C]` learned by the encoder. As shown in MAE [26], the class token position can summarize global features of the input image. Thus, instead of using a learnable masking token that is shared across different images, we use `[C]` that is specific to each image to pad the encoder outputs. We show in the Appendix that this design improves both generation and representation learning performance over using a masking token (as done in MAE [26]). The decoder then takes the padded features to reconstruct the original tokens.

Reconstructive Training. Let $Y = [y_i]_{i=1}^N$ denote the latent tokens obtained from the tokenizer, where N is the token sequence length, and $M = [m_i]_{i=1}^N$ denotes a corresponding binary mask determining which tokens are to be masked. The training objective is to reconstruct the masked tokens from the unmasked tokens. Therefore, we add a cross-entropy loss between the ground-truth one-hot tokens and the output of the decoder. Specifically,

$$\mathcal{L}_{reconstructive} = -\mathbb{E}_{Y \in \mathcal{D}}\Big(\sum_{\forall i, m_i=1} \log p(y_i | Y_M) \Big), \quad (1)$$

where Y_M are the (subset of) *unmasked* tokens in Y and $p(y_i|Y_M)$ is the probability predicted by the encoder-decoder network, conditioned on the unmasked tokens. Following MAE, we only optimize this loss on *masked* tokens (optimizing the loss on all tokens reduces both generation and representation learning performance, similar to the observations in [26]).

Contrastive Co-training. As shown in [35] and [32], adding a contrastive loss in MIM method can further improve its representation learning performance. In our MAGE framework, we can also add a contrastive loss to force better linear separability of the learned feature space. Similar to SimCLR [10], we add a two-layer MLP on top of the feature obtained by globally average pooling the output of the encoder. We then add an InfoNCE loss [44] on the output of the MLP head:

$$\mathcal{L}_{contrastive} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{e^{z_i^T \cdot z_i^+ / \tau}}{\sum\limits_{j=1}^{B} e^{z_i^T \cdot z_j / \tau}}, \quad (2)$$

where z denotes the normalized features after the two-layer MLP, B denotes the batch size, and τ denotes the temperature. The positive pairs z_i, z_i^+ are from two augmented views of the same image, and the negative samples z_j are all other samples in the same batch. Our final loss is:

$$\mathcal{L} = \mathcal{L}_{reconstructive} + \lambda \cdot \mathcal{L}_{contrastive} \qquad (3)$$

where $\lambda = 0.1$ balances the scale of the two losses. We do not use the extensive augmentations typically used in contrastive learning, such as color jitter, random grey scale, or gaussian noise. This is because the reconstructive loss acts as a regularizer that prevents the encoder from learning shortcut solutions [49]. Our approach achieves superior performance on both generative tasks and representation learning even without the contrastive loss, and representation learning performance can be further boosted with the contrastive loss.

3.2. Post-training Evaluation

To generate images for generative model evaluation, we use a **iterative decoding** strategy similar to MaskGIT [7]. We start from a blank image with all the tokens masked out. At each iteration, our model first predicts the tokens for the remaining masked tokens. Then we sample some of the predicted tokens (tokens that have a higher predicted probability are of higher probability to be sampled) and replace the corresponding masked tokens with these sampled predicted tokens. The number of masked tokens to be replaced in each iteration follows a cosine function, i.e., we replace fewer masked tokens in the early iterations and more masked tokens at later iterations. We use a total of 20 steps to generate an image. For representation learning, we globally average pool the features output from the ViT encoder, and use the pooled features as the input features for the classification head. A detailed description of our pre-training and evaluation implementations and architectures is provided in the Appendix.

4. Results

MAGE is a unified framework for both generative model and representation learning. In this section, we conduct extensive experiments to evaluate the generation as well as visual representation capabilities. To evaluate MAGE's generative performance, we conduct experiments on ImageNet-1K dataset [51] for the task of class-unconditional image generation. To evaluate the quality of the learned representations, we conduct experiments on ImageNet-1K dataset [51] under two protocols: first is linear probing, where we add a linear classification head on top of the learned representations and only train the classification head, while keeping the backbone frozen; second is fine-tuning, where we fine-tune the whole parameters for the classification task. We also include results on few-shot learning and transfer learning to better evaluate the quality of the representations. More results and ablation studies can be found in the Appendix.

4.1. Pre-training Setup

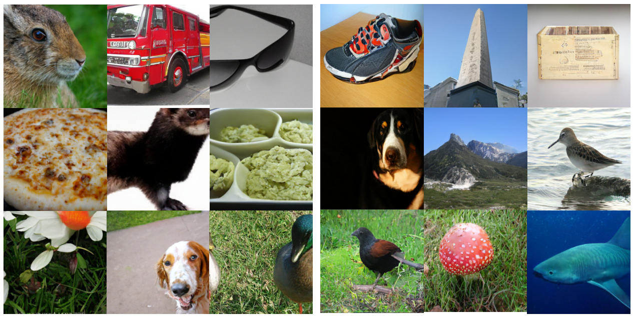
We set the input image resolution as 256x256 to be consistent with previous generative models. After passing through the VQGAN tokenizer, the token sequence length is 16x16 (256 tokens). Following MAE [26], we use strong random crop and resize (0.2 to 1) and random flipping as our default augmentations. We also trained models with a weaker version of random crop and resize (range from 0.8 to 1), which we call "w.a." in the results. We pre-train base- and large-size vision Transformers [20], i.e., ViT-B and ViT-L, respectively. We use AdamW to train the model for 1600 epochs with batch size of 4096 for ViT-B, and batch size of 2048 for ViT-L. We use a cosine learning rate schedule with an 80-epoch warmup. The base learning rate is 1.5×10^{-4} for both ViT-B and ViT-L, and is further scaled by batchsize/256. More details are in the Appendix.

4.2. Image Generation

Table 1. Quantitative comparison with state-of-the-art generative models on ImageNet 256x256 for class-unconditional generation. The number of parameters includes encoder, decoder, and detokenizer.

Methods	RES	FID↓	IS↑	#params
Self-Conditioned GAN [37]	128	40.3	15.82	-
BigGAN [17]	256	38.6	24.70	~70M
BigGAN [17]	128	30.9	23.56	~70M
BigGAN+Clustering [40]	128	22.0	23.5	~70M
HiT [66]	128	30.8	21.64	~30M
ADM [16]	256	26.2	39.70	554M
MaskGIT [7]	256	20.7	42.08	203M
MAGE (ViT-B)	256	11.1	81.17	176M
MAGE (ViT-B, w.a.)	256	**8.67**	**94.8**	176M
MAGE (ViT-L)	256	9.10	105.1	439M
MAGE (ViT-L, w.a.)	256	**7.04**	**123.5**	439M

Class-Unconditional Image Generation. Our pre-trained model can naturally perform class-unconditional image generation without any fine-tuning on the model parameters. Table 1 compares the class-unconditional image generation results of our model and SOTA generative models on ImageNet, reporting Frechet Inception Distance (FID) [29] and Inception Score (IS) [52] as standard metrics. As shown in the table, our method outperforms all previous image generation methods by a large margin. The previous SOTA can only achieve 20.7 FID and 42.08 IS, while our ViT-B model can achieve 11.1 FID and 81.17 IS with similar number of parameters. This is likely because our framework can extract much better representations than all previous generative models as shown in Table 2, leading to superior generative performance. Our ViT-L model further achieves 9.10 FID and 7.04 FID when trained with weak augmentation, which is very close to the *class-conditional* generation performance of transformer



(a) Default augmentation (b) Weak augmentation

Figure 4. Images generated by MAGE (ViT-L). (a) images generated from MAGE trained with default strong augmentation, i.e., crops out larger portion of the image. (b) images generated from MAGE trained with weak augmentations, i.e., crops out smaller portion of the image. We see that visual fidelity and diversity are very good for both models.

models (e.g. 6.18 FID in MaskGIT [7]), a much easier task than class-unconditional generation [40].

We also note that the augmentations used to train the tokenizer and the encoder-decoder model can affect the evaluation scores. As shown in Table 1 and Figure 4, with default "strong" augmentation (i.e. random resized crop scale from 0.2 to 1), the FID and IS of the model are worse than using a weaker augmentation (random resized crop scale from 0.8 to 1). One possible reason is that the ImageNet validation set used to compute the FID is resized to 256 and center cropped. Since FID is computed based on the similarity between the generated image and the images in ImageNet validation set, it will be higher if the scale of the generated image is smaller. However, this does not necessarily mean that the visual quality of the generated image is worse. As shown in Figure 4, images generated with default augmentation can be much more zoomed in and off-center, but the images are still realistic and of high quality. We include more results in the Appendix, including for class conditional generation and image editing tasks such as inpainting.

4.3. Image Classification

Linear Probing. Linear probing is a primary evaluation protocol for self-supervised learning. As shown in Table 2, MAGE outperforms MAE [26] by 6.7% on ViT-B and 3.1% on ViT-L for ImageNet-1K linear probe top-1 accuracy, achieving state-of-the-art results among all MIM methods. Moreover, a simple contrastive loss similar to SimCLR [8] can further boost our performance. We do not use color jitter, random grey scale, or multi-crop augmentations used in SwAV [5], DINO [5] and iBOT [67]. Multi-crop augmentation typically brings 3%-5% improvements on accuracy, but introduces large computational overheads. In spite of no multi-crop, MAGE-C achieves 78.2% accuracy with ViT-B and 80.9% accuracy with ViT-L. Our ViT-B performance surpasses that of ViT-H in MoCo v3 (632M parameters), indicating that the extra parameters (24M) in the tokenizer are *not* the reason for our good performance.

Table 2. Top-1 accuracy of linear probing on ImageNet-1k. † denotes methods which require additional teacher model (CLIP) trained from image-text data. * denotes methods using multi-crop augmentations. RN is short for ResNet. The number of parameters for MAGE includes VQ-GAN tokenizer and ViT encoder.

Methods	Model	#params	Acc.
Generative models			
BigBiGAN [17]	RN50	23M	56.6
MaskGIT [7]	BERT	227M	57.4
ViT-VQGAN [61]	VIM-Base	650M	65.1
ViT-VQGAN [61]	VIM-Large	1697M	73.2
MIM methods			
BEiT [2]	ViT-B	86M	56.7
MAE [26]	ViT-B	86M	68.0
Ge^2-AE [36]	ViT-B	86M	**75.3**
MAGE	ViT-B	24M+86M	74.7
MAE [26]	ViT-L	304M	75.8
MAGE	ViT-L	24M+304M	**78.9**
Contrastive methods			
SimCLRv2 [10]	RN50w2	94M	75.6
BYOL [24]	RN50w2	94M	77.4
CAE [11]	ViT-B	86M	70.4
CMAE [32]	ViT-B	86M	73.9
MoCo v3 [13]	ViT-B	86M	76.7
DINO [67]	ViT-B	86M	72.8
iBOT [67]	ViT-B	86M	76.0
MAGE-C	ViT-B	24M+86M	**78.2**
SimCLRv2 [10]	RN152w2	233M	77.4
BYOL [24]	RN200w2	250M	79.6
MoCo v3 [13]	ViT-L	304M	77.6
CAE [11]	ViT-L	304M	78.1
MoCo v3 [13]	ViT-H	632M	78.1
MAGE-C	ViT-L	24M+304M	**80.9**
Additional Data/Aug.			
MVP† [59]	ViT-B	86M	75.4
BEiT v2† [46]	ViT-B	86M	80.1
SwAV* [5]	RN50w5	586M	78.5
DINO* [6]	ViT-B	86M	78.2
iBOT* [67]	ViT-B	86M	79.5
iBOT* [67]	ViT-L	304M	81.0

Few-shot Learning. The premise of self-supervised learning is to learn representations on unlabeled data that can be effectively applied to prediction tasks with few labels [10]. Following [19], we freeze the weights of the pretrained model and train a linear classifier on top using a few labeled samples. As shown in Table 3, our methods with ViT-B outperform MAE [26] by a very large margin and achieves similar performance as MSN [1], which is the

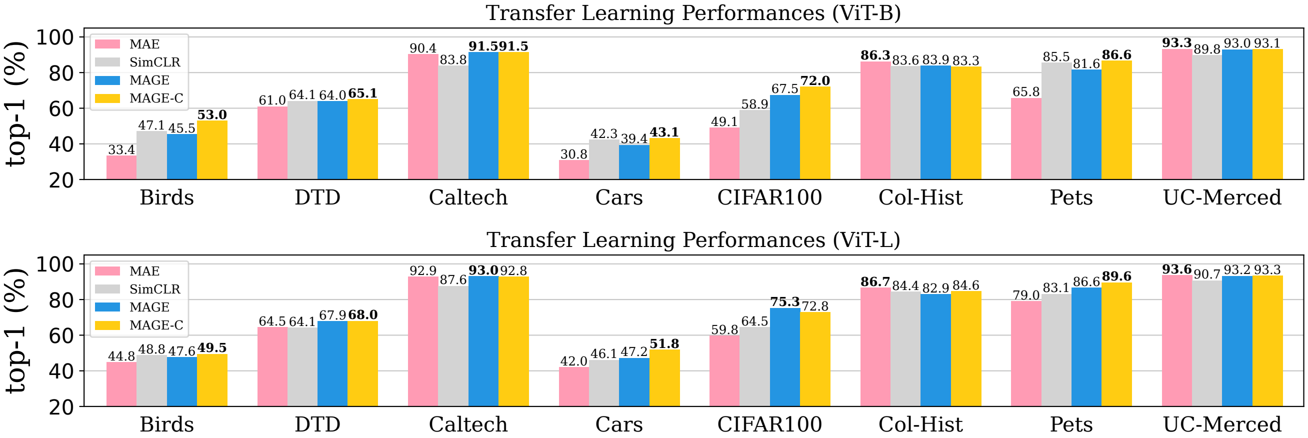


Figure 5. Transfer learning performance of ViT-B and ViT-L pre-trained on ImageNet-1K using different methods. Our methods outperforms SimCLR [9] and MAE [26] on 6 of the 8 datasets.

Table 3. Few-shot evaluation on ImageNet-1K. We report the top-1 accuracy with different self-supervised methods and different numbers of the ImageNet-1K labels used. We report the accuracy of MAE under our implementation (denoted by †). Note that MSN [1] uses multi-crop augmentation.

Method	Training images per ImageNet Class			
	5	10	13	25
ViT-B				
MAE† [26]	29.2	34.5	-	38.7
MSN [1]	**65.5**	-	**69.6**	-
MAGE	53.5	58.4	59.7	61.7
MAGE-C	62.7	66.9	67.8	69.1
ViT-L				
MAE† [26]	42.2	47.7	-	51.7
MSN [1]	-	-	70.1	-
MAGE	60.3	66.1	67.8	69.6
MAGE-C	**68.1**	**71.9**	**73.0**	**74.2**

Table 4. Fine-tuning performance on ImageNet-1K. We report the top-1 accuracy and the improvement over training-from-scratch for different methods (other numbers taken from the respective papers). The ViT models trained from scratch on semantic tokens follow the exact same training setting as the ViT models trained from scratch on original image pixels in [26].

Method	ViT-B	ViT-L
scratch on pixels	82.3	82.6
DINO [6]	82.8 (+0.5)	-
MoCo v3 [13]	83.2 (+0.9)	84.1 (+1.5)
BEiT [2]	83.2 (+0.9)	85.2 (+2.6)
MAE [26]	83.6 (+1.3)	85.9 (+3.3)
CAE [11]	83.9 (+1.6)	86.3 (+3.7)
MVP [59]	84.4 (+2.1)	86.3 (+3.7)
PeCo [18]	84.5 (+2.2)	86.5 (+3.9)
scratch on tokens	80.7	80.9
MAGE	82.5 (+1.8)	83.9 (+3.0)
MAGE-C	82.9 (+2.2)	84.3 (+3.4)

state-of-the-art method for self-supervised label-efficient learning. Moreover, the performance of MAGE-C with ViT-L even surpasses the performance of MSN using 13 images per class (1% of ImageNet-1K), even though MSN uses multi-crop augmentation.

Transfer Learning. Another important property of self-supervised representation is its transferability to different datasets. Following the protocol in [19], we evaluate the transfer learning performance of MAGE pre-trained on ImageNet-1K on 8 datasets under a few-shot setting (25 samples per class). Results are shown in Figure 5: we see that MAGE's superior performance on ImageNet-1K translates to strong performance on other tasks. Since our method operates on quantized semantic tokens instead of raw pixels, it is likely to be more robust to domain shift.

Fine-tuning. Table 4 shows the fine-tuning performance of MAGE and other self-supervised learning methods, when we can change all the pre-trained encoder parameters. Our method achieves performance at par with DINO [5] and slightly under MoCo-v3 [13]. We believe that the use of quantized tokens leads to a subpar from-scratch and fine-tune performance, and leave further investigations of this phenomenon to future work. We note, however, that our method still improves over our supervised training-from-scratch baseline by as large a margin as other methods.

4.4. Analysis

In this section, we analyze the two key components of MAGE that enables the unification of generative modeling and representation learning: variable masking ratio and quantized tokenization. All experiments are conducted on ViT-B. Experiments on variable masking ratio are all trained for 400 epochs, and experiments on quantized tokenization are all trained for 1600 epochs. More analysis and ablation studies are in the Appendix.

Table 5. Top-1 accuracy of linear probing and class unconditional generation FID of MAGE on ImageNet-1k with different masking ratio distribution. μ denotes the mode and σ the standard deviation of the truncated Gaussian distribution. When $\sigma = 0$, the masking ratio is fixed and generation has poor quality with very high FID (>50). Therefore we put N/A for FID in such cases.

	$\mu = 0.7$	$\mu = 0.6$	$\mu = 0.55$	$\mu = 0.5$	$\mu = 0.45$	$\mu = 0.55$	$\mu = 0.55$	$\mu = 0.55$	$\mu = 0.55$
	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0.15$	$\sigma = 0.25$	$\sigma = 0.5$
Linear Probing	69.7	70.1	71.5	70.9	70.4	71.5	72.0	**72.2**	71.8
FID	N/A	N/A	N/A	N/A	N/A	N/A	12.5	**12.2**	13.0

Table 6. Reconstruction loss and linear probe accuracy of MAGE with unquantized features and quantized tokens as input. Using unquantized features makes it much easier to infer masked tokens, and hence results in worse linear probe performance.

inputs	recon. loss	linear probe (%)
Unquantized features	3.31	49.5
Quantized tokens	5.76	74.7

Masking Design. Variable masking ratio is one of our key components. We find that the quality of the learned representation can also be affected by the distribution used to sample our masking ratio. We compare the results of MAGE on linear probing and class unconditional generation on ImageNet-1k using different masking ratio distributions in Table 5. We denote the mode of the truncated Gaussian distribution as μ, and the standard deviation of the truncated Gaussian distribution as σ. Note that $\sigma = 0$ indicates a fixed masking ratio. The left 5 columns ablate μ, and the right 4 columns ablate σ. The results show that a variable masking ratio is necessary to enable generation. Moreover, using a variable masking ratio also enables representation learning to learn better features and achieve better linear probe performance.

Tokenization. Previous self-supervised learning methods on images typically directly use raw images as the inputs of the transformer. However, in MAGE we use quantized semantic tokens as both inputs and reconstruction targets. We elaborate the benefits of this design as follows:

- First, during generation, it allows the network to iteratively use its output as the input in the next iteration, which enables high-quality and diverse image reconstruction and generation, as shown in Figure 2 and Figure 4.
- Second, it allows the whole network to operate at a semantic level without losing low-level details and thus extract better representations. We demonstrate this by comparing the linear probe performance on features from each transformer block of ViT-B trained using MAE and MAGE. As shown in Figure 6, the linear probe accuracy of MAGE at each transformer block is always higher than MAE throughout the encoder.
- Third, the quantizer prevents shortcuts created by the VQGAN CNN encoder. If we directly use features extracted by the VQGAN encoder without quantization as input to the transformer, since the receptive fields of neighboring feature pixels have significant overlap, it is much easier to infer masked feature pixels using nearby unquantized feature pixels. As shown in Table 6, with the same masking strategy, using unquantized features achieves much lower reconstructive loss (3.31 vs. 5.76), but also a much lower linear probe accuracy (49.5% vs. 74.7%). This suggests that the pre-training task is too easy, leading to shortcut solutions, and hence to poor representations. The quantization step is therefore necessary to learn good representations.

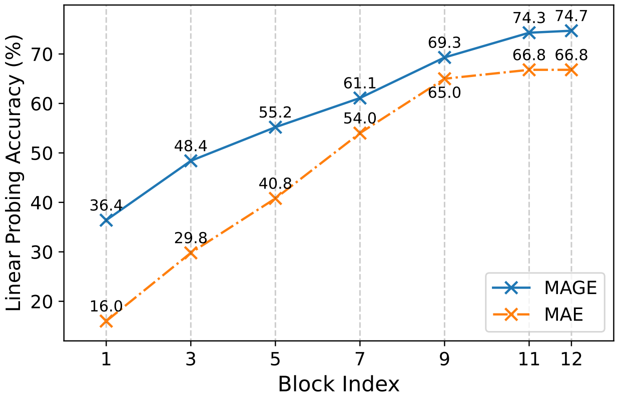


Figure 6. Linear probe accuracy of MAE and MAGE at different transformer blocks of ViT-B. MAGE consistently has higher accuracy across all transformer blocks due to the semantic nature of the quantized tokens.

5. Discussion

We have presented MAGE, a masking-based approach that unifies image generation and representation learning in a simple and effective framework. The key to our method is the use of quantized tokens and the use of variable masking ratios that adapt smoothly to both tasks (generation and representation). We have shown extensive results on linear probing, few-shot transfer learning, and unconditional image generation. To the best of our knowledge, this is the first model that achieves close to SOTA results for both tasks using the same data and training paradigm. A natural future extension is to pre-train on larger unlabeled datasets such as JFT300 to further improve performance.

6. Acknowledgement

We thank Google Cloud Platform (GCP) and MIT-IBM Watson Research Collaboration for providing necessary computation resources for this project.

References

[1] 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. *arXiv preprint arXiv:2204.07141*, 2022.

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

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. on Learning Representations (ICLR)*, 2019.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[6] 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)*, pages 9650–9660, 2021.

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

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

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *icml*, pages 1597–1607. PMLR, 2020.

[10] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33, 2020.

[11] 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. *arXiv preprint arXiv:2202.03026*, 2022.

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

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

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

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

[16] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[17] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in neural information processing systems*, 32, 2019.

[18] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

[19] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[20] 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)*, 2021.

[21] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[22] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[23] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. 2014.

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. `https://github.com/facebookresearch/mae`, 2021.

[26] 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), pages 16000–16009, June 2022.

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[29] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

[31] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[32] 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.

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[34] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] Tianhong Li, Lijie Fan, Yuan Yuan, Hao He, Yonglong Tian, Rogerio Feris, Piotr Indyk, and Dina Katabi. Making contrastive learning robust to shortcuts. *arXiv preprint arXiv:2012.09962*, 2020.

[36] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. *arXiv preprint arXiv:2204.08227*, 2022.

[37] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14286–14295, 2020.

[38] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[40] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International conference on machine learning*, pages 4183–4192. PMLR, 2019.

[41] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

[42] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[43] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

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

[45] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[46] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[47] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[48] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[49] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986, 2021.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[52] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[53] 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.

[54] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[56] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[58] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[59] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. *arXiv preprint arXiv:2203.05175*, 2022.

[60] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[61] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[62] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[63] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[64] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Int. Conference on Machine Learning (ICML)*, pages 7354–7363, 2019.

[65] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

[66] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris Metaxas, and Han Zhang. Improved transformer for high-resolution gans. *Advances in Neural Information Processing Systems*, 34:18367–18380, 2021.

[67] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

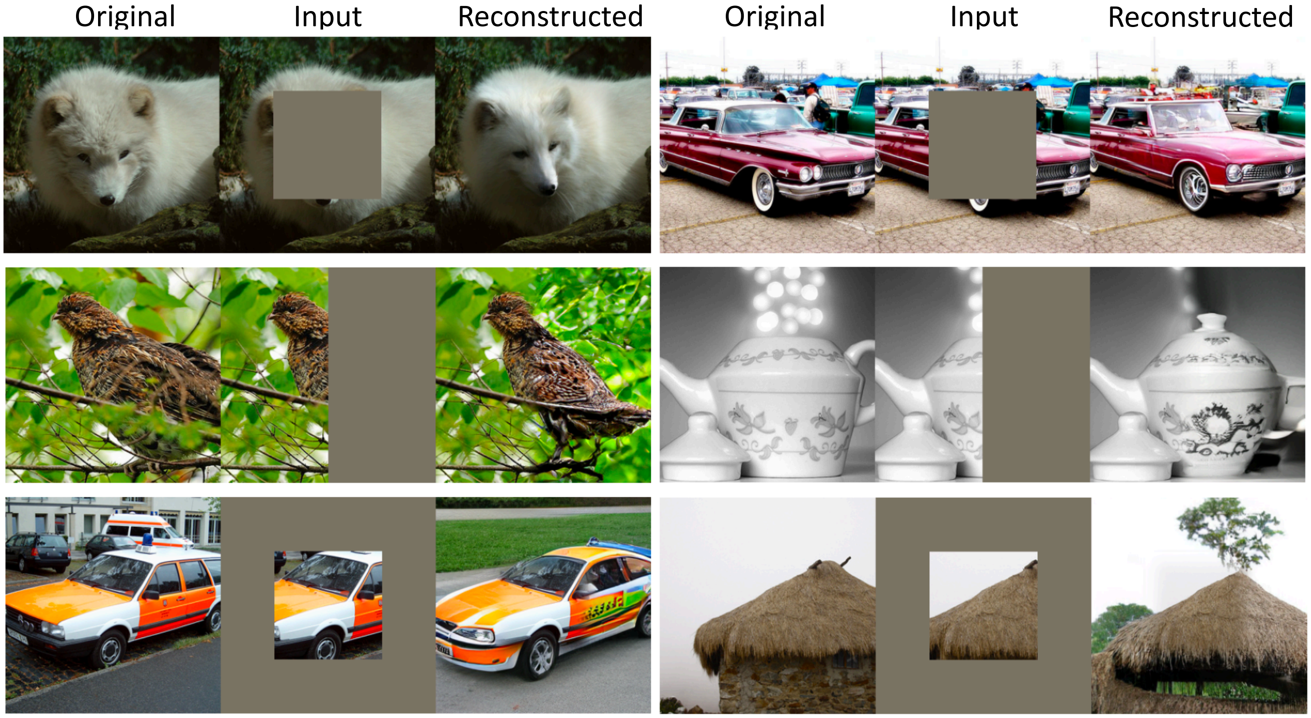


Figure 7. Examples of image inpainting (first row), outpainting (second row), and uncropping (outpainting on a large mask, third row) using MAGE (ViT-L).

A. Additional Results

A.1. Qualitative Results

Image Inpainting and Outpainting. With the superior class-unconditional reconstruction and generation power shown in the main paper, MAGE naturally enables many common image synthesis applications. As shown in Figure 7, MAGE can reconstruct realistic and high-quality images for different image editing tasks such as inpainting (first row), outpainting (second row), and uncropping (outpainting on large masks, third row). We also include more qualitative results in Figure 11, Figure 12, and Figure 13, demonstrating MAGE's excellent ability in such image synthesis tasks. All results are generated using MAGE based on ViT-L trained with default augmentations, and the original images are all from the ImageNet eval set.

Class Unconditional Generation. We include more class unconditional generation results using default strong augmentation (random crop and resize (0.2 to 1) and random flipping) and weak augmentation (random crop and resize (0.8 to 1) and random flipping) in Figure 9 and Figure 10.

A.2. Quantitative Results

Table 7. Quantitative comparison with state-of-the-art generative models on ImageNet 256x256 for class-conditional generation. Our method uses a MAGE pre-trained ViT-B as encoder and only trains a class-conditional decoder with 113M parameters.

Methods	FID↓	IS↑	#params
DCTransformer [41]	36.51	-	738M
VQVAE-2 [48]	31.11	~45	13.5B
VQGAN [7]	18.65	80.4	227M
VQGAN [21]	15.78	78.3	1.4B
Improved DDPM [42]	12.26	-	280M
ADM [16]	10.94	101.0	554M
LDM [50]	10.56	103.5	400M
BigGAN-deep [3]	6.95	198.2	160M
MaskGIT [7]	6.18	182.1	227M
MAGE (ViT-B)	6.93	195.8	117M+113M

Class-Conditional Image Generation. Our model can also be used for class-conditional image generation as downstream task. To enable class-conditional generation, we take the ViT encoder from pre-training, and replace the original ViT decoder with a class-conditional decoder (12-layer ViT with embedding dimension 768, 113M parameters) which takes the class label as another input (concatenated to the padded features). During training, we freeze the encoder parameters to better evaluate the quality of the learned representations. Similar to pre-training, the model will take masked tokenized images as input and try to reconstruct the masked tokens. The only difference is that the decoder will not only see representations from the encoder, but also know the class label of the input image. Then during inference, the class label will be used to guide the model to generate images of the same class.

As shown in Table 7, MAGE achieves comparable performance as SOTA image generation methods on the task of class-conditional image generation on ImageNet-1K. Note that our encoder is inherited from the pre-training and *is not* fine-tuned during the downstream class-conditional training. Only the decoder is trained and has information about the class label. This shows that MAGE's encoder can learn high-quality representations that can achieve similar generative performance as models trained end-to-end on class conditional generative tasks.

Few-shot Transfer Learning. In the main paper, we provide transfer learning results of MAGE on 8 different datasets with 25 samples per class. Here we further show our method's performance with 1, 5, and 10 samples per class. As shown in Figure 8, our method is consistently better than MAE and SimCLR on most datasets with different numbers of samples per class, demonstrating the effectiveness of our method on few-shot transfer learning.

B. Ablation Studies

In this section, we conduct extensive ablation studies on our method. Without further notice, we use ViT-B trained with 800 epochs for all ablation studies.

Table 8. Top-1 accuracy of linear probing on ImageNet-1k with different method. MAE with GAN loss significantly reduces its performance on linear probing.

Methods	Linear Probing (%)
MAE (ViT-L) [26]	75.1
MAE (ViT-L) + norm pixel loss [26]	75.8
MAE + GAN loss (ViT-L) [25]	64.1
MAGE	**78.9**

MAE with GAN loss. One trivial solution to force the previous MIM method to generate realistic images is to add a GAN loss on top of the reconstructed image. However, we show that introducing GAN loss during previous MIM pre-training could largely decrease the performance of linear probing. As shown in Table 8, we evaluate the linear probing performance of a ViT-L MAE model pre-trained with an extra GAN loss released in MAE's official GitHub repo [25]. Although this model can reconstruct much more realistic images than the original MAE, the linear probing performance decreases by 11% compared with the ViT-L MAE model pre-trained without the GAN loss. On the other hand, our MAGE framework enables generative modeling and representation learning to help each other, achieving SOTA performances on both tasks using one single model.

Table 9. FID and top-1 accuracy of linear probing on ImageNet-1k by padding with `[C]` or a universal `[MASK]` token.

Padding Token	FID	Linear Probing (%)
`[MASK]`	12.4	72.5
`[C]`	11.6	73.3

Pad with [CLS] token. To pad the output of the encoder, unlike MAE which uses a learnable mask token that is shared for different inputs, we use the class token feature which is specific to each image. This design allows the decoder to take the global features extracted by the encoder as input. As shown in Table 9, this design can improve both class-unconditional generation performance and linear-probing results.

Table 10. FID and top-1 accuracy of linear probing of ViT-B trained 1600 epochs on ImageNet-1k using strong augmentations (s.a.) and weak augmentations (w.a.).

Augmentations	FID	Linear Probing (%)
MAGE + w.a.	**8.67**	70.5
MAGE + s.a.	11.1	**74.7**

Augmentations. As shown in many previous works on generative modeling and representation learning [9, 17, 26, 49], the augmentation used to train the model is important for both generation and representation learning performance. In our paper, we use two different sets of augmentations: default augmentations, or strong augmentations (s.a.), which consist of random crop and resize (0.2 to 1) and random flipping; weak augmentations (w.a.), which consist of random crop and resize (0.8 to 1) and random flipping. The only difference between s.a. and w.a. is the zoom-in scale of random crop and resize. As shown in Table 10, strong augmentations favor representation learning and weak augmentation favor generation quality, which is consistent with findings in prior works [17, 26].

Table 11. FID and top-1 accuracy of linear probing of ViT-B trained 400, 800, and 1600 epochs on ImageNet-1k.

#Pre-training Epochs	FID	Linear Probing (%)
400	12.2	72.2
800	11.6	73.3
1600	11.1	74.7

Pre-training Epochs. One important factor in self-supervised learning methods is the number of pre-training epochs. Prior works have shown that longer pre-training epochs can largely improve the performance of self-supervised methods [9, 26]. We compare MAGE's performance on ViT-B using 400, 800 and 1600 epochs of pre-training in Table 11. We observe that MAGE achieves good performances in both generation and representation learning with 400 epochs of pre-training, and can consistently benefit from longer training epochs.

Table 12. FID and top-1 accuracy of linear probing on ImageNet-1k using different decoder architecture. d denotes decoder depth (number of transformer blocs in the decoder), and w denotes decoder width (feature dimension in the decoder).

Decoder Arch.	FID	Linear Probing (%)
$d = 8, w = 512$	12.4	72.1
$d = 8, w = 768$	11.6	73.3
$d = 8, w = 1024$	11.4	73.5
$d = 6, w = 768$	12.4	71.8
$d = 8, w = 768$	11.6	73.3
$d = 10, w = 768$	11.4	73.2
$d = 12, w = 768$	11.3	73.4

Decoder Design. MAE [26] shows that a small ViT decoder is enough to achieve good performance. We also try different decoder architectures and summarize the results in Table 12. As shown in the table, the decoder with 8 blocks and 768 feature dimension reaches the best balance between computation cost and performance for ViT-B. Therefore, we choose the decoder architecture to be 8 blocks with 768 feature dimensions for ViT-B and 1024 feature dimension for ViT-L in the paper.

Table 13. Top-1 accuracy of linear probing on ImageNet-1k using different methods. C denotes our contrastive loss, R denotes our reconstructive loss, † denotes our re-implementation.

Methods	Linear Probing (%)
MAE [26]	68.0
R only	73.3
SimCLR † [9]	74.2
C only	72.9
C+R (MAGE-C)	**77.1**

Complement MIM with Contrastive Loss. We show in the main paper that MAGE can be further combined with a simple contrastive loss (MAGE-C) to achieve better representation learning performance. In Table 13 we show more ablations regarding this contrastive loss. The performance of simply applying the contrastive loss without the reconstructive loss is worse than the SimCLR baseline. This is likely because we do not use augmentations such as color jittering and random grey scale, so applying only the contrastive loss could result in learning shortcut semantics such as color distribution [9, 49]. However, the reconstructive loss can prevent the network from falling into such shortcut solutions and help the network learn richer semantics.

Table 14. FID and top-1 accuracy of linear probing of MAGE-C on ImageNet-1k using different maximum masking ratio $\max(m_r)$.

	FID	Linear Probing (%)
$\max(m_r)$=1.0	14.1	75.0
$\max(m_r)$=0.7	23.5	76.3
$\max(m_r)$=0.6	27.0	77.1

We also notice one problem of applying contrastive training to MAGE: MAGE can see very high masking ratios during training, but applying positive loss to two augmented views of the same image both with a very high masking ratio is problematic. This is because such two views can share very little common information, leading to a performance drop as shown in [55]. Therefore, we only apply contrastive loss when the masking ratio is relatively low ($m_r < 0.6$). In Table 14, we show the performance of generation and representation learning w.r.t. the maximum masking ratio of our variable masking ratio distribution. Smaller $\max(m_r)$ leads to better linear probing but worse FID. We believe it is because, with smaller $\max(m_r)$, the contrastive loss can operate on more samples in the batch whose masking ratio $m_r < 0.6$, which is important for contrastive learning as shown in [9]. On the other hand, small $\max(m_r)$ harms the generation performance because the network should see a relatively high masking ratio to enable generation from blank image (100% masking ratio). We leave a further investigation of this phenomenon and a better combination strategy for future work.

C. Implementation Details

Tokenizer and Detokenizer. We use a CNN-based VQGAN encoder and quantizer to tokenize the 256x256 input images to 16x16 discrete tokens. The detokenizer operates on the 16x16 discrete tokens and reconstructs the 256x256 image. The encoder consists of 5 blocks and each block consists of 2 residual blocks. After each block in the encoder, the feature is down-sampled by 2 using average pooling. The quantizer then quantizes each pixel of the encoder's output feature map using a codebook with 1024 entries, each entry with dimension 256. The detokenizer consists of another 5 blocks where each block consists of 2 residual blocks. After each block in the decoder, the feature map is up-sampled by 2. Please refer to our code and the original VQGAN paper for more details [21].

ViT architecture. After the tokenizer, the latent se-

Table 15. **Pre-training Setting.**

config	value
optimizer	AdamW [39]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096 (B), 2048 (L)
learning rate schedule	cosine decay [38]
warmup epochs	40
training epochs	1600
gradient clip	3.0
label smoothing [54]	0.1
dropout	0.1
masking ratio min	0.5
masking ratio max	1.0 (MAGE) 0.6 (MAGE-C)
masking ratio mode	0.55
masking ratio std	0.25
MAGE-C only	
contrastive loss weight	0.1
temperature	0.2

Table 16. **Linear Probing Setting.**

config	value
optimizer	LARS [60]
base learning rate	0.1 (B) 0.05 (L)
weight decay	0
optimizer momentum	0.9
batch size	4096
learning rate schedule	cosine decay [38]
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

quence length becomes 256 (plus one 'fake' class token). We then follow a similar encoder-decoder Transformer architecture similar to MAE [26]. More specifically, we use standard ViT architecture [20], which consists of a stack of Transformer blocks [57], where each block consists of a multi-head self-attention block and an MLP block. We use two learnable positional embeddings, one added to the input of the encoder and another added to the input of the decoder.

We use features from the encoder output for classification tasks, such as linear probing, few-shot transfer learning, and fine-tuning. We average pool the encoder output without the class token to get the input of the linear classifier.

Pre-training. Please refer to Table 15 for our default pre-training setting. We use only random crop and resize (0.2 to 1) and random horizontal flip as our default augmentations.

Generation. We use iterative decoding as in MaskGIT [7] to iteratively fill in masked tokens and generate images. To generate an image at inference time, we start from a blank canvas with all the tokens masked out, i.e., $Y_M^{(0)}$. For iteration $t = 1, \cdots, T$, the algorithm runs as follows:

1. **Predict.** Given $Y_M^{(t)}$ which is the unmasked tokens at the beginning of iteration t, we first predict the probability of the remaining masked tokens using our model, denoted as $p^{(t)} \in \mathbb{R}^{N_t \times K}$, where N_t is the number of remaining masked tokens and K is the number of entries in the codebook.

2. **Sample.** At each masked location i, we sample a token $y_i^{(t)}$ based on the prediction probability $p_i^{(t)} \in \mathbb{R}^K$ over all possible tokens in the codebook, and form the unmasked prediction $Y^{(t)}$. After $y_i^{(t)}$ is sampled, its corresponding prediction score plus a noise sampled from a random Gumbel distribution multiplied by temperature τ is used as the

Table 17. **End-to-end fine-tuning Setting.**

config	value
optimizer	AdamW [39]
base learning rate	2.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
layer-wise lr decay [2]	0.65 (B) 0.75 (L)
batch size	1024
learning rate schedule	cosine decay [38]
warmup epochs	5
training epochs	100 (B) 50 (L)
label smoothing [54]	0.1
augmentation	RandAug (9, 0.5) [14]
mixup [63]	0.8
cutmix [62]	1.0
random erase	0
drop path [31]	0.1 (B) 0.2 (L)

Table 18. **Supervised training from scratch setting with ViT on semantic tokens.**

config	value
optimizer	AdamW [39]
base learning rate	1e-4
weight decay	0.3
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
batch size	4096
learning rate schedule	cosine decay [38]
warmup epochs	20
training epochs	300 (B) 200 (L)
label smoothing [54]	0.1
augmentation	RandAug (9, 0.5) [14]
mixup [63]	0.8
cutmix [62]	1.0
drop path [31]	0.1 (B) 0.2 (L)
exp. moving average (EMA)	0.9999

”confidence” score indicating the model’s belief of the prediction at location i. The confidence scores at the unmasked locations are set to $+\infty$.

3. **Mask.** We then determine the number of tokens N_{t+1} for the next iteration $t+1$ based on a cosine masking schedule $N_{t+1} = N_0 \cdot \cos\left(\frac{\pi t}{2T}\right)$. We then sample N_{t+1} locations with the lowest confidence scores and mask those locations from $Y^{(t)}$ to generate $Y_M^{(t+1)}$.

For class unconditional generation, we choose $\tau = 6.0$ and $T = 20$ to generate images in our experiment.

Linear Probing & Fine-tuning. Our linear probing and fine-tuning setup follow MAE [26]. Please see Table 16 and Table 17 for detailed configurations.

Training from scratch. We also follows MAE [26] for our training-from-scratch baseline. Table 18 summarizes our configurations.

Code. For more implementation details, please refer to our code at https://github.com/LTH14/mage.

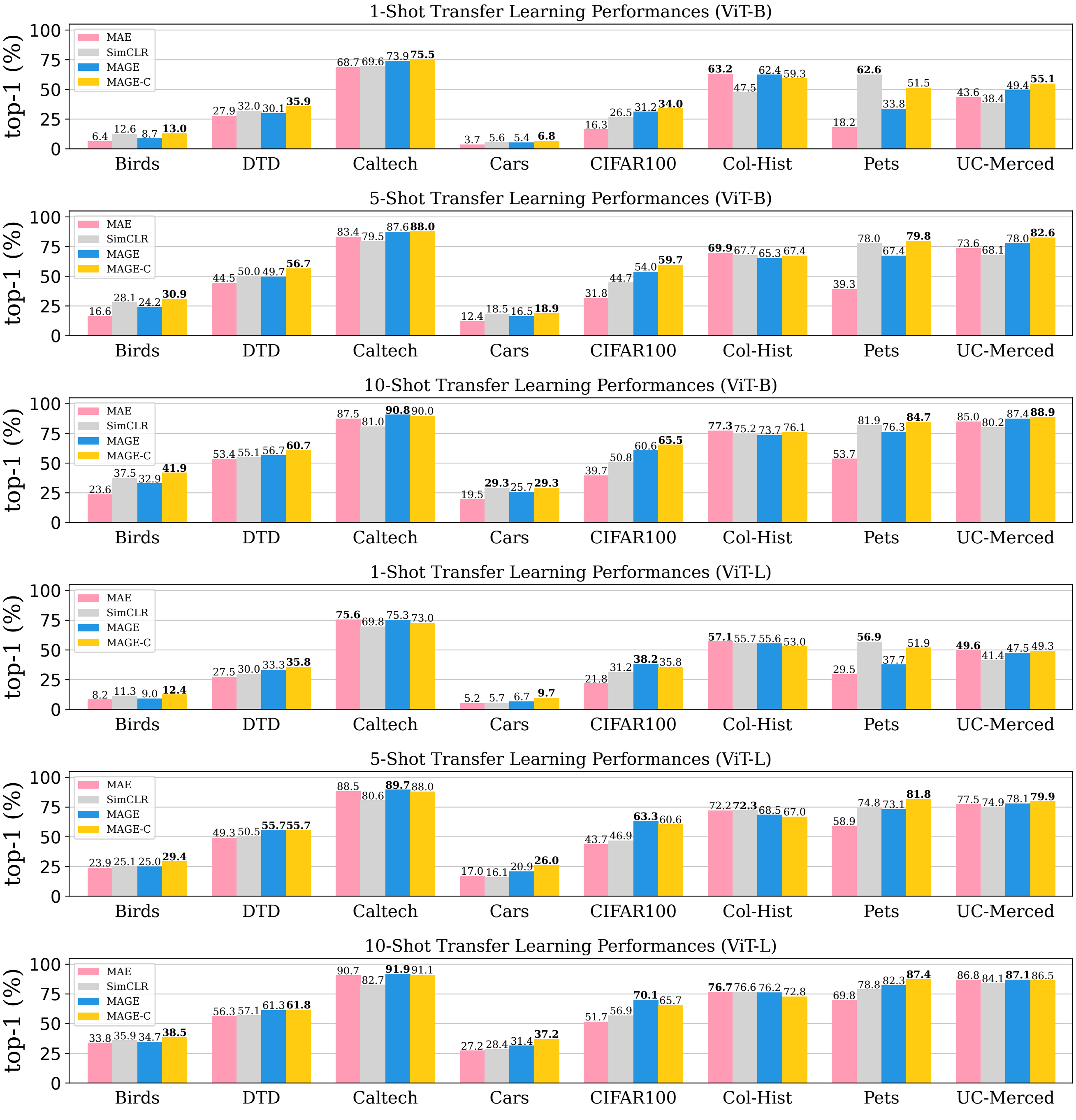


Figure 8. Transfer learning performance of ViT-B and ViT-L pre-trained on ImageNet-1K using different methods. Our methods outperform SimCLR [9] and MAE [26] on most datasets under different few-shot settings.

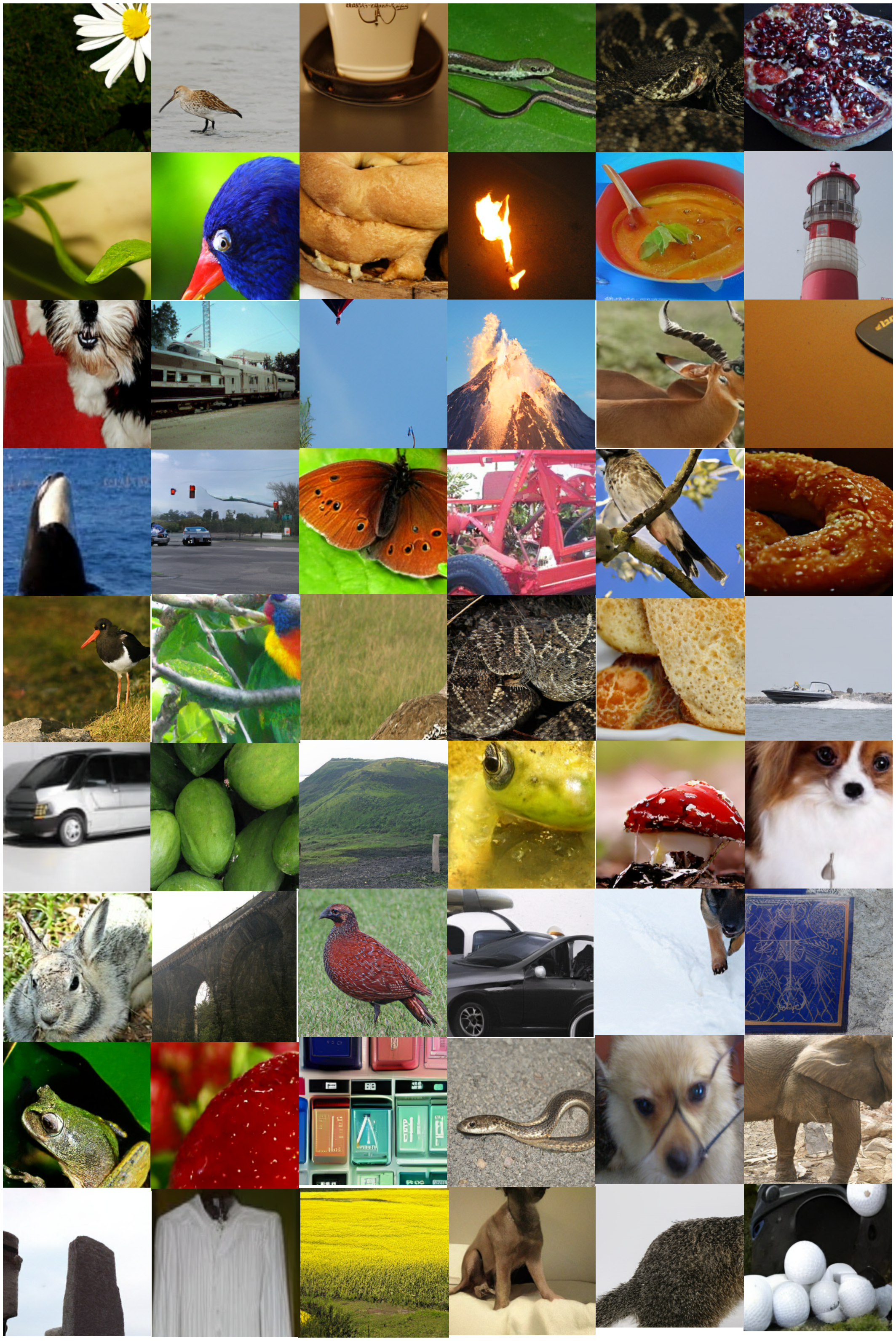


Figure 9. More uncurated examples of Class-unconditional image generation on ImageNet using MAGE trained with default strong augmentation.

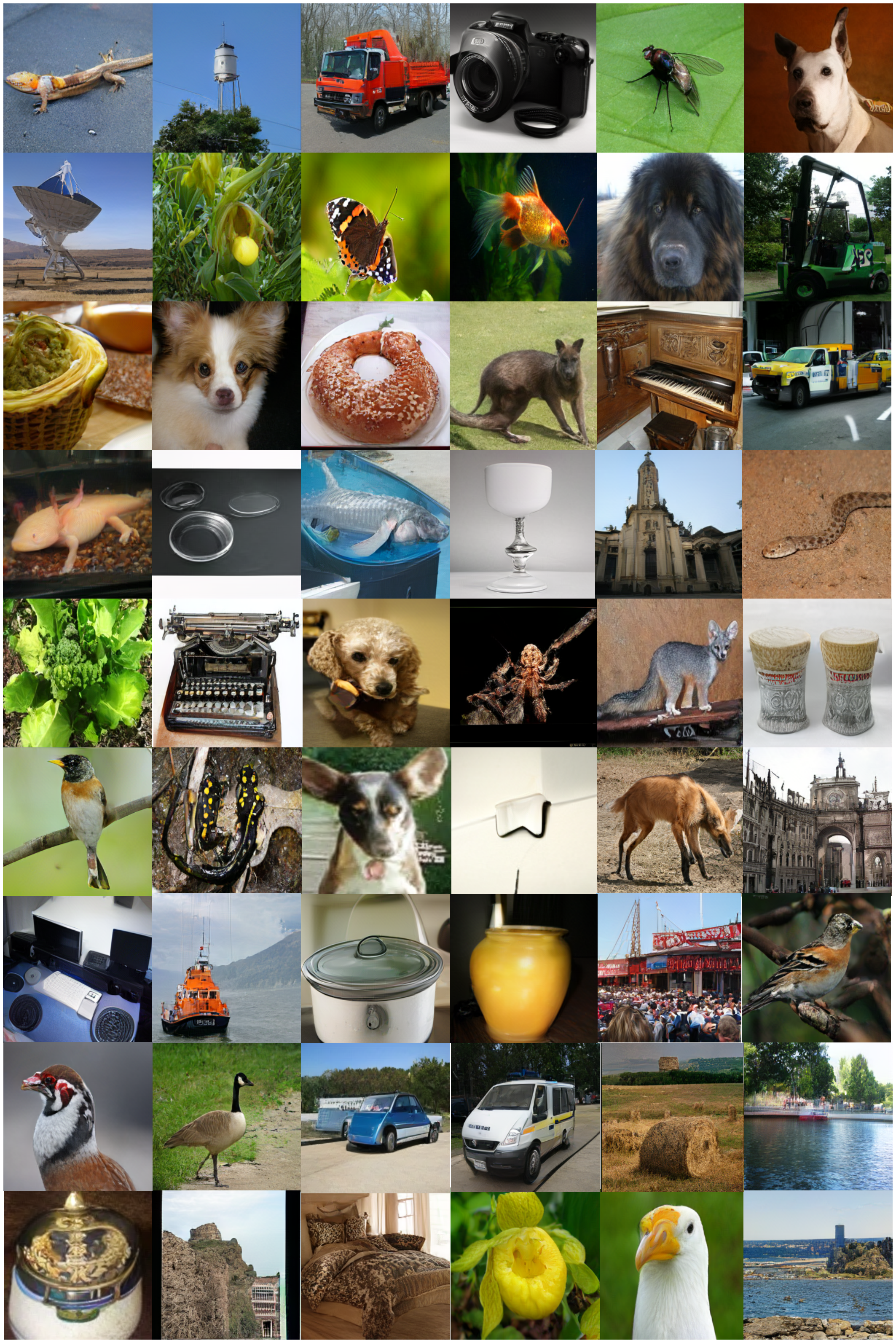


Figure 10. More uncurated examples of Class-unconditional image generation on ImageNet using MAGE trained with weak augmentation.

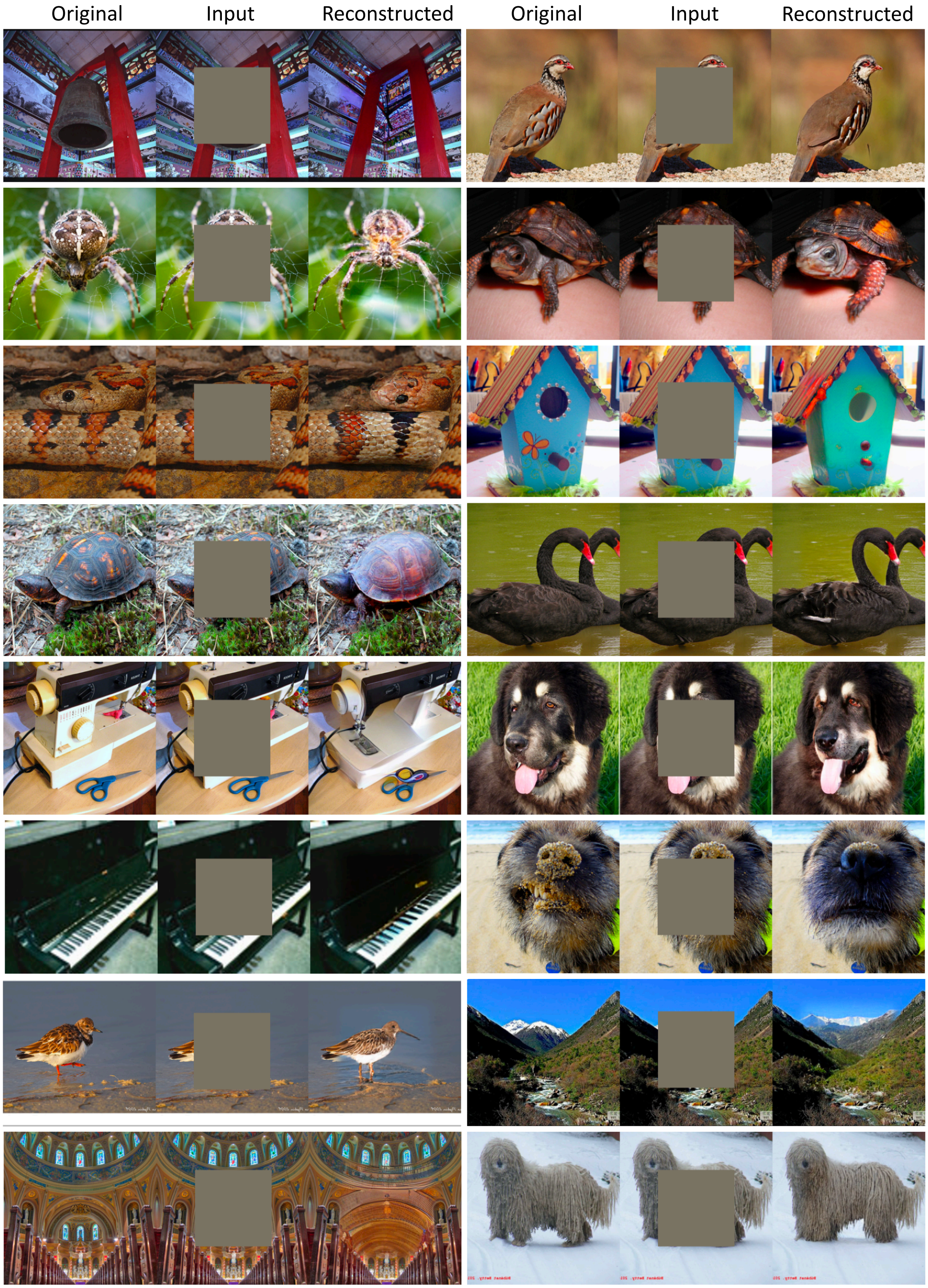


Figure 11. More examples of image inpainting using MAGE (ViT-L).

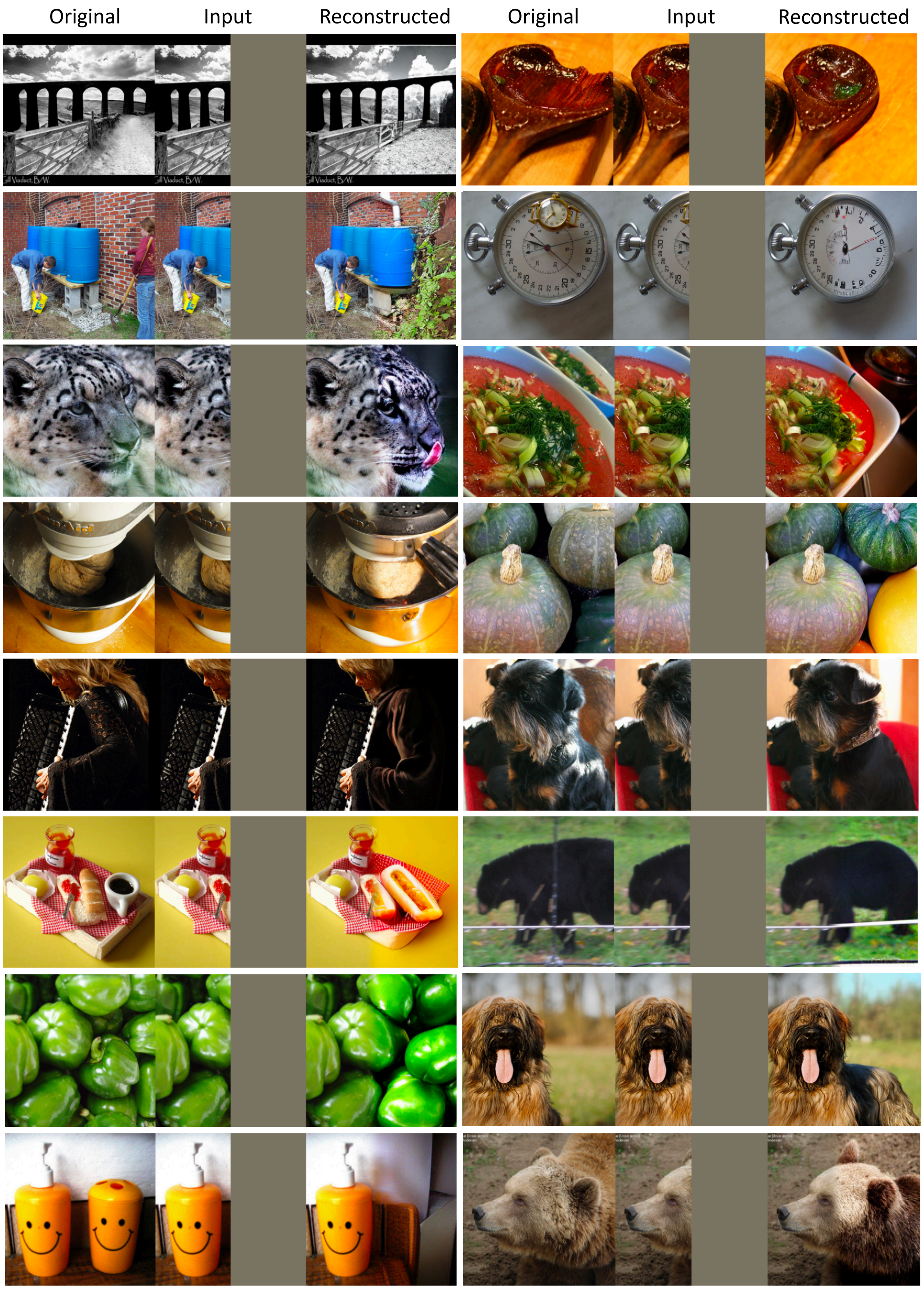


Figure 12. More examples of image outpainting using MAGE (ViT-L).

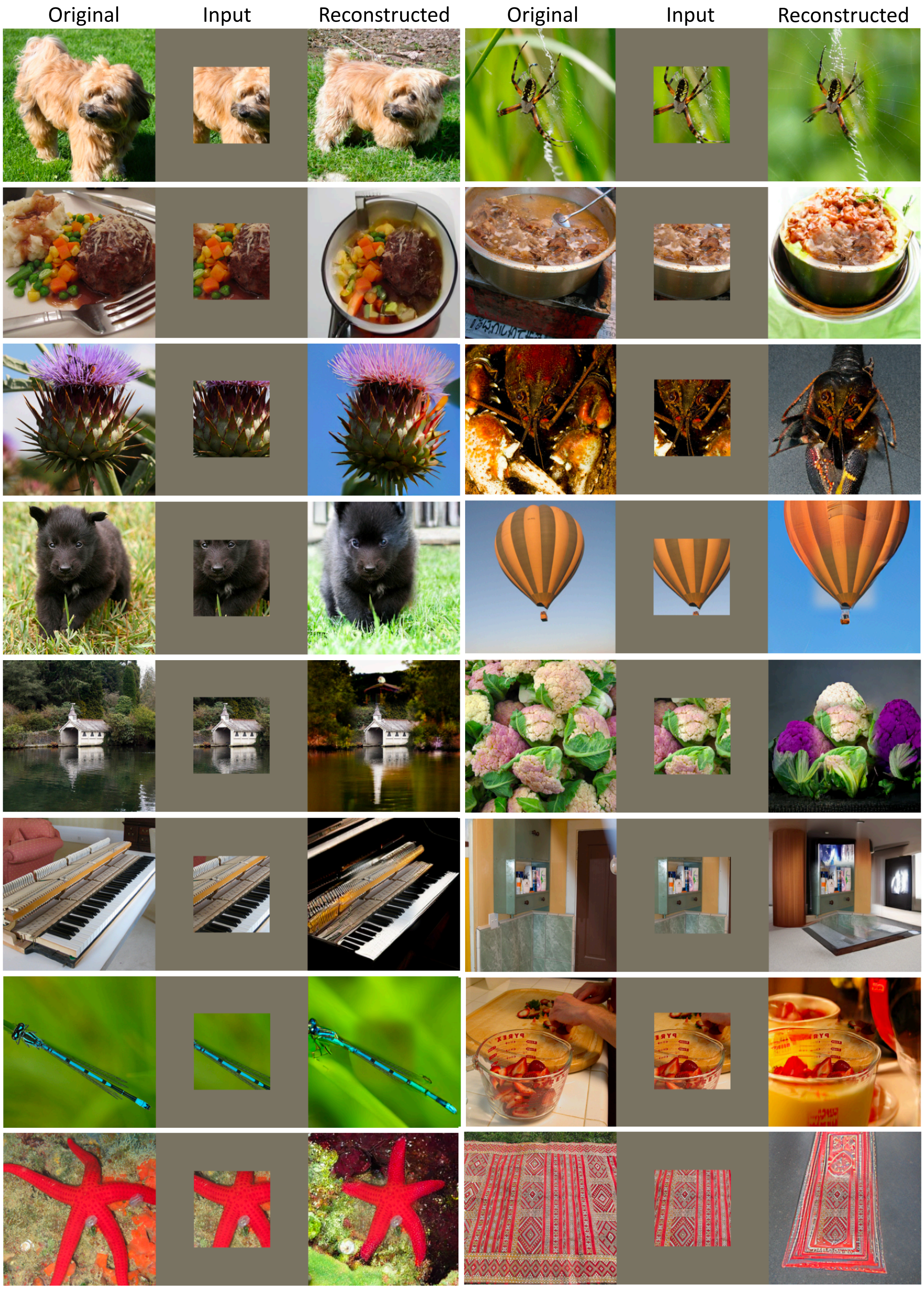


Figure 13. More examples of image outpainting on large outpainting mask (uncropping) using MAGE (ViT-L).