Exploring Stochastic Autoregressive Image Modeling for Visual Representation

Yu Qi[1*†], **Fan Yang**[2*], **Yousong Zhu**[3], **Yufei Liu**[1], **Liwei Wu**[2], **Rui Zhao**[2,4], **Wei Li**[2‡]

[1] Tsinghua University
[2] SenseTime Research
[3] Institute of Automation, Chinese Academy of Sciences
[4] Qing Yuan Research Institute, Shanghai Jiao Tong University, Shanghai, China
qiy20@mails.tsinghua.edu.cn, yousong.zhu@nlpr.ia.ac.cn, liuyufei@tsinghua.edu.cn
{yangfan1,wuliwei,zhaorui,liwei1}@sensetime.com

Abstract

Autoregressive language modeling (ALM) has been successfully used in self-supervised pre-training in Natural language processing (NLP). However, this paradigm has not achieved comparable results with other self-supervised approaches in computer vision (*e.g.*, contrastive learning, masked image modeling). In this paper, we try to find the reason why autoregressive modeling does not work well on vision tasks. To tackle this problem, we fully analyze the limitation of visual autoregressive methods and proposed a novel stochastic autoregressive image modeling (named SAIM) by the two simple designs. First, we serialize the image into patches. Second, we employ the stochastic permutation strategy to generate an effective and robust image context which is critical for vision tasks. To realize this task, we create a parallel encoder-decoder training process in which the encoder serves a similar role to the standard vision transformer focusing on learning the whole contextual information, and meanwhile the decoder predicts the content of the current position so that the encoder and decoder can reinforce each other. Our method significantly improves the performance of autoregressive image modeling and achieves the best accuracy (83.9%) on the vanilla ViT-Base model among methods using only ImageNet-1K data. Transfer performance in downstream tasks also shows that our model achieves competitive performance. Code is available at https://github.com/qiy20/SAIM.

Introduction

Un-/Self-supervised representation learning has achieved great success in natural language processing (NLP) (Yang et al. 2019; Brown et al. 2020; Devlin et al. 2018; Radford et al. 2018, 2019). Autoregressive language modeling (ALM) and masked language modeling (MLM) (*e.g.*, GPT (Radford et al. 2018, 2019; Brown et al. 2020) BERT (Devlin et al. 2018)), are capable of training large-scale language models with human-like performances using billions of unlabeled

training data. Motivated by the success of BERT and the outstanding performance of Vision Transformers (Dosovitskiy et al. 2020), recent works (He et al. 2021; Bao, Dong, and Wei 2021; Wei et al. 2021; Xie et al. 2021) introduce BERT-style pretraining by reconstructing the masked patches, which achieve an overall improvement in downstream tasks and greatly narrow the gap between vision and language. OpenAI made an attempt to propose an influential work called iGPT (Chen et al. 2020a), it learns representations by predicting pixels in raster order but the performances lag behind other self-supervised learning methods in both capacity and efficiency.

We ask: *what makes autoregressive modeling different between vision and language tasks?* We try to answer this question from the following aspects:

- **Input signal.** Different from language which follows the fixed natural order, images are not sequential signals. The lack of a well-defined order is the main challenge of applying autoregressive methods to process images. Previous methods, such as PixelCNN (Van Oord, Kalchbrenner, and Kavukcuoglu 2016) and iGPT, just follow the raster order to generate pixels. Such order is perhaps optimal for image generation, but it may not be the best order for visual representation learning. Because when people look at an image, they first focus on the main object or the object they are interested in, which is randomly distributed in any position instead of fixed in the top left corner. So we propose to learn representations by predicting the object in stochastic order, which can take advantage of the richness and variety of visual signals.

- **Architecture**. In vision, convolutional networks have been the mainstream model until recently, but in the field of NLP, Transformer was dominant. iGPT firstly trains a sequence Transformer to auto-regressively predict pixels. But it can only work with low-resolution images because the computational complexity of self-attention is quadratic to image size, which prevents the capabilities and scaling of such approaches. So we introduce Vision Transformer as the encoder to solve this problem. As for the decoder, we design the parallel encoder-decoder architecture which is inspired by XLNET (Yang et al. 2019) but doesn't share

---
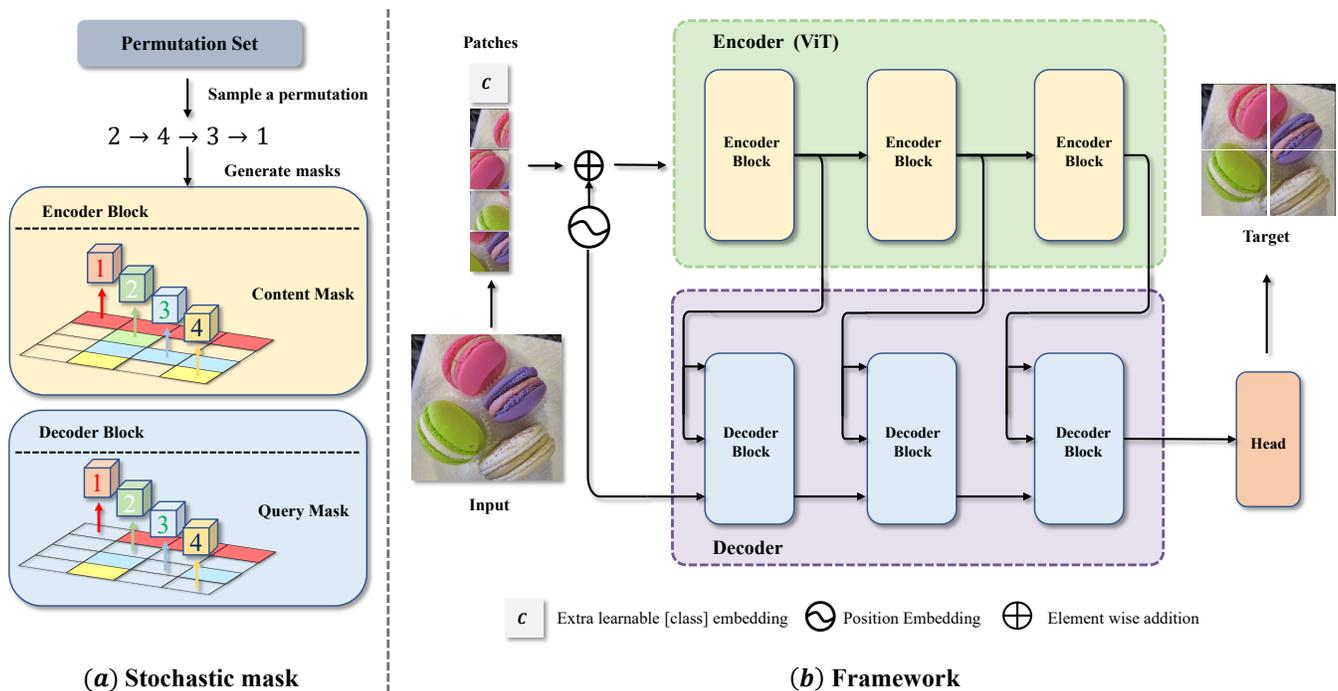
Figure 1: Training pipeline of Stochastic Autoregressive Pretraining. First, we randomly sample a permutation from the permutation set and generate two masks: content mask and query mask. Second, we employ a parallel encoder-decoder for autoregressive prediction. The encoder focus on learning contextual information(only the visible positions on the content mask), and the decoder reconstructs the original image from the latent representation along with position embeddings.

the weights. Benefit from our architecture, the encoder can focus on learning semantic representations without participating in pixel prediction.

- **Prediction target.** For autoregressive image modeling, the prediction target is raw pixels containing a lot of "noise". This is in contrast to language, where the model predicts words that are high-level concepts generated by humans. We think that the prediction of low-level signals will overfit the high-frequency details and textures, which are not helpful for high-level recognition tasks. To overcome this problem we employ Gaussian smoothing to discourage the model from learning high-frequency details and encourage learning semantic information.

Based on the findings and analysis, we propose the stochastic autoregressive image modeling (SAIM), as shown in Fig. 1. First, we generate two masks from a random permutation to achieve stochastic autoregression, the content mask and the query mask will apply them to the encoder and the decoder respectively. With the masks, each query can only hold the preceding information and the position embedding of itself. Second, we design the parallel encoder-decoder architecture. The encoder focus on learning contextual information, and the decoder reconstructs the original image from the latent representation along with position embeddings. Coupling these two designs enables us to realize stochastic sequence autoregression and achieve good performance. In addition, applying Gaussian smooth to target images also improves our performance. We pretrain the model on ImageNet-1K

and then fine-tune on three downstream tasks. For ViT-base, it achieves 83.9% top-1 fine-tuning accuracy on ImageNet-1K (Russakovsky et al. 2015), 49.4/43.9 box/mask mAP on COCO object detection (Lin et al. 2014) and 47.8 mIoU on ADE20K semantic segmentation (Zhou et al. 2019). Experimental results indicate that SAIM consistently improves performance and achieves competitive performance with the state-of-the-art methods, it proves that the simple *Autoregressive Image Modeling* is also an effective pretext task for visual representation learning.

Related Work

Autoregressive language modeling (ALM) and masked language modeling (MLM) are two main self-supervised learning approaches in the field of NLP. Given a sequence, BERT holds out a portion of the input sequence and tries to predict the missing tokens. GPT, on the hand, predicts all tokens following the left-to-right natural order of languages. This series of works have achieved great success in the field of natural language processing. To leverage the best of both BERT and GPT, the stochastic autoregressive language modelings *e.g.*, XLNet (Yang et al. 2019) enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. This strategy enables the ALM to acquire contextual information since the predicted targets can view all parts of the sequence during training. We mainly take inspiration from the permutation-based autoregressive pretraining.

Self-supervised learning (SSL) aims to learn a general visual representation from unlabeled data and have good transfer performance in downstream tasks. In the field of computer vision, researchers proposed a wide range of pretext tasks, such as image colorization (Zhang, Isola, and Efros 2016), jigsaw puzzle (Noroozi and Favaro 2016), image inpainting (Pathak et al. 2016), rotation prediction (Gidaris, Singh, and Komodakis 2018), and so on. Such self-supervised pre-training has limited success, but has seen significant interest in this field. Contrastive learning (He et al. 2020; Chen et al. 2020c; Caron et al. 2021; Chen, Xie, and He 2021; Chen et al. 2020b; Koch et al. 2015; Grill et al. 2020) have dominated self-supervised pre-training in recent years, which makes the representations of positive pairs similar while pushing negative pairs away. While, these methods need to learn representation from the object-centered datasets, and have poor transferability to dense prediction tasks (e.g. object detection and semantic segmentation).

Masked image modeling (MIM) introduces BERT-style pretraining into computer vision. ViT (Dosovitskiy et al. 2020) predicts the mean colors of masked patches. BEiT (Bao, Dong, and Wei 2021) encodes masked patches with discrete variational autoencoder (Ramesh et al. 2021) and uses the visual token as the prediction target. MAE (He et al. 2021) masks a high proportion of the input image and just predicts raw pixels. SimMIM (Xie et al. 2021) simply the pipeline of MIM methods. IBOT (Zhou et al. 2021) and data2vec (Baevski et al. 2022) perform masked prediction with an online tokenizer. These methods achieve better transfer performance than supervised learning and contrastive learning. We observe that on the NLP side, GPT models also have shown to be powerful, while researchers pay more attention to BERT-style pre-training in computer vision.

Autoregressive image modeling (AIM) is a classic approach in computer vision but located in a non-mainstream position for a long time. PixelCNN (Van den Oord et al. 2016) and VQ-VAE (Van Den Oord, Vinyals et al. 2017) model the distribution of natural images and generate new images based on pre-trained models. Pix2Seq (Chen et al. 2021) uses autoregressive modelling for object detection. For representation learning, CPC (Oord, Li, and Vinyals 2018) predicts patches by learning an autoregressive model in the latent space. The iGPT (Chen et al. 2020a) serializes pixels in raster order to make autoregressive predictions on low-resolution images. The computational complexity of self-attention limits the extension of this method. Recent advances in vision architectures, such as ViT, which serializes visual 2D data, provide an opportunity to apply similar large-scale pre-training in vision. Our work follows this line, attempting to promote the development of AIM and bridge the gap between vision and natural language processing.

Methods

Background

Autoregressive modeling try to learn a good visual representation by modeling the distribution of natural signals, which is a landmark problem in SSL (Van Oord, Kalchbrenner, and Kavukcuoglu 2016).

Specifically, given an unlabeled detaset $\mathcal{D}$ consisting of high dimensional data $\boldsymbol{x} = [x_1, x_2, \cdots, x_N]$, we can pick a permutation $\boldsymbol{z}$ of the set $[1, N]$, and we use z_i and $\boldsymbol{z}_{<i}$ to denote the i-th element and the first i-1 elements of the permutation. Autoregressive methods perform pretraining by maximizing the likelihood function:

$$\mathcal{L} = - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}} \sum_{i=1}^{N} \log p_\theta \left(x_{z_i} \mid \boldsymbol{x}_{z_{<i}} \right) \tag{1}$$

Where θ is the parameters of the autoregressive model. When working with images, (Chen et al. 2020a) just flatten the pixels in fixed raster order, e.g., x_i donates a pixel and $z_i = i$ for $1 \leq i \leq n$. But as we have analyzed before, there are two main limitations of this method: first, processing the pixel sequence is particularly time/space consuming (Chen et al. 2020a); second, predicting in raster-order is not consistent with the human visual mechanisms.

Stochastic Autoregressive Image Modeling

Our proposed SAIM first serializes the image into patches and then predicts patches in stochastic order, which overcomes the above limitations.

Image serialization. Following ViT (Dosovitskiy et al. 2020), we first split the 2D image $\boldsymbol{x} \in \mathcal{D}$ into patches, and the image patches are flattened into vectors $\{x_i\}_{i=1}^{N}$, where N is the number of patches. Then, the vectors are linearly projected to obtain patch embeddings $\boldsymbol{W} x_i \in \mathbb{R}^D$. Finally, We add 2D sin-cos position embeddings $\boldsymbol{E}_{pos} = [e_1, e_2, \cdots, e_N]$ to patch embeddings, where $\boldsymbol{E}_{pos} \in \mathbb{R}^{N \times D}$. So we get the initialized sequence $\boldsymbol{s} = [s_1, s_2, \cdots, s_N] = [\boldsymbol{W} x_1, \boldsymbol{W} x_2, \cdots, \boldsymbol{W} x_N] + \boldsymbol{E}_{pos}$.

Stochastic prediction. Instead of using a fixed permutation as in conventional autoregressive (AR) models. Our approach borrows ideas from XLNet (Yang et al. 2019), using all possible permutations as the prediction order. Specifically, for a sequence of length N, there are $N!$ different orders to perform an auto-regressive model. Let $\mathcal{Z}_N$ be the set of all possible permutations of the index set $\{1, 2, \cdots, N\}$. We predict the target patch depending on the preceding sequence $s_{\boldsymbol{z}_{<i}}$ and target position embedding e_{z_i} and train our model by minimizing the mean squared error between the reconstruction and original image pixels:

$$\mathcal{L} = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{Z}_N} \sum_{i=1}^{N} ||f_\theta(s_{\boldsymbol{z}_{<i}}, \; e_{z_i}) - x_{z_i}||^2 \tag{2}$$

where θ is the parameters of the model and $f_\theta(s_{\boldsymbol{z}_{<i}}, \; e_{z_i})$ is the output of the model.

Parallel Encoder and Decoder Architecture

Inspired by (Yang et al. 2019), we don't permute the input sequence directly but rely on the two-stream self-attention with mask to implement SAIM. Based on the analysis in introduction chapter, the design of decoder is important for our method. So we design a parallel encoder-decoder architecture, in which the encoder doesn't share weights with the decoder. In the pre-training stage, the encoder focuses on learning contextual information(only the visible positions on

the content mask), and the decoder reconstructs the original image from the latent representation along with position embeddings. In the finetune stage, only encoder will be reserved and no masks will be applied.

Mask generation. For a sequence of length N, we first randomly generate a vector $\boldsymbol{r} = [r_1, r_2, \cdots, r_N]$, where r_i follows a uniform distribution. So we can get the permutation $\boldsymbol{z} = \mathrm{argsort}(\boldsymbol{r})$. The **content mask** can be generated as follows:

$$\mathrm{content\_mask}_{ij} = \begin{cases} 0, & r_i < r_j \\ 1, & r_i \geq r_j \end{cases} \quad (3)$$

Where i, j are the coordinate of attention matrix, $\mathrm{content\_mask}_{ij} = 1$ represents the i-th token have access to the j-th token, $\mathrm{content\_mask}_{ij} = 0$ just represents the opposite. The **query mask** is basically the same shape as the content mask, except that the diagonal is all zero.

Encoder. The encoder has the same structure as the Vision Transformer, which consists of M layers of self-attention blocks, and we apply a **content mask** to the self-attention blocks, which makes the current token only gather information from the preceding positions. Computationally, we define $h_i^{(m)}$ as the output of the m-th encoder layer, where i is the token index. And we use the initialized sequence $\boldsymbol{s}$ as the input of the first encoder layer, i.e. $h_i^{(0)} = s_i$, the forward process of the encoder can be described as follows:

$$h_{z_t}^{(m)} = \mathrm{Attention}(\mathrm{Q} = h_{z_t}^{(m-1)}, \mathrm{KV} = h_{\boldsymbol{z}_{\leq t}}^{(m-1)}; \theta_e^{(m)})$$
$$where\ 1 \leq m \leq M \quad (4)$$

Where $\theta_e^{(m)}$ is the parameters of the m-th encoder layer. We omit the layer norm, MLP, and residual connection in the notation.

Decoder. The decoder consists of M layers of cross-attention blocks and a MLP layer. The cross-attention blocks reconstruct the original signals, and the MLP layer projects the signal to the initial dimension. We also apply a **query mask** to the cross-attention blocks.

We can define $g_i^{(m)}$ as the output of the m-th decoder layer. And we use the position embeddings $\boldsymbol{E}_{pos}$ as the input of the first decoder layer, i.e. $g_i^{(0)} = e_i$, the forward process of the decoder can be described as follows:

$$g_{z_t}^{(m)} = \mathrm{Attention}(\mathrm{Q} = g_{z_t}^{(m-1)}, \mathrm{KV} = h_{\boldsymbol{z}_{<t}}^{(m-1)}; \theta_d^{(m)})$$
$$where\ 1 \leq m \leq M \quad (5)$$

$$g_{z_t}^{(m)} = \mathrm{MLP}(g_{z_t}^{(m-1)}; \theta_d^{(m)})$$
$$where\ m = M + 1 \quad (6)$$

where $\theta_d^{(m)}$ is the parameters of the m-th decoder layer, which is different from $\theta_e^{(m)}$. Finally, we can use the output of the last decoder layer $g_{z_t}^{(M+1)}$ to compute loss, e.g. $f_\theta(s_{\boldsymbol{z}_{<i}}, e_{z_i}) = g_{z_t}^{(M+1)}$ in Eq. (2).

Gaussian Smoothing Application

Vision signals are raw and low-level, and high-frequency details and textures are not helpful for common recognition tasks. To make the model focus on learning semantic information, we construct a two-dimensional Gaussian filter kernel to reduce the texture detail of the target. The Eq. (2) can update as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \ \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_N} \ \sum_{i=1}^{N} ||f_\theta(s_{\mathbf{z}_{<i}}, e_{z_i}) - g_\xi(x_{z_i})||^2 \quad (7)$$

where g_ξ denotes the Gaussian filter. This simple strategy works well in our method and the computation is negligible.

Experiments

Pretraing Setup

Dataset and models. Our method is pretrained on the popular ImageNet-1k (Russakovsky et al. 2015) dataset. The dataset contains 1.28 million images from the training set of 1000 classes and 50,000 images from the validation set. We only use the training set during self-supervised learning. Our default model architecture has the parallel encoder and decoder. The encoder keeps the same structure as Vision Transformer (Dosovitskiy et al. 2020), and the decoder consists of cross-attention blocks and a MLP layer. Visual Transformer will load our trained encoder weights for downstream task evaluation.

Training configurations. We use AdamW for optimization and pretraining for 300/800 epochs with the batch size being 2048. We set the base learning rate as 2e-4, with cosine learning rate decay and a 30-epoch warmup, and set the weight decay as 0.05. We do not employ drop path and dropout. A light data augmentation strategy is used: random resize cropping with a scale range of [0.67, 1] and an aspect ratio range of [3/4, 4/3], followed by random flipping and color normalization steps.

Transfer Learning on Downstream Vision Tasks

Image classification on ImageNet-1K. We conduct fine-tuning and linear probing experiments on ImageNet-1K image classification in Tab. 1. The fine-tuning setting follows the common practice of supervised ViT (Dosovitskiy et al. 2020) training. Implementation details are in Supplemental. Our model pretrained with 300 epochs achieves the same accuracy as MAE (He et al. 2021) pretrained with 1600 epochs, which indicates that our method converges faster in the pretraining stage. Our hypothesis here is that our method can straightforwardly model dependencies between any two tokens and force the model to pay attention to every token, which is more efficient for representational learning. Under the longer training schedule (800 epochs), our model reaches 83.9% accuracy, 0.4% higher than MAE (He et al. 2021) and 0.9% higher than RandSAC (Hua et al. 2022) (a concurrent autoregressive work of ours).

Though we focus on learning representations that are better for fine-tuning, we also report the linear probing accuracy in Tab. 1. For linear probing, we use the feature of the last block and adopt an extra BatchNorm (Ioffe and Szegedy 2015) layer before the linear classifier following MAE.

Object detection and instance segmentation on COCO. We conduct object detection and instance segmentation experiments on the MS COCO dataset (Lin et al. 2014). We

Method	Arch	Pretrain epochs	Linear	Fine-tune
DeiT (Touvron et al. 2021)	ViT-B	300	-	81.8
MoCoV3 (Chen, Xie, and He 2021)	ViT-B	300	76.2	83.0
DINO (Caron et al. 2021)	ViT-B	400	77.3	83.3
BEiT (Bao, Dong, and Wei 2021)	ViT-B	800	-	83.2
MAE (He et al. 2021)	ViT-B	1600	67.8	83.6
simMIM (Xie et al. 2021)	ViT-B	1600	56.7	83.8
CAE (Chen et al. 2022)	ViT-B	800	68.3	83.6
iGPT* (Chen et al. 2020a)	iGPT-L	-	65.2	72.6
RandSAC (Hua et al. 2022)	ViT-B	1600	68.3	83.0
ViT-iGPT[†]	ViT-B	300	20.4	82.7
SAIM	ViT-B	300	58.5	83.6
SAIM	ViT-B	800	62.5	**83.9**

Table 1: Comparison with previous results on ImageNet-1K. ∗: iGPT-L contains 1.36 billion parameters, while others use ViT-base model. †: combine patch-wise modeling of ViT and raster ordering prediction of iGPT

Method	Pretrain data	Pretrain epochs	COCO-AP^{bb}	COCO-AP^{mk}	ADE20K-mIOU
supervised*	IN1K w/ labels	-	47.9	42.9	47.4
MoCoV3* (Chen, Xie, and He 2021)	IN1K	300	47.9	42.7	47.3
BEiT* (Bao, Dong, and Wei 2021)	IN1K+DALLE	1600	49.8	44.4	47.1
MAE* (He et al. 2021)	IN1K	1600	50.3	44.9	48.1
RandSAC (Hua et al. 2022)	IN1K	1600	-	-	47.3
ViT-iGPT	IN1K	300	45.5	41.2	42.0
SAIM	IN1K	300	47.4	42.8	46.1
SAIM	IN1K	800	49.4	44.0	47.8

Table 2: Comparison with previous on COCO and ADE20K. ∗: the result is taken from (Li et al. 2021), in which a grid-search was done to find the best hyperparameters for each method.

adopt ViT (Dosovitskiy et al. 2020) as the backbone of Mask-RCNN (He et al. 2017), following the architecture of ViT Benchmarking (Li et al. 2021). Implementation details are in Supplemental. In Tab. 2, we show the performance of representations learned through different self-supervised methods and supervised training. We report box AP for object detection and mask AP for instance segmentation. We observe that our method achieves 49.4 bbox mAP and 44.0 mask mAP. The result is better than supervised learning and contrastive learning but lags behind MIM methods. We note that BEIT and MAE are pretrained with 1600 epochs and use grid-search to find the best hyperparameters, while we only pretrain 800 epochs and don't tune any parameters in the fine-tune stage due to limited access to computation.

Semantic segmentation on ADE20K. We conduct semantic segmentation experiments on ADE20K dataset (Zhou et al. 2019). We adopt ViT (Dosovitskiy et al. 2020) as the backbone of UperNet (Xiao et al. 2018), following the implementation of BEiT (Bao, Dong, and Wei 2021). Implementation details are in Supplemental. In Tab. 2, we show the performance of our method. We report mIoU for semantic segmentation. We observe that our method achieves 47.8 mIoU, which is slightly lower than MAE by 0.3, but higher than all others.

In all tasks, our method surpasses the supervised results by large margins and achieves competitive performance with masked image modeling methods. Compared to the ViT-iGPT, our method improves 0.94%, on Imagenet-1k, 4.07 mIOU on ADE20K and 1.9/1.6 mAP on COCO. All these results indicate that our method can learn high-quality representations, and bridge the gap between MIM and AIM methods for unsupervised representation learning.

Ablation Study

Tokenization. We first compare two different tokenization strategies introduced by iGPT (Chen et al. 2020a) and ViT (Dosovitskiy et al. 2020), i.e., pixel-level modeling and patch-level modeling. ViT-style tokenization greatly improves the performance of the visual auto-aggressive method. The patch-based iGPT achieves 82.70 Top1 accuracy, which is higher than DeiT-base (Touvron et al. 2021)(81.8) but still lags behind the MIM method (e.g. 83.6 for MAE).

Prediction order. We compare the influence of different prediction orders. Our stochastic-order strategy (83.40) outperforms raster-order (82.70) 0.7% in accuracy, which indicates that the stochastic-order strategy plays a key role in our method. Our explanation here is that it leverages the richness and diversity of the visual signal and models full

Perform order		Decoder design		Gaussian smoothing		Tokenization		Fine-tune
raster	stochastic	shared weight	depth	kernel size	sigma	pixel	patch	
Ablation of tokenization:								
✓	-	-	0	-	-	✓		72.90*
✓	-	-	0	-	-		✓	**82.70**
Ablation of prediction order:								
✓		-	0	-	-		✓	82.70
✓		✗	12	-	-		✓	82.76
	✓	✗	12	-	-		✓	**83.40**
Ablation of decoder design:								
	✓	✗	12	-	-		✓	**83.40**
	✓	✓	12	-	-		✓	83.02
	✓	✗	6	-	-		✓	83.01
	✓	✗	1	-	-		✓	82.76
Ablation of Gaussian smoothing:								
	✓	✗	12	-	-		✓	83.40
	✓	✗	12	3	1		✓	83.52
	✓	✗	12	9	1		✓	**83.64**
	✓	✗	12	9	1.5		✓	83.45

Table 3: Ablation study on tokenization strategies, prediction order, decoder design, and Gaussian smoothing experiments by SAIM. ∗: the result is taken from (Chen et al. 2020a).

dependencies between all patches.

Decoder design. We first try to verify the effect of applying parallel architecture, in which the decoder doesn't share weights with the encoder. As is shown in the result, the parallel architecture gets a higher accuracy than the coupling architecture 0.4%. By this design, the encoder can focus on learning semantic representations without participating in image generation. We also vary the decoder depth to quantify its influence and find a sufficiently deep decoder performs better.

Gaussian kernel parameters. We ablate Gaussian kernel parameters in Tab. 3. We investigate the kernel size and standard deviation in Gaussian distribution. The larger the convolution kernel size or standard deviation is, the more texture will be removed from the target image. We obtain the best result when the kernel size is 9 and the standard deviation is 1, and all experiments with Gaussian smooth outperform those without.

Visualization and Analysis

Autoregressive image modeling attention maps. As shown in Fig. 2, we observed that the fixed-order autoregressive model cannot pay attention to the main object of the input image, and is only concerned with the local area of the image. While, SAIM with stochastic order focuses on the main information of the image, and obtains human-level attention representation with unlabeled data. In the field of computer vision, images are high-dimensional spatial information, and the main information of the input image is randomly distributed in tokens. SAIM allows each prediction token can

have the opportunity to see the global context information, which is crucial for autoregressive image modeling.

The reconstruction of autoregressive and autoencoder. As shown in Fig. 3, we discovered that MAE masked out 75% of the image tokens, resulting in blurred reconstruction results in large masked regions. According to the MAE, reducing the number of image masks can improve reconstruction results, but it will decrease model representation due to image local dependency. While, our proposed stochastic autoregressive image modeling, SAIM, utilizes all the information of the image to generate clear images, and achieve better fine-tuning accuracy than MAE on ImageNet-1K.

For example, the first quadruple in Fig. 3, shows that the baby in the image is ignored by MAE because it's totally masked, but SAIM can efficiently generate the baby in the image. Therefore, if the subject feature of the image is totally masked, the model will misunderstand the subject of the image. This problem will be harmful to downstream tasks. We argue that autoencoder methods introduce independence assumptions in image tasks, e.g., the masked subset of patches can't establish decencies with each other. The stochastic autoregressive image model enables the model to obtain the global dependency of the image by establishing the global context of the image.

Conclusion

In this paper, we fully analyze the differences between autoregressive modeling on visual and textual tasks and propose how to improve the performance of visual autoregressive models. Therefore, we explore a novel stochastic autore-
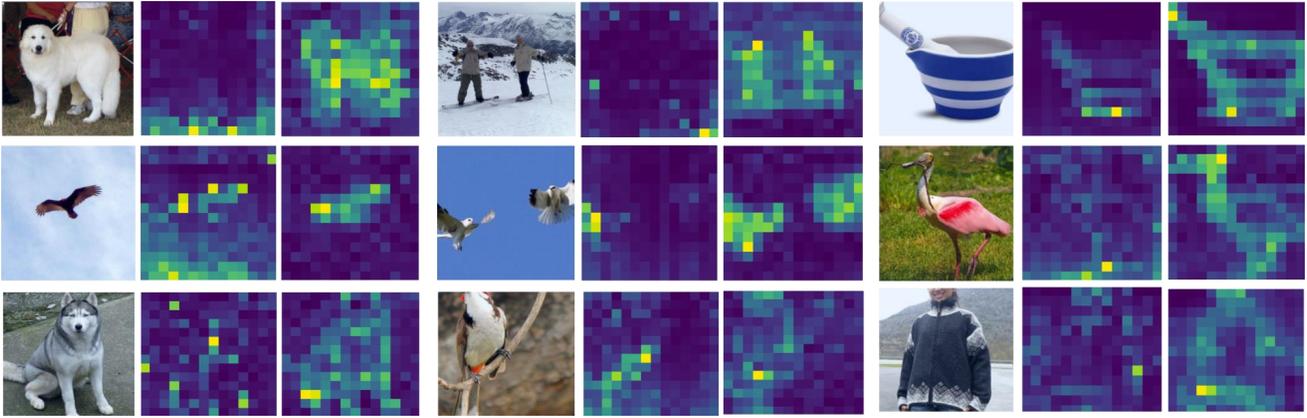
Figure 2: Attention maps of different prediction order. We show example results for ImageNet validation set. Description of images from left to right: (a) the original image, (b) the attention map of raster-order AIM, (c) the attention map of SAIM, more examples are in the appendix.

Figure 3: Reconstruction image of different methods. We show example results for ImageNet validation set. Description of images from left to right: (a) the origin image, (b) the mask map of the MAE, (c) the reconstruction image of MAE, (d) the reconstruction image of SAIM, more examples are in the appendix.

gressive image modeling (SAIM). By introducing ViT-style tokenization, stochastic order prediction, and the parallel encoder-decoder, we significantly improve the performance of autoregressive image modeling and bridge the gap between computer vision and NLP. Experiments on downstream tasks(image classification, object detection, and semantic segmentation) demonstrate the effectiveness of our method. Our method achieves competitive performance compared with other self-supervised methods, this proved that the autoregressive image modeling is also an effective pretext task for visual representation learning.

Discussion. The major drawback of this work is that our approach is time-consuming since our parallel architecture needs to calculate the self-attention blocks twice. Designing a lightweight decoder is challenging but worth exploring. We

will conduct future studies on this issue.

References

Baevski, A.; Hsu, W.-N.; Xu, Q.; Babu, A.; Gu, J.; and Auli, M. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*.

Bao, H.; Dong, L.; and Wei, F. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; and Sutskever, I. 2020a. Generative pretraining from pixels. In *International Conference on Machine Learning*, 1691–1703. PMLR.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. E. 2020c. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33: 22243–22255.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2021. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*.

Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; and Wang, J. 2022. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised visual transformers. *arXiv e-prints*, arXiv–2104.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

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

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33: 21271–21284.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2021. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Hua, T.; Tian, Y.; Ren, S.; Zhao, H.; and Sigal, L. 2022. Self-supervision through Random Segments with Autoregressive Coding (RandSAC). *arXiv preprint arXiv:2203.12054*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. PMLR.

Koch, G.; Zemel, R.; Salakhutdinov, R.; et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 0. Lille.

Li, Y.; Xie, S.; Chen, X.; Dollar, P.; He, K.; and Girshick, R. 2021. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, 69–84. Springer.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

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

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

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

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

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

Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; and Feichtenhofer, C. 2021. Masked Feature Prediction for Self-Supervised Visual Pre-Training. *arXiv preprint arXiv:2112.09133*.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 418–434.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2021. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *European conference on computer vision*, 649–666. Springer.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3): 302–321.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; and Kong, T. 2021. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*.