Rethinking Patch Dependence for Masked Autoencoders

Letian Fu[1*] Long Lian[1*] Renhao Wang[1] Baifeng Shi[1] Xudong Wang[1]
Adam Yala[1,2†] Trevor Darrell[1†] Alexei A. Efros[1†] Ken Goldberg[1†]
[1]UC Berkeley [2]UCSF

Abstract

In this work, we re-examine inter-patch dependencies in the decoding mechanism of masked autoencoders (MAE). We decompose this decoding mechanism for masked patch reconstruction in MAE into self-attention and cross-attention. Our investigations suggest that self-attention between mask patches is not essential for learning good representations. To this end, we propose a novel pretraining framework: Cross-Attention Masked Autoencoders (CrossMAE). CrossMAE's decoder leverages only cross-attention between masked and visible tokens, with no degradation in downstream performance. This design also enables decoding only a small subset of mask tokens, boosting efficiency. Furthermore, each decoder block can now leverage different encoder features, resulting in improved representation learning. CrossMAE matches MAE in performance with 2.5 to 3.7× less decoding compute. It also surpasses MAE on ImageNet classification and COCO instance segmentation under the same compute. Code and models: https://crossmae.github.io.

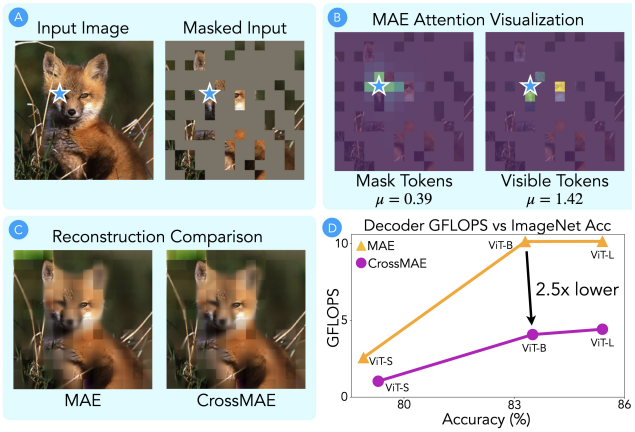

Figure 1. *Method Overview.* (A) Masked autoencoder (MAE) starts by masking random patches of the input image. (B) To reconstruct a mask token (i.e. marked by the blue star), MAE attends to both the masked tokens (B.Left) and the visible tokens (B.Right). A quantitative comparison over the ImageNet validation set shows that the masked tokens in MAE disproportionally attend to the visible tokens (1.42 vs 0.39), questioning the necessity of attention within mask tokens. (C) We propose CrossMAE, where each mask token can only attend to the visible tokens. (D) CrossMAE is equivalent to or better than MAE (Tabs. 1 and 3), with significantly less decoder FLOPS (2.5x lower).

1. Introduction

Masked image modeling [5, 31, 46, 60] has emerged as a pivotal unsupervised learning technique in computer vision. One such recent work following this paradigm is masked autoencoders (MAE): given only a small, random subset of visible image patches, the model is tasked to reconstruct the missing pixels. By operating strictly on this small subset of visible tokens, MAE can efficiently pre-train high-capacity models on large-scale vision datasets, demonstrating impressive results on a wide array of downstream tasks [34, 39, 49].

The MAE framework uses multi-headed *self-attention* throughout the model to perform the self-supervised reconstruction task, where the masked and visible tokens not only attend to each other but also to themselves, to generate a holistic and contextually aware representation. Yet, the mask tokens themselves do not contain information. Intuitively, allowing the exchange of information within nearby mask tokens is supposed to allow the model to synthesize a more consistent image; however, is dense self-attention necessary for learning good representation? We analyze the two parallel components involved in decoding each mask token: self-attention with other mask tokens, as well as cross-attention to the encoded visible tokens. If MAE relies on the self-attention with other mask tokens, its average should be on par with the cross-attention. Yet, the quantitative comparison in Figure 1.(b) shows the magnitude of mask token-to-visible token cross-attention (1.42) in the MAE decoder evaluated over the entire ImageNet validation set far exceeds that of mask token-to-mask token self-attention (0.39).

This initial observation prompts two questions: **1)** Is the self-attention mechanism within mask tokens in the decoder truly necessary for effective representation learning? **2)** If

*Equal contribution. †Equal advising.

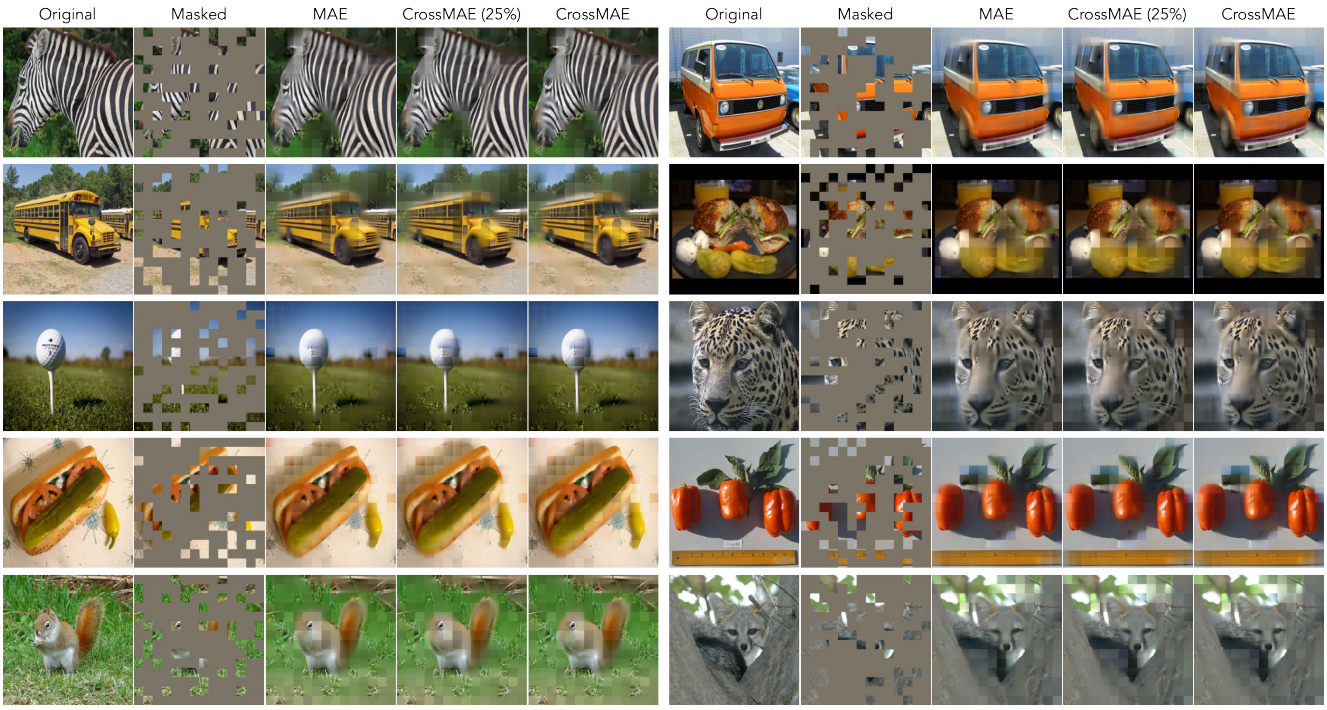

Figure 2. Example reconstructions of ImageNet *validation* images. For each set of 5 images, from left to right, are the original image, masked image with a mask ratio of 75%, MAE [31], CrossMAE (trained to reconstruct 25% of image tokens, or 1/3 of the mask tokens), and CrossMAE (trained to reconstruct all masked tokens). Since CrossMAE does not reconstruct them, all model outputs have the visible patches overlaid. Intriguingly, CrossMAE, when trained for partial reconstruction, can decode all mask tokens in one forward pass (shown above), which deviates from its training methodology. Its comparable reconstruction quality to full-image-trained models suggests that full-image reconstruction might not be essential for effective representation learning.

not, can each patch be *independently* generated, allowing the reconstruction of only a small subset of masked patches, which in turn, can facilitate faster pretraining without sacrificing downstream performance?

In addressing these questions, we introduce CrossMAE, which diverges from MAE in three ways:

1. **Cross-attention for decoding.** Rather than passing a concatenation of mask and visible tokens to a *self-attention* decoder, CrossMAE uses mask tokens to query the visible tokens in a *cross-attention decoder* to reconstruct the masked patches. In this setting, mask tokens incorporate information from the visible tokens but do not interact with other mask tokens, thereby reducing the sequence length for the decoder and cutting down computational costs.

2. **Partial reconstruction.** After the removal of self-attention, given the encoded features from the visible tokens, the decoding of each mask token becomes conditionally independent from one another. This enables the decoding of only a fraction of masked tokens rather than the entire image.

3. **Inter-block attention.** Due to the separation of visible and mask tokens, we are able to use features from different encoder blocks for each decoder block. Empirically, we find solely relying on the last encoder feature map for reconstruction, the design present in MAE, hurts feature learning. We propose a lightweight inter-block attention mechanism that allows the CrossMAE decoder to dynamically leverage a mix of low-level and high-level feature maps from the encoder, improving the learned representation.

Upon training the model, contrary to prior belief, we find that a ViT encoder pretrained with the reconstruction objective can learn a holistic representation of the image regardless of whether self-attention is used in the decoder. Visually, in Fig. 1.(c) and 2, the reconstruction results of CrossMAE are similar to MAE, although CrossMAE can only attend to the visible patches instead of having diffuse attention over all nearby patches. To our surprise, the downstream performance of CrossMAE is on par with MAE, while maintaining a higher efficiency enabled by cross-attention and partial reconstruction. We show that a ViT-B model trained with CrossMAE partial reconstruction achieves a classification accuracy of 83.5% on the ImageNet validation set, which surpasses its full-reconstruction MAE counterpart. In object detection and instance segmentation on COCO, CrossMAE achieves 52.1 AP and 46.3 AP, again surpassing MAE. Finally, we show that with ViT-L, CrossMAE enjoys improved scalability when compared to MAE.

2. Related Works

2.1. Self-Supervised Learning

In self-supervised representation learning (SSL), a model trains on a pretext task where the supervision comes from the input data itself without relying on labeled data. Contrastive learning is a popular approach to SSL that aims to learn representations by contrasting positive and negative samples, such as SimCLR [13], CPC [45], MoCo [30], MoCo v2 [14], BYOL [27], and DINO [10]. Additionally, group-instance contrastive learning works, such as DeepCluster [8], CLD [58] and SwAV [9], integrate clustering into contrastive learning to improve the quality of learned representations.

An alternative method for self-supervised learning is generative modeling, which focuses on acquiring a generative model capable of capturing the underlying data distribution. VAE/GAN [36] merges the strengths of variational autoencoders (VAEs) and generative adversarial networks (GANs) to acquire disentangled representations of data. PixelCNN, PixelVAE, and PixelTransformer [28, 54, 55] generate images pixel by pixel, taking into account the context of previously generated pixels. Masked modeling, a large subclass of generative modeling, is discussed in the following subsection. After the pre-training stage, these generative models can be finetuned for many downstream applications.

2.2. Masked Modeling

Masked modeling learns representations by reconstructing a masked portion of the input. Pioneering works in natural language processing (NLP) present various such pretraining objectives. BERT [20] and its extensions [35, 42] use a bidirectional transformer and present few-shot learning capabilities from masked language modeling. GPT [6, 47, 48], uses autoregressive, causal masking and demonstrates multi-task, few-shot, and in-context learning capabilities.

Early works in computer vision, such as Stacked Denoising Autoencoders [57] and Context Encoder [46], investigated masked image modeling as a form of denoising or representation learning. Recently, with the widespread use of transformer [21] as a backbone vision architecture, where images are patchified and tokenized as sequences, researchers are interested in how to transfer the success in language sequence modeling to scale vision transformers. BEiT [4], MAE [31], and SimMIM [60] are a few of the early works that explored BERT-style pretraining of vision transformers. Compared to works in NLP, both MAE and SimMIM [31, 60] find that a much higher mask ratio compared to works in NLP is necessary to learn good visual representation. Many recent works further extend masked pretraining to hierarchical architectures [41, 60] and study data the role of data augmentation [11, 22]. Many subsequent works present similar successes of masked pretraining for video [23, 29, 52], language-vision and multi-modal

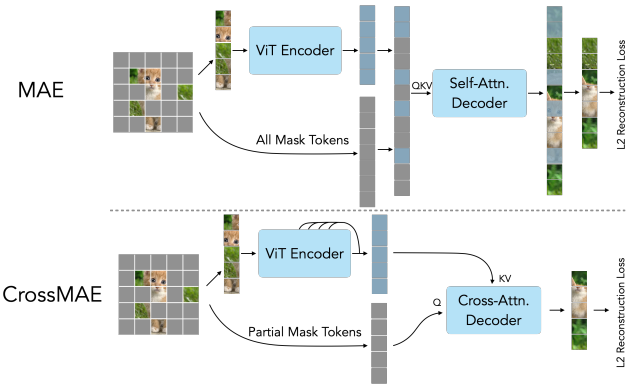

Figure 3. MAE [31] concatenates *all* mask tokens with the visible patch features from a ViT encoder and passes them to a decoder with self-attention blocks to reconstruct the original image. Patches that correspond to visible tokens are then dropped, and an L2 loss is applied to the rest of the reconstruction as the pretraining objective. CrossMAE instead uses cross-attention blocks in the decoder to reconstruct only a subset of the masked tokens.

pretraining [2, 24, 40] and for learning both good representations and reconstruction capabilities [38, 59].

However, BERT-style pretraining requires heavy use of self-attention, which makes computational complexity scale as a polynomial of sequence length. PixelTransformer [54] and DiffMAE [59] both use cross-attention for masked image generation and representation learning. Siamese MAE [29] uses an asymmetric masking pattern and decodes frames of a video condition on an earlier frame. In these settings, *all* masked patches are reconstructed. In this work, we investigate if learning good features necessitates high reconstruction quality and if the entire image needs to be reconstructed to facilitate representation learning. Similar in spirit, PCAE [37] progressively discards redundant mask tokens through its network, leading to a few tokens for reconstruction. In comparison, we minimally modify MAE and start decoding with a random subset of mask tokens.

2.3. Applications of Cross-Attention

In addition to the prevalent use of self-attention in computer vision, cross-attention has shown to be a cost-effective way to perform pooling from a large set of visible tokens. Intuitively, cross-attention can be seen as a parametric form of pooling, which learnably weighs different features. [53] replaces mean pooling with cross-attention pooling and finds improvement in ImageNet classification performance. [33] uses cross-attention to efficiently process large volumes of multi-modal data. Cross-attention is also widely used for object detection. [7] utilizes query tokens as placeholders for potential objects in the scene. [16, 17] further extend this concept by introducing additional query tokens to specifically tackle object segmentation in addition to the query tokens for object detection. In this work, we are interested

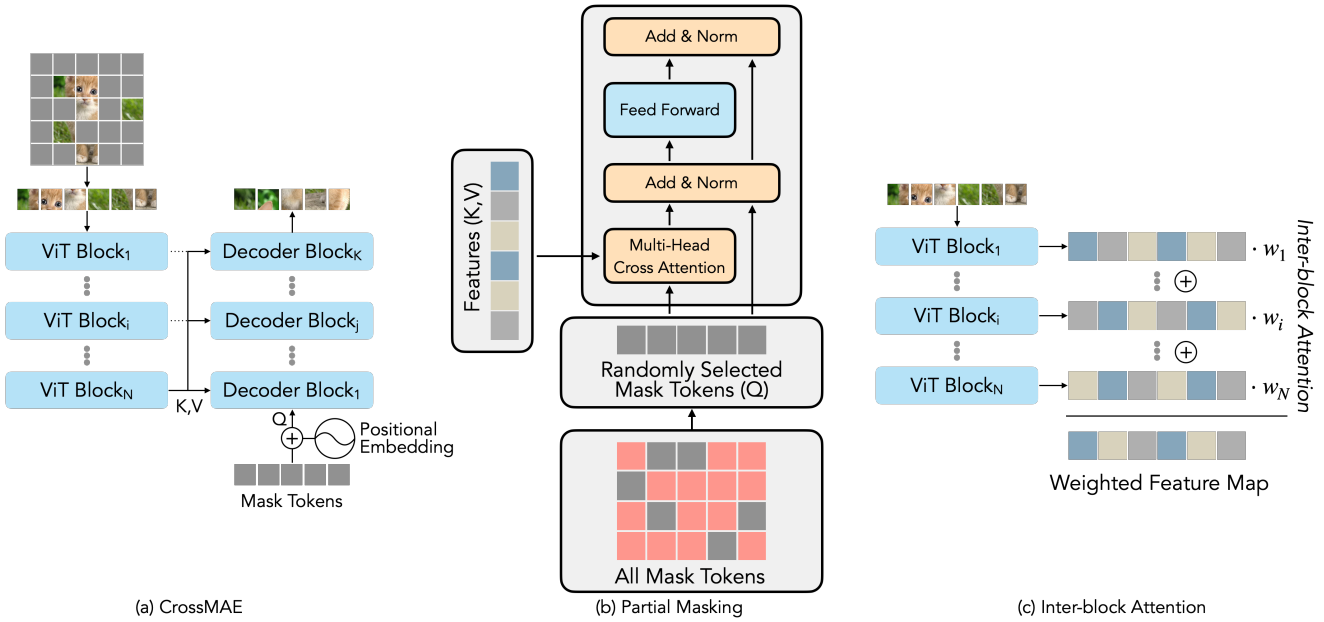

Figure 4. Overview of CrossMAE. (a) The vanilla version of CrossMAE uses the output of the last encoder block as the keys and queries for cross-attention. The first decoder block takes the sum of mask tokens and their corresponding positional embeddings as queries, and subsequent layers use the output of the previous decoder block as queries to reconstruct the masked patches. **(b)** Unlike the decoder block in [56], the cross-attention decoder block does not contain self-attention, decoupling the generation of different masked patches. **(c)** CrossMAE's decoder blocks can leverage low-level features for reconstruction via inter-block attention. It weighs the intermediate feature maps, and the weighted sum of feature maps is used as the key and value for each decoder block.

in cross-attention as an efficient method for self-supervised representation learning.

3. CrossMAE

This section is organized as follows. In Sec. 3.1, we first revisit vanilla Masked Autoencoders. In Sec. 3.2, we propose to use cross-attention instead of self-attention in the decoder for reconstruction. Thanks to a decoding architecture without self-attention, we achieve further efficiency gains by reconstructing only a subset of mask tokens for faster pretraining in Sec. 3.3. Since the use of cross-attention allows different features for different decoder blocks, in Sec. 3.4, we further propose inter-block attention to allow different decoder blocks to focus on different encoder features, which relieves the need to carry all information throughout the neural networks and allows for enhanced feature learning.

3.1. Preliminaries: Masked Autoencoders

Masked Autoencoders (MAE) [31] pretrain Vision Transformers (ViTs) [21]. Each image input is first patchified, and then a random subset of the patches is selected as the visible patches. As depicted in Fig. 3, the visible patches, concatenated with a learnable class token (<cls>), are subsequently fed into the ViT encoder, which outputs a set of feature latents. The latent vectors, concatenated with the sum of the positional embeddings of the masked patches and the learnable mask token, are passed into the MAE decoder. The decoder blocks share the same architecture as the encoder blocks (i.e., both are transformer blocks with self-attention layers). Note that the number of tokens fed into the decoder is the *same* length as the original input, and the decoding process assumes that the decoded tokens depend on both visible and masked tokens. Decoder outputs are then passed through a fully connected layer per patch for image reconstruction. After the reconstruction is generated, the loss is applied only to the masked positions, while the reconstructions for visible spatial locations are discarded.

Recall in Sec. 1, to study the properties of MAE, we measure the mean attention value across all attention maps over the ImageNet validation set. We group the attention values by cross-attention and self-attention between visible and masked tokens. We observe that in the decoding process of an MAE, mask tokens attend disproportionately to the class token and the visible tokens (see Figure 1.(b)). This motivates us to make design decisions and conduct experiments specifically to answer the following two questions:

1. Can good representations be learned if masked tokens can attend only to visible tokens?
2. Can we improve pretraining efficiency by reconstructing only *part* of an image?

3.2. Reconstruction with Cross-Attention

To address the first question, our method substitutes the self-attention mechanism in the decoder blocks with cross-attention. Specifically, the decoder employs multi-head cross-attention where the queries are the output from previous decoder blocks (or the sum of position embedding of the masked patches and mask token for the first decoder block.) The keys and values are derived from the encoded features.

In the most basic CrossMAE, the output from the final encoder block is used as the key and value tokens for all layers of the decoder, as illustrated in Fig. 4(a). Further exploration in Sec.3.4 reveals that utilizing a weighted mean of selected encoder feature maps can be beneficial. The residual connections in each decoder block enable iterative refinement of decoded tokens as they progress through decoder blocks.

Diverging from the original transformer architecture [56], our decoder omits the precursory causal self-attention stage before the introduction of multi-head cross-attention. This elimination, coupled with the fact that layer normalization and residual connections are only applied along the feature axis but not the token axis, enables the independent decoding of tokens. This design choice is evaluated in the ablation study section to determine its impact on performance.

Given the disparity in the dimensions of the encoder and decoder, MAE adapts the visible features to the decoder's latent space using a Multilayer Perceptron (MLP). However, in CrossMAE, as encoder features are integrated at various decoder blocks, we embed the projection within the multi-head cross-attention module.

This approach also mirrors aspects of Perceiver IO [33], where cross-attention is employed to amalgamate features pertinent to the task at hand. However, in contrast, we do not restrict the architecture to a single cross-attention unit. In our method, the masked tokens are decoded in a manner more akin to the traditional transformer paradigm [56].

3.3. Partial Reconstruction

The fact that CrossMAE uses cross-attention rather than self-attention in the decoder blocks brings an additional benefit over the original MAE architecture. Recall that mask tokens are decoded independently and thus there is no exchange of information between them, to obtain the reconstructions at a specific spatial location, CrossMAE only needs to pass the corresponding mask tokens to the cross-attention decoder. This allows efficient partial reconstruction in contrast to the original MAE architecture which needs to pass all the masked tokens as the input of the decoder blocks due to the existence of self-attention in the decoder blocks.

To address the second question in Sec. 3.1, rather than decoding the reconstruction for all masked locations, we only compute the reconstruction on a random subset of the locations and apply the loss to the decoded locations. Specifically, we name the ratio of predicted tokens to all image tokens as *prediction ratio* (γ), and the mask ratio (p). Then the prediction ratio is bounded between $\gamma \in (0, p]$. Because we are sampling within the masked tokens uniformly at random and the reconstruction loss is a mean square error on the reconstructed patches, the expected loss is the same as in MAE, while the variance is (p/γ) times larger than the variance in MAE. Empirically, we find that scaling the learning rate of MAE (β) to match the variance (i.e. setting the learning rate as $\gamma\beta/p$)) helps with model performance. Since cross-attention has linear complexity with respect to the number of masked tokens, this partial reconstruction paradigm leads to a decrease in computation complexity. Empirically, we find that the quality of the learned representations is not compromised by this approach.

3.4. Inter-block Attention

MAE combines the feature of the last encoder block with the mask tokens as the input to the self-attention decoder. This design poses the latent feature as an information bottleneck. Concretely, it leaves no room for any residual connection for the decoder to leverage early encoder features since the decoder blocks sequentially update and improve on the decoded features. In contrast, the cross-attention decoder in CrossMAE decouples queries from keys and values. This adds flexibility as different cross-attention blocks can take different feature maps as keys and values and thus no longer need to only use features from the last encoder block.

Besides simply using the feature from the last encoder block, one naive choice is to give the feature of the ith encoder block to the last ith decoder (*e.g.*, feed the feature of the first encoder to the last decoder), in a U-Net-like fashion. However, this assumes the depth of the decoder to be the same as the depth of the encoder, which does not hold in MAE where an asymmetric design is adopted.

Instead of manually selecting the feature for each decoder block, we follow our design choice of using cross-attention blocks for inter-patch spatial cross-attention and propose inter-block attention for feature fusion for each decoder block 4(c). Analogous to the inter-patch cross-attention that takes a weighted sum of the visible token embeddings across the patch dimensions to update the embeddings of masked tokens, inter-block attention takes a weighted sum of the visible token embeddings *across different input blocks* at the same spatial location to fuse the input features from multiple blocks into one feature map for each decoder block.

Concretely, each decoder block takes a weighted linear combination of encoder feature maps $\{f_i\}$ as keys and values. Specifically, for each key/value token t_k in decoder block k in a model with encoder depth n, we initialize a weight $w^k \in \mathcal{R}^n \sim \mathcal{N}(0, 1/n)$. Then t_k is defined as

$$t_k = \sum_{j=1}^{n} w_j^k f_j. \tag{1}$$

Method	Pre-train Data	ViT-S	ViT-B	ViT-L
Supervised [31, 50]	-	79.0	82.3	82.6
DINO [10]	IN1K	-	82.8	-
MoCo v3 [15]	IN1K	81.4	83.2	84.1
BEiT [4]	IN1K+DALLE	-	83.2	85.2
MultiMAE [3]	IN1K	-	83.3	-
MixedAE [11]	IN1K	-	83.5	-
CIM [22]	IN1K	**81.6**	83.3	-
MAE [31]	IN1K	78.9	83.3	**85.4**
CrossMAE (25%)	IN1K	79.2	83.5	**85.4**
CrossMAE (75%)	IN1K	79.3	**83.7**	**85.4**

Table 1. *ImageNet-1K classification accuracy*. CrossMAE performs on par, if not better than MAE without self-attention in the decoder. All experiments are run with 800 epochs. The best results are in **bold** while the second best results are underlined.

In addition to feature maps from different encoder blocks, we also include the inputs to the first encoder block to allow the decoder to leverage more low-level information to reconstruct the original image. We can select a subset of the feature maps from the encoder layers instead of all feature maps. This reduces the computation complexity of the system. We ablate over this design choice in Tab. 3d.

We show that using the weighted features rather than simply using the features from the last block greatly improves the performance of our CrossMAE. Interestingly, as we will show in Sec. 4.4, early decoder blocks focus on the information from the later encoder blocks, and later decoder blocks focus on the information from the early encoder blocks in the process of learning to achieve better reconstructions.

4. Experiments

We perform self-supervised pretraining on ImageNet-1K, following MAE [31]. We perform minimal hyperparameter tuning, maintaining consistency with MAE's parameters except for the learning rate and decoder depth. The hyperparameters were initially determined on ViT-Base and then directly applied to both ViT-Small and ViT-Large. Both CrossMAE and MAE are trained for 800 epochs. Please refer to the supplementary material for implementation details.

4.1. ImageNet Classification

Setup. The model performance is evaluated with end-to-end fine-tuning, with top-1 accuracy used for comparison. Same as in Figure. 2, we compare two versions of CrossMAE: one with a prediction ratio of 25% (1/3 of the mask tokens) and another with 75% (all mask tokens). Both models are trained with a mask ratio of 75% and a decoder depth of 12.

Results. As shown in Tab. 1, CrossMAE outperforms vanilla MAE trained on the same ViT-B encoder in terms of fine-tuning accuracy. This shows that replacing the self-attention with cross-attention *does not degrade* the quality of the pre-trained model in terms of downstream classification. CrossMAE outperforms other strong baselines such as DINO [10], MoCo v3 [15], BEiT [4], and MultiMAE [3].

		AP^{box}		AP^{mask}	
Method	Pre-train Data	ViT-B	ViT-L	ViT-B	ViT-L
Supervised [39]	IN1K w/ labels	47.6	49.6	42.4	43.8
MoCo v3 [15]	IN1K	47.9	49.3	42.7	44.0
BEiT [5]	IN1K+DALLE	49.8	53.3	44.4	47.1
MixedAE [11]	IN1K	50.3	-	43.5	-
MAE [39]	IN1K	51.2	54.6	45.5	48.6
CrossMAE	IN1K	**52.1**	**54.9**	**46.3**	**48.8**

Table 2. *COCO instance segmentation.* Compared to previous masked visual pretraining works, CrossMAE performs favorably on object detection and instance segmentation tasks.

4.2. Object Detection and Instance Segmentation

Setup. We additionally evaluate models pretrained with CrossMAE for object detection and instance segmentation, which require deeper spatial understanding than ImageNet classification. Specifically, we follow ViTDet [39], a method that leverages a Vision Transformer backbone for object detection and instance segmentation. We report box AP for object detection and mask AP for instance segmentation, following MAE [31]. We compare against 4 baselines: supervised pre-training, MoCo-v3 [15], BEiT [5], and MAE [31].

Results. As listed in Tab. 2, CrossMAE, with the default 75% prediction ratio, performs better compared to these strong baselines, including vanilla MAE. This suggests that similar to MAE, CrossMAE performance on ImageNet positively correlates with instance segmentation. Additionally, CrossMAE's downstream performance scales similarly to MAE as the model capacity increases from ViT-B to ViT-L.

4.3. Ablations

Cross-Attention vs Self-Attention. As summarized in Tab. 3a, CrossMAE that uses a cross-attention-only decoder has better downstream performance than vanilla MAE, which is consistent with the primary observations in Sec. 4.1. We further show that applying both cross-attention and self-attention together does not lead to additional benefits in terms of fine-tuning performance compared to using cross-attention only. This suggests that using only cross-attention is sufficient for learning good representations.

Mask Ratio and Prediction Ratio. We ablate over a range of mask ratio (*i.e.*, the ratio of mask tokens to all tokens) and the prediction ratio (*i.e.*, the ratio of reconstructed tokens to all tokens) in Tab. 3b and Tab. 3c. We observe that our method is not sensitive to varying masked tokens. Furthermore, although predicting the reconstructions from all the mask tokens leads to the best pre-trained model, the difference between full prediction and partial reconstruction is marginal. Specifically, CrossMAE can be trained to reconstruct as few as 15% of the tokens in the decoder rather than 100% of the tokens as required by the vanilla MAE baseline,

Method	Acc. (%)
MAE	83.0
CrossMAE	**83.3**
CrossMAE + Self-Attn	83.3

(a) Attention type in decoder blocks. Adding back self-attention between mask tokens does not improve performance.

Mask Ratio	Acc. (%)
65%	**83.5**
75%	83.3
85%	83.3

(b) Mask ratio. CrossMAE has consistent performance across high mask ratios.

Pred. Ratio	Acc. (%)
15%	83.1
25%	83.2
75%	**83.3**

(c) Prediction ratio. CrossMAE performs well even when only a fraction of mask tokens are reconstructed.

# Feature Maps Fused	Acc. (%)
1	82.9
3	83.3
6	**83.5**
12	83.3

(d) Inter-block attention. A combination of six select encoder feature maps is best.

Decoder Depth	Acc. (%)
1	83.0
4	83.1
8	83.1
12	**83.3**

(e) Decoder depth. CrossMAE performance scales with decoder depth.

Image Resolution	Acc. (%)
224	83.2
448	**84.6**

(f) Input resolution. CrossMAE scales to longer input sequences.

Table 3. *Ablations on CrossMAE*. We report fine-tuning performance on ImageNet-1K classification with 400 epochs (*i.e.*, half of the full experiments) with ViT-B/16. MAE performance is reproduced using the official MAE code. Underline indicates the default setting for CrossMAE. **Bold** indicates the best hyperparameter among the tested ones. 1 feature map fused (row 1, Table 2d) indicates using only the feature from the last encoder block. We use 25% prediction ratio for both settings in Table 2f to accelerate training.

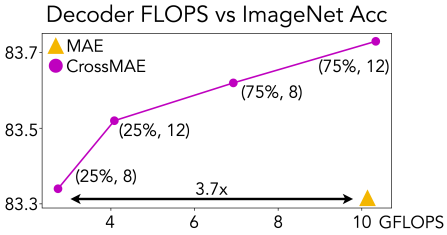

Figure 5. We compare ViT-B which is pre-trained for 800 epochs with different variants of CrossMAE v.s. MAE. For CrossMAE, we vary the prediction ratio p and number of decoder blocks n, and we denote each as (p, n). While all experiments are run with inter-block attention, CrossMAE has lower decoder FLOPS than MAE [31] and performs on par or better.

Method	Pred. Ratio	Decoder Depth	Memory	Runtime
MAE	0.75	8	OOM (>81920)	103.45
CrossMAE	0.25	12	41630	65.80

Table 4. Pretraining runtime and GPU memory comparison on ViT-B for 10 epochs using 2 NVIDIA A100 80GB GPUs. Memory is measured in MB per GPU, and runtime is in minutes. Please refer to Tab. 5 for a complete comparison. MAE trained without using gradient accumulation at the default batch size does not fit onto 2 GPUs, thus the memory usage is not reported.

yet achieving similar downstream finetuning performance. This result suggests that a good representation can be learned by reconstructing only part of an image.

Inter-block Attention. We also vary the number of encoder feature maps that are fused with our inter-block attention as an ablation. In addition to simply taking the feature from the last encoder block (*i.e.*, using only one feature map) and all encoder blocks (*i.e.*, using all 12 feature maps), we uniformly select feature maps to be fused in terms of their encoder block index. As shown in Tab. 3d, using only the last feature map leads to a minor degradation of performance compared to using all feature maps. Furthermore, adding even a subset of feature maps boosts the performance of CrossMAE, with the best performance reached when 6 feature maps are fused. This indicates that CrossMAE does not require all feature maps to obtain its optimal performance, which further justifies the efficiency of CrossMAE.

Decoder Depth. As shown in Tab. 3e, we show that using a decoder of 12 blocks slightly improves downstream performance compared to shallower decoders. CrossMAE performs on par with the vanilla MAE even with as few as one decoder block, which shows CrossMAE's capability for efficiently pre-train vision transformers. We further conduct full-scale experiments to compare the impact of decoder depth and prediction ratio, as they can both significantly impact the efficiency of the model. Results are summarized in Fig. 5 and the runtimes is compared in Tab. 4. We find that a model trained with a small prediction ratio can benefit more from a deeper decoder.

Input Resolution. We test CrossMAE on longer token lengths by increasing the image resolution without changing the patch size. As the resolution grows from 224 to 448, the image token length increases from 197 to 785, which poses great challenges to the scalability of existing methods. Thus, we deliberately choose the CrossMAE variant with a 25% prediction ratio for higher efficiency. In Tab. 3f, we observe that the classification accuracy positively correlates with the input resolution, suggesting that CrossMAE can scale to long input sequences.

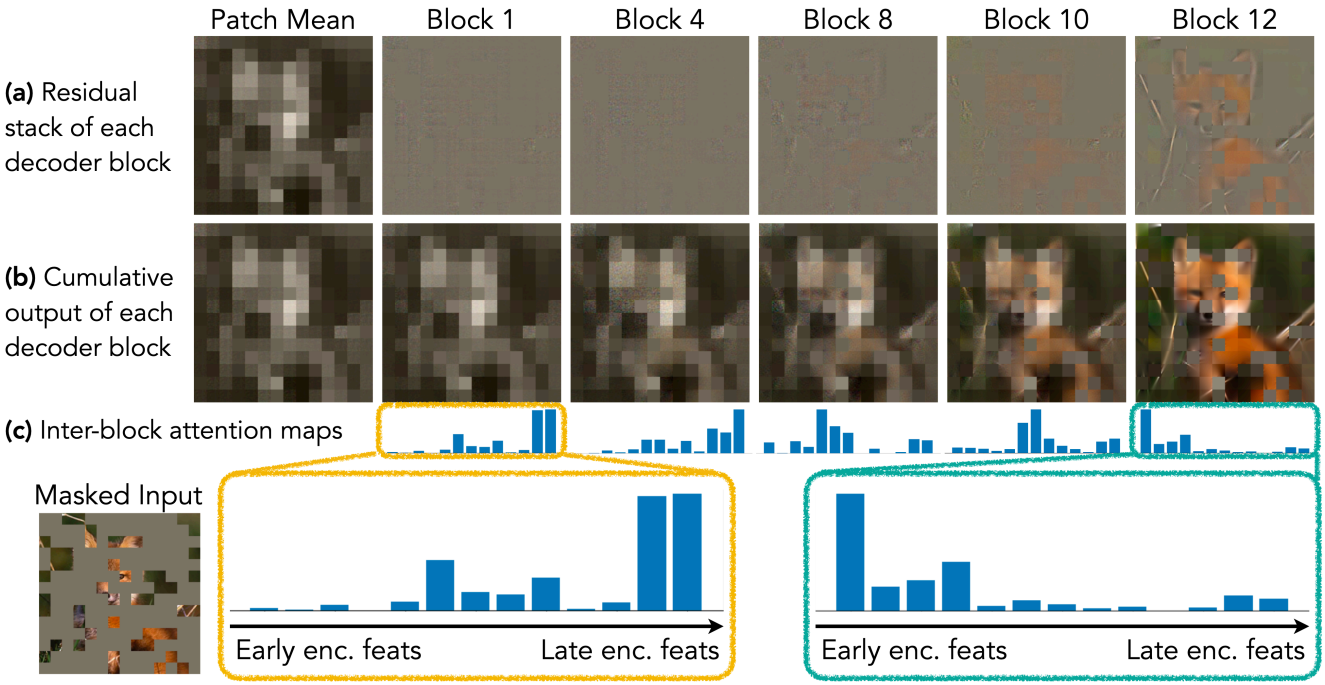

Figure 6. We visualize the output of each decoder block. (a-b) **Different decoder blocks play different roles in the reconstruction**, with most details emerging at later decoder blocks, which confirms the motivation for inter-block attention. (c) Visualizations of inter-block attention shows that **different decoder blocks indeed attend to feature from different encoder blocks**, with later blocks focusing on earlier encoder features to achieve reconstruction.

4.4. Visualizations

To further understand the properties of the updated reconstruction objective, we devise a method to visualize the reconstruction of CrossMAE's decoder blocks. Additionally, we visualize the weights applied to each encoder feature map to provide intuition for inter-block attention.

Visualizing Per-block Reconstruction. Rather than only visualizing the final reconstruction, we propose a more fine-grained visualization approach that allows us to precisely understand the effect and contribution of each decoder block.

Two key observations allow for such visualization. **1)** Transformer blocks have residual connections from their inputs to outputs. We denote f_i as the output of decoder i, $g_i(\cdot)$ as its residual path, with $f_i = f_{i-1} + g_i(f_{i-1})$. **2)** The output of the last decoder block is processed by the reconstruction head h to generate the reconstruction. Note that $h(\cdot)$ is linear because it is composed of two linear functions: a layer-norm [1] and a linear layer. Let D be the decoder depth, f_0 be the input to the first decoder block and y be the final reconstruction. We can recursively define $y = h(f_{D-1} + g_D(f_{D-1}))$. We expand it and by linearity of h we have:

$$\begin{aligned}\mathbf{y} &= h(f_0 + g_1(f_0) + \cdots + g_D(f_{D-1})) \\ &= \underbrace{h(f_0)}_{\text{Pos Embed. + Mask Token}} + \underbrace{h(g_1(f_0))}_{\text{Block 1}} + \cdots + \underbrace{h(g_D(f_{D-1}))}_{\text{Block D}}\end{aligned}$$

This decomposition allows us to express the reconstruction as an image stack, where the sum of all the levels gives us the final reconstruction. We present the visualization in Fig. 6 and analyze the contribution of each layer in the final reconstruction. We denormalize the output by patch mean and std. for visualization.

From Fig. 6 (a) and (b), we observe that different decoder blocks play different roles in reconstruction, with most details emerging at later decoder blocks. This supports the hypothesis for the need to get low-level information from early encoder blocks, motivating inter-block attention.

Visualizing Inter-block Attention Maps We visualize the attention maps of inter-block attention in 6(c). This shows that the CrossMAE model naturally leverages the inter-block attention to allow the later decoder blocks to focus on earlier encoder features to achieve reconstruction and allow the earlier decoder blocks to focus on later encoder features. This also motivates the need for different decoder blocks to attend to different encoder features, which is aligned with the performance gains obtained with inter-block attention.

5. Discussion and Conclusion

In this paper, we reassess the decoding mechanisms within MAE. Our exploration of MAE questions the necessity of using self-attention for reconstructing masked patches. To test this hypothesis, we design CrossMAE, an MAE-based

framework that **1)** uses cross-attention for reconstruction, **2)** decodes a fraction of the masked patches, and **3)** leverages different encoder features for reconstruction. CrossMAE shows similar performance and scaling properties as MAE while being more efficient. On one hand, CrossMAE opens the possibility to scale visual pretraining to longer contexts, especially in the setting of video pretraining, covering large swaths of in-the-wild visual data that have so far been computationally prohibitive to fully utilize. On the other hand, our investigations also give rise to worry: intuitively, self-attention among mask tokens in MAE should aid in consistent image reconstruction. However, as we show, the presence or absence of self-attention is almost uncorrelated with the quality of the learned representations in MAE. This may suggest that there exists a better way to leverage self-attention in masked visual pretraining. We hope that CrossMAE can serve as a starting point for the field to better explore the trade-off between self-attention and cross-attention for masked pretraining methods, potentially leading to truly scalable vision learners.

Acknowledgments. We thank Sophia Koepke, Yossi Gandelsman, and Qianqian Wang for their helpful discussions.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 8

[2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv:2204.01678*, 2022. 3

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision*, pages 348–367. Springer, 2022. 6

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3, 6

[5] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 1, 6

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. 2020. 3

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 3

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 3

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3, 6

[11] Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Mixed autoencoder for self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22742–22751, 2023. 3, 6

[12] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. 2020. 2

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[15] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 6

[16] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 3

[17] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 3

[18] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. arxiv e-prints, page. *arXiv preprint arXiv:1909.13719*, 4, 2019. 2

[19] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023. 2, 3

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019. 3

[21] 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 *ICLR*, 2020. 3, 4

[22] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 6

[23] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems*, 2022. 3

[24] Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurams, Sergey Levine, and Pieter Abbeel. Multimodal masked autoencoders learn transferable representations. *arXiv preprint arXiv:2205.14204*, 2022. 3

[25] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 2

[26] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[27] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 3

[28] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv preprint arXiv:1611.05013*, 2016. 3

[29] Agrim Gupta, Jiajun Wu, Jia Deng, and Li Fei-Fei. Siamese masked autoencoders. *arXiv preprint arXiv:2305.14344*, 2023. 3

[30] 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. 3

[31] 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 (CVPR)*, pages 16000–16009, 2022. 1, 2, 3, 4, 6, 7

[32] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016. 2

[33] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 3, 5

[34] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 1

[35] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. 3

[36] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016. 3

[37] Jin Li, Yaoming Wang, XIAOPENG ZHANG, Yabo Chen, Dongsheng Jiang, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Qi Tian. Progressively compressed auto-encoder for self-supervised representation learning. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[38] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. *arXiv preprint arXiv:2211.09117*, 2022. 3

[39] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. 1, 6, 2

[40] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023. 3

[41] Jihao Liu, Xin Huang, Jinliang Zheng, Yu Liu, and Hongsheng Li. Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. *arXiv:2205.13137*, 2022. 3

[42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[43] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. 2017. 2

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[46] 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. 1, 3

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 3

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2019. 3

[49] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 1

[50] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization

in vision transformers. *Transactions on Machine Learning Research*, 2022. 6

[51] 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. 2

[52] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 3

[53] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation, 2021. 3

[54] Shubham Tulsiani and Abhinav Gupta. Pixeltransformer: Sample conditioned signal generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10455–10464. PMLR, 2021. 3

[55] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 3

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 4, 5, 1

[57] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 3

[58] Xudong Wang, Ziwei Liu, and Stella X Yu. Unsupervised feature learning by cross-level instance-group discrimination. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12586–12595, 2021. 3

[59] Chen Wei, Karttikeya Mangalam, Po-Yao Huang, Yanghao Li, Haoqi Fan, Hu Xu, Huiyu Wang, Cihang Xie, Alan Yuille, and Christoph Feichtenhofer. Diffusion models as masked autoencoder. In *ICCV*, 2023. 3

[60] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9653–9663, 2022. 1, 3

[61] 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. 2

[62] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2

Rethinking Patch Dependence for Masked Autoencoders

Supplementary Material

1. Implementation details

1.1. Attention Calculation

To compare the attention values for mask tokens in vanilla MAE (Fig. 1), we trained a ViT-B/16 MAE for 800 epochs using the default hyperparameters provided in [31]. For each image, we randomly generate a 75% binary mask (m) for all tokens, with $m_i = 1$ representing a token being masked and $m_i = 0$ otherwise. During the forward pass of the decoder, for each self-attention operation, the attention map is stored. This means that for the default MAE, a total of 8 attention maps, each with 16 attention heads are stored. Based on the mask pattern, we calculate the outer product ($m \cdot m^\top$) for the self-attention among mask tokens, and $m \cdot (1 - m^\top)$ for the cross-attention from the mask token to the visible tokens. We then calculate the average across all feature maps and attention heads for self-attention and cross-attention to get the image average values. Lastly, we averaged across the entire ImageNet validation set to obtain the final values.

1.2. Inter-Block Attention

We tried a few implementations for inter-block attention and found the following implementation to be the fastest and most memory-efficient. In this implementation, we combine inter-block attention for all encoder layers as a single forward pass of a linear layer. For each decoder block, we index into the output tensor to extract the corresponding feature map, and a layer norm will be applied before the feature map is fed into the decoder block. Other alternatives we tried include 1) performing separate inter-block attentions before each decoder block, and 2) 1x1 convolution on the stacked encoder feature maps.

In MAE, there exists a layer norm after the last encoder feature map before feeding into the decoder. In our implementation, we only add layer norm after inter-block attention. We find that adding an additional layer norm before inter-block attention to each encoder feature map does not lead to improvements in model performance but will significantly increase GPU memory usage.

The pseudo-code of inter-block attention is the following:

```
class InterBlockAttention():
    def __init__(self, num_feat_maps, decoder_depth):
        self.linear = Linear(num_feat_maps, decoder_depth, bias=False)
        std_dev = 1. / sqrt(num_feat_maps)
        init.normal_(self.linear.weight, mean=0., std=std_dev)

    def forward(self, feature_maps : list):
        """
        feature_maps: a list of length num_feat_maps, each with dimension
        Batch Size x Num. Tokens x Embedding Dim.
        """
        stacked_feature_maps = stack(feature_maps, dim=-1)
        return self.linear(stacked_feature_maps)
```

1.3. Ablation that Adds Self-Attention

In Section 4.3 (a), we propose adding self-attention back to CrossMAE as an ablation. In that particular ablation study, we analyze the effect of self-attention between the masked tokens, which can be used to improve the consistency for reconstruction. Specifically, we modify the formulation in the original transformer paper [56], where the mask/query tokens are first passed through a multi-head self-attention and a residual connection before being used in the multi-headed cross-attention with the features from the encoder. The primary difference with the vanilla transformer decoder implementation [56] is we do not perform casual masking in the multi-head self-attention. Please reference Fig. 7 for a more visual presentation of the method.

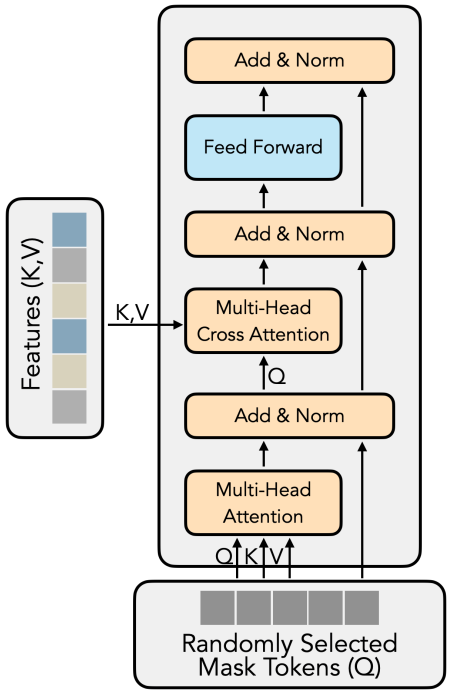

Figure 7. Modification for self-attention ablation

Method	Mask Ratio	Prediction Ratio	Decoder Depth	Interblock Attention	FlashAttn-2 [19]	Memory↓	Runtime (Minutes)↓
MAE	0.75	0.75	8	-	x	-	103.45
MAE	0.75	0.75	8	-	✓	53858	74.80
MAE	0.75	0.75	12	-	✓	68386	93.48
CrossMAE	0.75	0.75	8	✓	✓	46930	69.33
CrossMAE	0.75	0.75	12	x	✓	55358	73.75
CrossMAE	0.75	0.75	12	✓	✓	57987	84.05
CrossMAE	0.75	0.25	8	✓	✓	32055	54.88
CrossMAE	0.75	0.25	12	x	✓	34191	52.45
CrossMAE	0.75	0.25	12	✓	x	41630	65.80
CrossMAE	0.75	0.25	12	✓	✓	36805	63.23

Table 5. Pretraining runtime and GPU memory under different configurations. All time trials are conducted for 10 epochs (instead of the full 800 epochs) using 2 NVIDIA A100 GPUs. Memory is measured in MB per GPU, and runtime is measured in minutes. Note that the memory measured here contains both the encoder and decoder.

1.4. Ablation on Inter-block Attention

In Table 3d, the following cases are considered. 1 feature map (row 1) does not use inter-block attention. Each decoder block only takes the last feature map from the encoder as the keys and values. For scenarios where more than one feature map is used, the output of the patch embedding (input to the ViT) is also used.

1.5. Hyperparameters

Pre-training: The default setting is in Table 6, which is consistent with the official MAE [31] implementation. As mentioned in Sec. 3.4, we scale the learning rate by the ratio between mask ratio (p) and prediction ratio (γ) to ensure the variance of the loss is consistent with [31]. Additionally, we use the linear learning rate scaling rule [26]. This results in $lr = \gamma * base\_lr * batchsize/(256 * p)$. For Table 1, we use 12 decoder blocks, with mask ratio and prediction ratio both 75%, and interblock attention takes in all encoder feature maps. For the 400 epochs experiments in Table 2, we scale the warm-up epochs correspondingly. Other hyperparameters are the same as MAE.

Finetuning: We use the same hyperparameters as MAE finetuning. We use global average pooling for finetuning. In MAE, the layer norm for the last encoder feature map is removed for finetuning, which is consistent with our pretraining setup. Please refer to Table 7 for more detail.

1.6. Compute Infrastructure

Each of the pretraining and finetuning experiments is run on 2 or 4 NVIDIA A100 80GB GPUs. The batch size per GPU is scaled accordingly and we use gradient accumulation to avoid out-of-memory errors. ViTDet [39] experiments use a single machine equipped with 8 NVIDIA A100 (80GB) GPUs. We copy the datasets to the shared memory on the machines to accelerate dataloading. We use FlashAttention-2 [19] to accelerate attention calculation.

Config	Value
optimizer	AdamW [44]
base learning rate	1.5e-4
learning rate schedule	cosine decay [43]
batch size	4096
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.95 [12]
warm up epoch [25]	20, 40
total epochs	400, 800
augmentation	RandomResizedCrop, RandomHorizontalFlip

Table 6. Pretraining Hyperparameters

Config	Value
optimizer	AdamW
base learning rate	1e-3
learning rate schedule	cosine decay
batch size	1024
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.999
warm up epoch	5
total epochs	100 (B), 50 (L)
augmentation	RandAug (9, 0.5) [18]
label smoothing [51]	0.1
mixup [62]	0.8
cutmix [61]	1.0
drop path [32]	0.1

Table 7. Finetuning Hyperparameters

1.7. Runtime and GPU Memory

In this section, we provide quantitative comparisons of the effect of mask ratios, prediction ratios, and interblock attention on GPU memory usage (for both the encoder and decoder) and runtime. We provide runtimes (in minutes) of different settings of MAE and CrossMAE for 10 epochs with

their corresponding GPU memory usage (in MB). All experiments here are conducted with 2 NVIDIA A100 (80GB) GPUs, with the standard hyperparameters provided above for pretraining. The results are listed in Tab. 5. Additionally, to compare against the vanilla MAE implementation, we provide rows where FlashAttention-2[19] is not enabled. Note that for MAE, the model does not fit on 2 A100 when FlashAttention-2 is not enabled, yet by adjusting the prediction ratio of CrossMAE, the model fits within the memory limit.