Rethinking Patch Dependence for Masked Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we examine the impact of inter-patch dependencies in the decoder of masked autoencoders (MAE) on representation learning. We decompose the decoding mechanism for masked reconstruction into self-attention between mask tokens and cross-attention between masked and visible tokens. Our findings reveal that MAE reconstructs coherent images from visible patches not through interactions between patches in the decoder but by learning a global representation within the encoder. This discovery leads us to propose a simple visual pretraining framework: cross-attention masked autoencoders (CrossMAE). This framework employs only cross-attention in the decoder to independently read out reconstructions for a small subset of masked patches from encoder outputs, yet it achieves comparable or superior performance to traditional MAE across models ranging from ViT-S to ViT-H. By its design, CrossMAE challenges the necessity of interaction between mask tokens for effective masked pretraining. Code is available here.

1 Introduction

Masked image modeling [46, 30, 61, 4] 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 mostly 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 [33, 38, 49].

The MAE framework employs *self-attention* across the entire model for self-supervised reconstruction tasks. In this setup, both masked and visible tokens engage in self-attention, not just with each other but also with themselves, aiming to generate a holistic and context-aware representation. However, the masked tokens inherently lack information. Intuitively, facilitating information exchange among adjacent masked tokens should enable the model to synthesize a more coherent image, thereby accomplishing the task of masked reconstruction and improving representation learning. A question arises, though: Is this truly the case?

We decompose the decoding process of each mask token into two parallel components: 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 among mask tokens in the decoder necessary for effective representation learning? **2)** If not, can each patch be

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

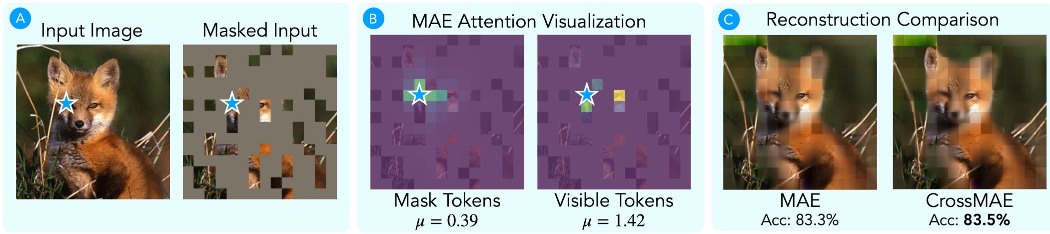


Figure 1: *Method Overview*. **(A)** Masked autoencoder (MAE) starts by masking random patches of the input image. **(B)** To reconstruct a mask token (marked by the blue star), MAE attends to both the masked tokens (B.Left) and the visible tokens (B.Right). A quantitative comparison over the ImageNet 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, the masked patches are reconstructed from only the cross attention between the masked tokens and the visible tokens. Surprisingly, CrossMAE attains the same or better performance than MAE on ImageNet classification and COCO instance segmentation.

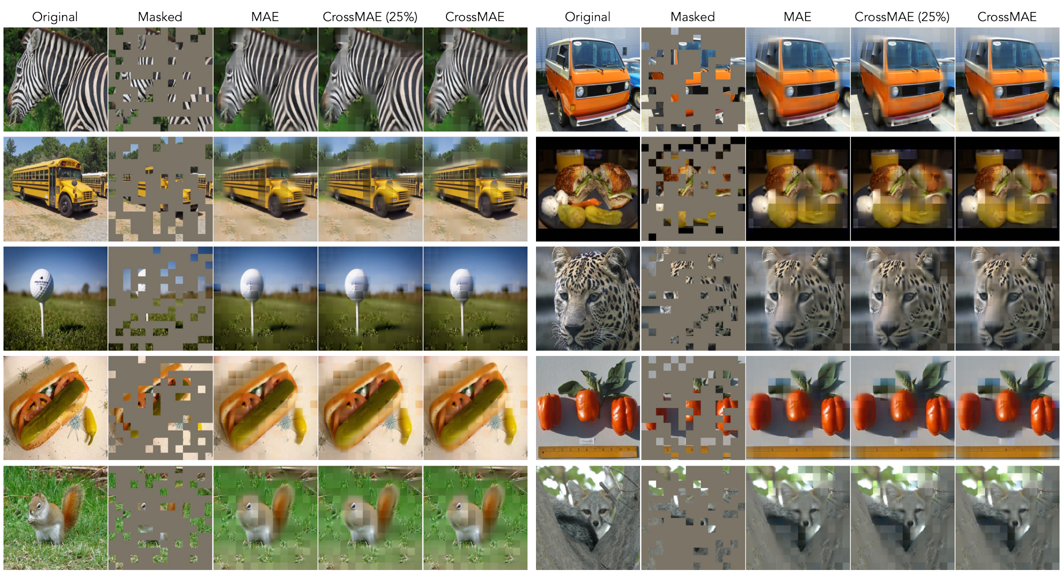


Figure 2: Example reconstructions of ImageNet *validation* images. For each set of 5 images, from left to right, are the original image, masked image with a mask ratio of 75%, MAE [30], CrossMAE (trained to reconstruct 25% of image tokens, or 1/3 of the mask tokens), and CrossMAE (trained to reconstruct 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), indicating that the encoder rather than the decoder effectively captures global image information in its output tokens. Its comparable reconstruction quality to full-image-trained models suggests that full-image reconstruction might not be essential for effective representation learning.

independently read out from the encoder output, allowing the reconstruction of only a small subset of masked patches, which in turn, accelerates the pretraining without performance degradation?

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 as queries to read out the masked reconstructions from the visible tokens in a *cross-attention decoder*. 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. **Independent partial reconstruction.** With self-attention removed, the decoding of each mask token, based on the encoded features from visible tokens, becomes conditionally independent. 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 can 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 leverage a mix of low-level and high-level feature maps from the encoder, improving the learned representation.

The analysis performed on CrossMAE led to a novel way to understand MAE. Even though the patches to be reconstructed are independently decoded, our findings demonstrate that *coherent* reconstruction for each masked patch can be independently read out from the encoder output, without any interactions among masked tokens in the decoder for consistency (Figure 2). Furthermore, the downstream performance of the model remains robust even without these interactions (Figure 1.(c), Tables 1 and 2). Both pieces of evidence confirm that the encoder's output features already encapsulate the necessary global context for image reconstruction, while the decoder simply performs a readout from the encoder output to reconstruct the pixels at the location of each patch.

To sum up, our main contributions are the following:

1. **We present a novel understanding of MAE.** Our findings show that MAE reconstructs coherent images from visible patches *not through interactions between patches to be reconstructed* in the decoder but by *learning a global representation within the encoder*. This is evidenced by the model's ability to generate coherent images and maintain robust downstream performance without such interactions, indicating the encoder effectively captures global image information.
2. **We advocate replacing self-attention layers with a simple cross-attention readout function.** Given our discovery that the encoder in MAE already captures a comprehensive global representation, we propose replacing self-attention layers in the decoder with a more efficient information readout function. Specifically, we suggest utilizing *cross-attention* to aggregate the output tokens of the encoder into each input token within the decoder layers *independently*, thereby eliminating the need for token-to-token communication within the decoder.
3. **CrossMAE achieves comparable or superior performance with reduced computational costs** in image classification and instance segmentation compared to MAE on vision transformer models *ranging from ViT-S to ViT-H*. Code is available here.

2 Related Works

2.1 Self-Supervised Learning

In self-supervised representation learning, a model trains on a pretext task where the supervision comes from the input data itself without labels. Contrastive learning methods learn representations by contrasting positive and negative samples, such as SimCLR [11], CPC [44], MoCo [29, 12, 13], CLD [59] and SwAV [7]. Additionally, in BYOL [26], iBOT [65], DINO [8], DINOv2 [45], and MaskAlign [62] make a student model to imitate a teacher model without negative pairs.

Generative modeling, focusing on acquiring a generative model capable of capturing the underlying data distribution, is an alternative method for self-supervised learning. VAE/GAN [35] merges the strengths of variational autoencoders and generative adversarial networks to acquire disentangled representations of data. PixelCNN, PixelVAE, and PixelTransformer [55, 27, 54] 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 [19] and its extensions [41, 34] use a bidirectional transformer and present few-shot learning capabilities from masked language modeling. GPT [47, 48, 5], 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 [20] 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 [3], MAE [30], and Sim-

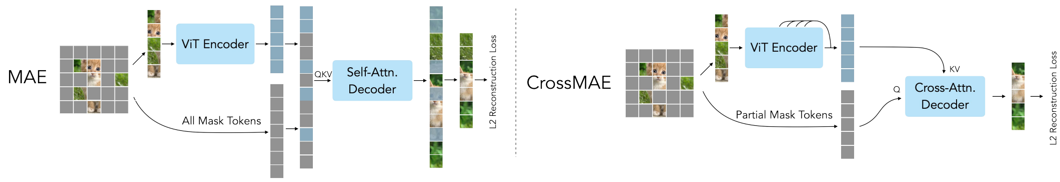


Figure 3: MAE [30] concatenates *all* mask tokens with the visible patch features from a ViT encoder and passes them to a decoder with self-attention blocks to reconstruct the original image. Patches that correspond to visible tokens are then dropped, and an L2 loss is applied to the rest of the reconstruction as the pretraining objective. CrossMAE instead uses cross-attention blocks in the decoder to reconstruct only a subset of the masked tokens.

MIM [61] are a few of the early works that explored BERT-style pretraining of vision transformers. Compared to works in NLP, both MAE and SimMIM [30, 61] 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 [61, 40] and study data the role of data augmentation [9, 21]. Many subsequent works present similar successes of masked pretraining for video [52, 58, 22, 28], language-vision and multi-modal pretraining [1, 39, 23] and for learning both good representations and reconstruction capabilities [60, 37].

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 [60] both use cross-attention for masked image generation and representation learning. Siamese MAE [28] 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. PCAE [36] progressively discards redundant mask tokens through its network, leading to a few tokens for reconstruction. VideoMAEv2 [58] concatenates randomly sampled masked tokens with visible tokens and uses self-attention to reconstruct the masked patches. In comparison, we minimally modify MAE with a cross-attention-only decoder and masked tokens are decoded in a conditional independent way.

2.3 Applications of Cross-Attention

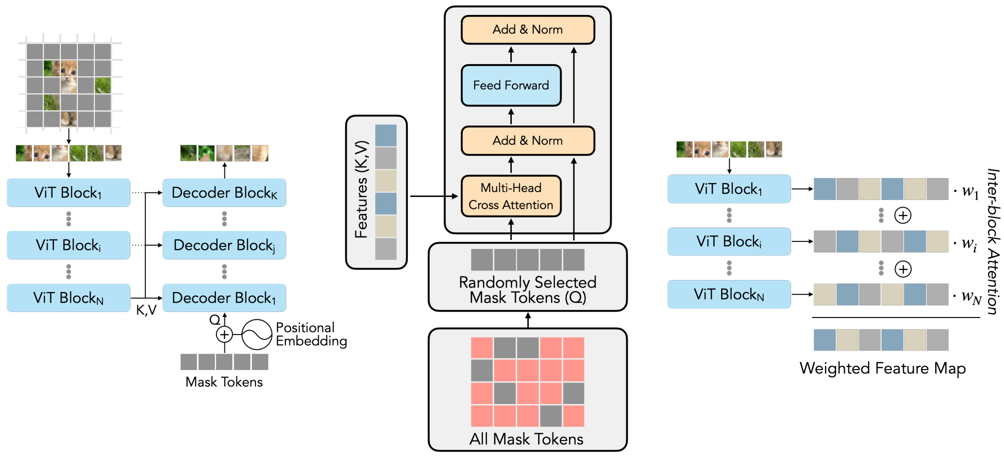
In addition to the prevalent use of self-attention in computer vision, cross-attention has shown to be a cost-effective way to perform pooling from a large set of visible tokens. Intuitively, cross-attention can be seen as a parametric form of pooling, which learnably weighs different features. Touvron et al. [53] replace mean pooling with cross-attention pooling and find improvement in ImageNet classification performance. Jaegle et al. [32] uses cross-attention to efficiently process large volumes of multi-modal data. Cross-attention is also widely used for object detection. Carion et al. [6] utilizes query tokens as placeholders for potential objects in the scene. Cheng et al. [16, 15] further extend this concept by introducing additional query tokens to specifically tackle object segmentation in addition to the query tokens for object detection. Distinct from thes prior works, we are interested the role of cross-anttention for representation learning in a self-supervised manner.

3 CrossMAE

We start with an overview of vanilla masked autoencoders in Section 3.1. Next, in Section 3.2, we introduce the use of cross-attention in place of self-attention in the decoder for testing the necessity of interaction between mask tokens for representation learning. In Section 3.3, we discuss how eliminating self-attention in the decoding process enables us to reconstruct only a subset of masked tokens, leading to faster pretraining. Finally, Section 3.4 presents our inter-block attention mechanism, which allows decoder blocks to leverage varied encoder features.

3.1 Preliminaries: Masked Autoencoders

Masked Autoencoders (MAE) [30] pretrain Vision Transformers (ViTs) [20]. Each image input is first patchified, and then a random subset of the patches is selected as the visible patches. As depicted in Figure 3, the visible patches, concatenated with a learnable class token `[CLS]`, are subsequently



(a) CrossMAE (b) Partial Masking (c) Inter-block Attention

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.

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 pass 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 we measure the mean attention value across all attention maps over the ImageNet validation set to study the properties of MAE. We grouped the attention values by cross-attention and self-attention between visible and masked tokens. We observed 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 question: *Can we simplify the decoding process by eliminating self-attention among masked tokens without compromising the model's ability to generate coherent images and perform well on downstream tasks?*

3.2 Reconstruction with Cross-Attention

To address this question, we substitute the self-attention mechanism in the decoder blocks with cross-attention, using it as a readout function to decode the latent embedding from the encoder to raw pixel values. 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 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 causal self-attention layer 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 an MLP. However, in CrossMAE, as encoder features are integrated at various decoder blocks, we embed the projection within the multi-head cross-attention module.

Cross-attention layers serve as a readout function that decodes the global representation provided in the encoder's output tokens to the pixel values within each patch to be reconstructed. However, CrossMAE does not restrict the architecture to a single cross-attention block. Instead, we stack multiple cross-attention decoder blocks in a manner more akin to the traditional transformer [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 partial reconstruction in contrast to the original full-image reconstruction in the 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 decreases 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 mask tokens as the input to the self-attention decoder, which creates an information bottleneck by making early encoder features inaccessible for the decoder. In contrast, CrossMAE's cross-attention decoder decouples queries from keys and values. This decoupling allows different cross-attention decoder blocks to take in feature maps from different encoder blocks. This added degree of flexibility comes with a design choice for selecting encoder features for each decoder block. One naive choice is to give the feature of the ith encoder block to the last ith decoder (*e.g.*, feeding the feature of the first encoder to the last decoder), in a U-Net-like fashion. However, this assumes the decoder's depth matches the depth of the encoder, which is not the case for MAE or CrossMAE.

Instead of manually matching each decoder block with an encoder feature map, we make the selection *learnable* and propose inter-block attention for feature fusion for each decoder block (Figure 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

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

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

Method	ViT-S	ViT-B	ViT-L	ViT-H
Supervised [50]	79.0	82.3	82.6	83.1
DINO [8]	-	82.8	-	-
MoCo v3 [14]	81.4	83.2	84.1	-
BEiT [3]	-	83.2	85.2	-
MultiMAE [2]	-	83.3	-	-
MixedAE [9]	-	83.5	-	-
CIM [21]	**81.6**	83.3	-	-
MAE [30]	78.9	83.3	**85.4**	85.8
CrossMAE (25%)	79.2	83.5	**85.4**	**86.3**
CrossMAE (75%)	79.3	**83.7**	**85.4**	-

Table 1: *ImageNet-1K classification accuracy*. CrossMAE performs on par or better than MAE. All experiments are run with 800 epochs. The best results are in **bold** while the second best results are underlined.

	AP^{box}		AP^{mask}	
Method	ViT-B	ViT-L	ViT-B	ViT-L
Supervised [38]	47.6	49.6	42.4	43.8
MoCo v3 [14]	47.9	49.3	42.7	44.0
BEiT [3]	49.8	53.3	44.4	47.1
MixedAE [9]	50.3	-	43.5	-
MAE [38]	51.2	54.6	45.5	48.6
CrossMAE	**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.

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 this in Table 3d.

We show that using the weighted features rather than simply using the features from the last block greatly improves the performance of CrossMAE. Intriguingly, in the process of learning to achieve better reconstructions, early decoder blocks tend to prioritize information from later encoder blocks, while later decoder blocks focus on earlier encoder block information, as demonstrated in Section 4.5.

4 Experiments

We perform self-supervised pretraining on ImageNet-1K, following MAE [30]'s hyperparameter settings, only modifying the learning rate and decoder depth. The hyperparameters were initially determined on ViT-Base and then directly applied to ViT-Small, ViT-Large, and ViT-Huge. Both CrossMAE and MAE are trained for 800 epochs. We provide implementation details and more experiments in the appendix.

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 Table 1, CrossMAE outperforms vanilla MAE using 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 downstream classification performance of the pre-trained model. Moreover, CrossMAE outperforms other self-supervised and masked image modeling baselines, *e.g.*, DINO [8], MoCo v3 [14], BEiT [3], and MultiMAE [2].

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 [38], 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 [30]. We compare against supervised pre-training, MoCo-v3 [14], BEiT [4], and MAE [30].

Results. As listed in Table 2, CrossMAE, with the default 75% prediction ratio, performs better compared to these 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. This observation also supports our hypothesis that partial reconstruction is suprisingly sufficient for learning dense visual representation.

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 3(d)) indicates using only the feature from the last encoder block. We use 25% prediction ratio for both settings in Table 3(f) to accelerate training.

4.3 Ablations

Cross-Attention vs Self-Attention. As shown in Table 3a, CrossMAE, with its cross-attention-only decoder, outperforms vanilla MAE in downstream tasks as noted in Section 4.1. Additionally, combining cross-attention with self-attention does not enhance fine-tuning performance, indicating that cross-attention alone is adequate for effective representation learning.

Mask Ratio and Prediction Ratio. In our experiments with different mask and prediction ratios (*i.e.*, the ratio of mask tokens to all tokens and the ratio of reconstructed tokens to all tokens, respectively) (see Table 3b and Table 3c), we found that our method's performance is not significantly affected by variations in the number of masked tokens. Notably, CrossMAE effectively learns representations by reconstructing as few as 15% of tokens, compared to the 100% required by vanilla MAE, with minimal impact on downstream fine-tuning performance, which shows that partial reconstruction is sufficient for effective representation learning.

Inter-block Attention. Our ablation study, detailed in Table 3d, explored the impact of varying the number of encoder feature maps in our inter-block attention mechanism. We found that using only the last feature map slightly lowers performance compared to using all 12. However, even a partial selection of feature maps improves CrossMAE's performance, with the best results obtained using 6 feature maps. This indicates that CrossMAE does not require all features for optimal performance.

Decoder Depth. Table 3e shows that a 12-block decoder slightly improves performance compared to shallower ones. Remarkably, CrossMAE achieves similar results to MAE with just one decoder block, demonstrating its efficiency. Our experiments in Figure 7 that models with lower prediction ratios benefit more from deeper decoders.

Input Resolution. We extend CrossMAE to longer token lengths by increasing the image resolution with constant patch size. Escalating the resolution from 224 to 448 increases the token length from 197 to 785, challenging the scalability of current approaches. Thus, we opt for a CrossMAE variant with a 25% prediction ratio. In Table 3f, we observe that the classification accuracy positively correlates with the input resolution, indicating that CrossMAE can scale to long input sequences.

4.4 Training Throughput and Memory Utilization

Due to partial reconstruction and confining attention to between mask tokens and visible tokens, CrossMAE improves pre-training efficiency over MAE. Results in Table 10 show that the FLOPs

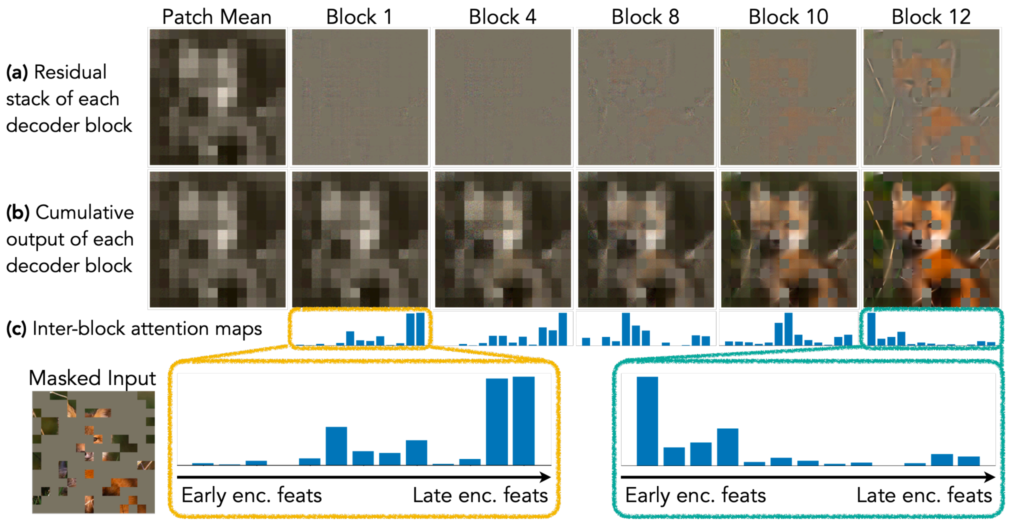


Figure 5: We visualize the output of each decoder block. (a-b) **Different decoder blocks play different roles in the reconstruction**, with most details emerging at later decoder blocks, which confirms the motivation for inter-block attention. (c) Visualizations of inter-block attention shows that **different decoder blocks indeed attend to feature from different encoder blocks**, with later blocks focusing on earlier encoder features to achieve reconstruction. The reconstructions are unnormalized w.r.t ground truth mean and std for each patch.

reduction does translate to an 1.54× training throughput and at least 50% reduction in GPU memory utilization compared to MAE.

4.5 Visualizations

Visualizing Per-block Reconstruction. Rather than only visualizing the final reconstruction, we have two key observations that allow us to visualize the work performed by each decoder block: **1)** Transformer blocks have skip connections from their inputs to outputs. **2)** The final decoder block's output goes through a linear reconstruction head to produce the reconstruction. As detailed in Appendix D, we can factor out each block's contribution in the final reconstruction with linearity.

This decomposition allows expressing the reconstruction as an image stack, where summing up all the levels gives us the final reconstruction. As shown in Figure 5 (a,b), we observe that different decoder blocks play different roles in reconstruction, with most details emerging at later decoder blocks. This justifies the need for low-level features from early encoder blocks, motivating inter-block attention.

Visualizing Inter-block Attention Maps. As shown in the visualizations of the attention maps of inter-block attention in 5(c), CrossMAE 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 underscores the necessity for different decoder blocks to attend to different encoder features, correlating with the performance improvements when inter-block attention is used.

5 Discussion and Conclusion

In our study, we present a novel understanding of MAE, demonstrating that coherent image reconstruction is achieved not through interactions between patches in the decoder but by learning a global representation within the encoder. Based on this insight, we propose replacing self-attention layers in the decoder with a simple readout function, specifically utilizing cross-attention to aggregate encoder outputs into each input token within the decoder layers independently. This approach, tested across models ranging from ViT-S to ViT-H, achieves comparable or better performance in image classification and instance segmentation with reduced computational requirements, showcasing the potential for more efficient and scalable visual pretraining methods. Our findings underscore the efficacy of the encoder's global representation learning, paving the way for streamlined decoder architectures in future MAE implementations. CrossMAE's efficiency and scalability demonstrate potential for large-scale visual pretraining, particularly on underutilized in-the-wild video datasets. However, our work has not yet explored scaling to models larger than ViT-H, the largest model examined in MAE, leaving this for future research.

References

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

[2] 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.

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

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021.

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

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

[16] 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.

[17] 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.

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

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

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[21] 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.

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

[23] 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.

[24] 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.

[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 preprint arXiv:1706.02677*, 2017.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

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

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

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

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[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.

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

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

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[58] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023.

[59] 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.

[60] 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.

[61] 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.

[62] Hongwei Xue, Peng Gao, Hongyang Li, Yu Qiao, Hao Sun, Houqiang Li, and Jiebo Luo. Stare at what you see: Masked image modeling without reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22732–22741, 2023.

[63] 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.

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

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

A Implementation details

A.1 Attention Calculation

To compare the attention values for mask tokens in vanilla MAE (Figure 1), we trained a ViT-B/16 MAE for 800 epochs using the default hyperparameters provided in [30]. 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.

A.2 Inter-Block Attention

We tried a few implementations for inter-block attention (IBA) 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:

```python
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)
```

Additionally, we further investigate the importance of using a cross-attention decoder, where each decoder block can use different feature maps from the encoder for decoding. In this experiment, we incorporated IBA into MAE, which uses only a self-attention decoder. Specifically, we concatenate the interblock attention features with the masked tokens. We then feed the combined features into MAE's self-attention decoder. We pre-trained the model and finetuned it for Imagenet classification. The results are presented in Table. 4, where all models are pre-trained for 400 epochs. We observe that inter-block attention has negligible performance improvements for MAE, potentially because MAE only takes in one feature map in its decoder. In contrast, inter-block attention allows cross-attention layers in CrossMAE to attend to features from different encoder blocks, thanks to its decoupling of queries with keys and values.

A.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

Method	Acc. (%)
MAE	83.0
MAE + IBA	83.0
CrossMAE (25%)	83.2
CrossMAE (75%)	**83.3**

Table 4: For MAE, inter-block attention has very small differences in terms of finetuning performance, potentially due to the fact that MAE's decoder only takes in one set of features.

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 multiheaded 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 Figure 6 for a more visual presentation of the method.

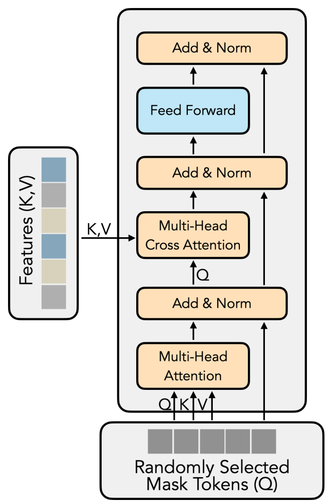


Figure 6: Modification for self-attention ablation

A.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.

In addition to the simple design of inter-block attention proposed above, we also experimented with a variant of inter-block attention by further parameterizing the attention with linear projections. Specifically, rather than directly performing weighted sum aggregation to form the features for each cross-attention layer in the decoder, we added a linear projection for each encoder feature before the feature aggregation. We denote this variant as *CrossMAE+LP*. As shown in the Table. 5 (with ViT-B pre-trained for 400 epochs, consistent with the setting in Table. 3), adding a linear projection slightly improves the performance. This indicates that it is possible to design variants of readout functions, such as through improved inter-block attention, to improve the feature quality of CrossMAE.

Method	Acc. (%)
CrossMAE	83.3
CrossMAE + LP	**83.5**

Table 5: Improving inter-block attention by adding linear projections to the input features. The performance gain indicates that it is possible to design variants of readout functions to improve CrossMAE.

A.5 Hyperparameters

Pre-training: The default setting is in Table 6, which is consistent with the official MAE [30] 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 [30]. Additionally, we use the linear learning rate scaling rule [25]. 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, such as decoder block width, 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.

Config	Value
optimizer	AdamW [43]
base learning rate	1.5e-4
learning rate schedule	cosine decay [42]
batch size	4096
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.95 [10]
warm up epoch [24]	20, 40
total epochs	400, 800
augmentation	RandomResizedCrop, RandomHorizontalFlip

Table 6: Pretraining Hyperparameters

A.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 [38] 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 [18] to accelerate attention calculation.

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) [17]
label smoothing [51]	0.1
mixup [64]	0.8
cutmix [63]	1.0
drop path [31]	0.1

Table 7: Finetuning Hyperparameters

B Additional Experiments

B.1 Linear Probe

We provide linear probe comparisons (at 800 epochs) for ViT-Small and ViT-Base in Table. 8. For both of these experiments, we run CrossMAE with a prediction ratio of 75% (reconstruction of all masked patches). These results show that CrossMAE achieves slightly better linear probe performance than vanilla MAE.

Method	ViT-S	ViT-B
MAE	49.7	65.1
CrossMAE	**51.5**	**65.4**

Table 8: Linear probe experiments of CrossMAE.

B.2 Masking Strategy

Method	Acc. (%)
Grid Masking	83.2
Random Masking	**83.3**

Table 9: Ablation of masking strategies.

Similar to MAE [30], we here ablate the masking pattern. Instead of random masking, we perform grid-wise sampling that "keeps one of every four patches" (see MAE Figure 6). The finetuning performance is reported in Table. 9 for ViT-B (at 400 epochs), which shows that grid masking does not lead to additional improvements in downstream performance.

C Runtime and GPU Memory Comparisons with MAE

Method	Memory (MB/GPU)	Runtime (min/epoch)	Acc. (%)
MAE	OOM (>81920)	5.19*	83.3
CrossMAE	**41177**	**3.38**	**83.5**

Table 10: CrossMAE greatly improves the training throughput and reduces the memory requirements, lowering the barrier for masked pretraining. Statistics are measured on 2 NVIDIA A100 80GB GPUs. Please refer to Appendix C for comparison details. *: MAE's default batch size exceeds the capacity of 4 GPUs, requiring gradient accumulation for runtime measurement.

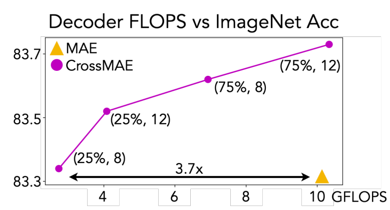


Figure 7: We compare ViT-B which is pre-trained for 800 epochs with different variants of CrossMAE v.s. MAE. For CrossMAE, we vary the prediction ratio p and number of decoder blocks n, and we denote each as (p, n). While all experiments are run with inter-block attention, CrossMAE has lower decoder FLOPS than MAE [30] and performs on par or better.

All experiments in Table 10 are conducted on a server with 4 NVIDIA A100 (80GB) GPUs, with the standard hyperparameters provided above for pretraining. NVLink is equipped across the GPUs. We use the default setting for MAE and set the global batch size to 4096. For CrossMAE, we also use the default setting with a prediction ratio 0.25, and this takes around 41GB memory per GPU without gradient accumulation (i.e., local batch size is set to 1024 samples per GPU). However, the same local batch size results in out-of-memory (OOM), which indicates that the total memory requirement is larger than the available memory for each GPU (80GB). To run MAE on same hardware, we thus employ gradient accumulation with a local batch size of 512 to maintain the global batch size. The benchmark runs each method and measures the average per epoch runtime as well as the max memory allocation for 10 training epochs. Our experiments in Figure 7 show that models with lower prediction ratios benefit more from deeper decoders. Our model performs on par or better when compared to MAE, with up to 3.7× lower decoder FLOPS.

D Visualizing the Contributions per Decoder Block

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 enable per-block visualization: **1)** Transformer blocks have residual connections from their inputs to outputs. Let f_i be the output and $g_i(\cdot)$ the residual function of decoder i, so $f_i = f_{i-1} + g_i(f_{i-1})$. **2)** The final decoder block's output goes through a reconstruction head h, which is linear, consisting of a layer-norm and a linear layer, to produce the reconstruction. With D as the decoder depth, f_0 the initial input, and y the final output, y is recursively defined as $y = h(f_{D-1} + g_D(f_{D-1}))$, which simplifies due to the linearity of h:

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

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 Figure 5.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract are justified in the method and the experiments section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work have been discussed in the Discussion and Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This work offers observations and hypotheses justified with empirical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our code, which reproduces our results, is provided through an anonymous link in the abstract.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our method is evaluated on open datasets that are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We follow the hyperparam selection from MAE. The hyperparams introduced by our work, such as the mask ratio and the number of feature maps used, are ablated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because they would be too computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We described the compute requirements in Appendix A.6. We do not use GPUs from a cloud provider.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This paper aims to advance the field of self-supervised learning. Like other self-supervised learning methods, our work may have various societal implications. However, we do not believe any specific consequences need to be highlighted in this context.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The code and datasets used in this work follow the original MAE work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.