Self-Guided Masked Autoencoder

Jeongwoo Shin[1], **Inseo Lee**[1], **Junho Lee**[1], **Joonseok Lee**[1,2*]
[1]Seoul National University, [2]Google Research
{swswss, ian.lee, joon2003, joonseok}@snu.ac.kr

Abstract

Masked Autoencoder (MAE) is a self-supervised approach for representation learning, widely applicable to a variety of downstream tasks in computer vision. In spite of its success, it is still not fully uncovered what and how MAE exactly learns. In this paper, with an in-depth analysis, we discover that MAE intrinsically learns pattern-based patch-level clustering from surprisingly early stages of pre-training. Upon this understanding, we propose *self-guided masked autoencoder*, which internally generates informed mask by utilizing its progress in patch clustering, substituting the naive random masking of the vanilla MAE. Our approach significantly boosts its learning process without relying on any external models or supplementary information, keeping the benefit of self-supervised nature of MAE intact. Comprehensive experiments on various downstream tasks verify the effectiveness of the proposed method.

1 Introduction

Self-supervised learning has been an attractive direction to alleviate the substantial cost for data annotation. For example, Masked Language Modeling (MLM), predicting masked words of an input sentence, is demonstrated to capture contextual meaning of a word by BERT [13] and GPT [8]. Motivated from the success of MLM, Masked Image Modeling (MIM) has been introduced in computer vision, utilizing abundant unlabeled image data. Among them, Masked Autoencoder (MAE) [22], equipped with a Vision Transformer (ViT) [15]-based asymmetric encoder-decoder structure, demonstrates that simple reconstruction of the RGB pixels for the masked patches is enough to achieve competitive performance on various downstream tasks.

In the wake of MAE's impressive performance, a succession of studies have emerged aiming to augment its capabilities through the integration of informed masking techniques. These innovative endeavors leverage diverse sources of additional information, including attention maps generated by a supervised ViT [28], knowledge learned by pre-trained self-supervised models [32, 10], or supplementary adversarial modules [45], all aiming at refining the quality of the masks. However, these prevailing approaches have merely applied informed masking without truly understanding the mechanism of MAE, relying on external resources such as pre-trained models or labels.

To this end, we embark on an in-depth analysis through extensive experiments to understand the internal operation of MAE, as it is still not fully uncovered *what* and *how* MAE exactly learns, despite the several prior endeavors [9, 41, 61, 29]. Based on our analysis of MAE, we then explore the potential of MAE to produce informed masks on its own. We first demonstrate that MAE intrinsically learns *pattern-based patch-level clustering* and this property emerges from *extremely early stages* of pre-training (Section 3.3). We then unveil the underlying mechanism of the mask tokens in the decoder (Section 3.4). Upon this understanding, we propose a novel method to *boost* the training process of MAE via informed masks, generated in an *entirely unsupervised manner* without incurring any external models or supplementary information, unlike the previous informed masking methods.

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

(a) Input image (b) Random masking (c) MAE feature (d) Bi-partitioned (e) Informed masking (f) Informed with hint (g) Ours feature

Figure 1: **Illustration of our self-guided MAE**.

Figure 1 illustrates our model compared to the original MAE. Unlike the random masking (b), our method generates informed masks covering the main object entirely (e-f) using the distinguishable patch representations (d) emerging from a very early stage of the training. With the internally produced informed masks, MAE accelerates its training process of learning patch-level clustering, leading to clearer and finer embedding space (c, g). Our contributions are summarized as follows:

- We discover that MAE learns *pattern-based patch-level clustering* within each image, emerging from *incredibly early stage* of the pre-training process.
- We propose a new masking strategy, *self-guided masked autoencoder*, relying solely on internal quantification of the progress in patch-clustering, free from external models or labels.
- Our comprehensive experiments across various downstream tasks validate that our proposed method genuinely expedites the learning process of MAE.

2 Preliminary

Masked Autoencoder (MAE). MAE [22] aims to learn task-agnostic feature representations for various downstream vision tasks, *e.g.*, classification, detection, or segmentation.

Given an image of size $H \times W$, MAE first splits it into to same-sized $P \times P$ image patches. Each patch is linearly mapped to a d-dimensional embedding. As a result, the input image is represented as a set of these features, denoted by $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)} : \mathbf{x}^{(i)} \in \mathbb{R}^d\}$, where $n = HW/P^2$ is the number of patches. We call $\mathbf{x}^{(i)}$ as a 'patch' or 'token' embedding interchangeably. MAE randomly masks out a subset of n patches in X. The set of masked and visible patches are denoted by X_m and X_v, respectively, where $X_m \cup X_v = X$ and $X_m \cap X_v = \emptyset$.

MAE adopts an asymmetric encoder-decoder structure based on ViT [15]. The encoder E takes X_v as input and produces a same-sized set of embeddings, denoted by $X'_v = \{\mathbf{x}'^{(i)} : \mathbf{x}'^{(i)} \in E(X_v)\}$. Through the encoding, the patch representations are updated to reflect the context of the entire image, only from the visible parts. Then, the decoder D takes a set of n patch embeddings, denoted by $\tilde{X} = \{\tilde{\mathbf{x}}^{(i)} : \tilde{\mathbf{x}}^{(i)} = \mathbf{m} \text{ if } \mathbf{x}^{(i)} \in X_m; \tilde{\mathbf{x}}^{(i)} = \mathbf{x}'^{(i)} \text{ if } \mathbf{x}^{(i)} \in X_v\}$, as input, where $\mathbf{m} \in \mathbb{R}^d$ is a learnable mask token. Each $\mathbf{x} \in X_m$ is substituted with a mask token $\mathbf{m}$, and a corresponding positional encoding is applied to distinguish them. The decoder targets to reconstruct the raw RGB pixels of X_m. Once trained, only the encoder is deployed for downstream tasks.

Hierarchical Latent Variable Model. Kong *et al.* [29] recently discovers that the internal operation of MAE can be explained under the framework of a hierarchical latent variable model. There exists high-level shared information $\mathbf{c}$ in the input image, and it is equivalent to statistical dependency among the patches in X. The MAE encoder $E(X_v)$ learns high-level latent variables by estimating

Image 1

Image 2

Representations of image 1 Representations of image 2

Figure 2: Relationships among the patch embeddings. (a) Pairwise similarity matrix for all 196×196 pairs of patches. (b) Similarity between the mean patch and all individual patches. (c) Attention score of the class token.

the shared information $\hat{c}$ from the visible patches X_v, and the decoder $D([E(X_v); m])$ performs the reconstruction task by inducing $\hat{X}_m$ from $\hat{c}$ via the mask tokens.

3 Analysis of MAE

We study what and how MAE learns with a concept of token relation (Section 3.1), and demonstrate that it learns pattern-based patch-level clustering (Section 3.2) from early stages of training (Section 3.3). We then illuminate the underlying mechanism of the MAE decoder (Section 3.4). In this section, we use the ViT-B MAE [22] pre-trained for 400 epochs on ImageNet-1K [12] and all experiments have been conducted on 10% of ImageNet-1K training set, unless noted otherwise.

3.1 Token Relation

In order to understand what MAE learns, we analyze its token embeddings with their quantified pair-wise relationships, *i.e.*, attention score matrix A and cosine similarity matrix M. For the input patches $X \in \mathbb{R}^{n \times d}$ and the transformer weights $W^{\{Q,K,V\}} \in \mathbb{R}^{d \times d'}$ for queries, keys, and values, respectively, $A \in \mathbb{R}^{n \times n}$ and $M \in \mathbb{R}^{n \times n}$ are given by

$$A = \mathrm{Softmax}\left(XW^Q (XW^K)^\top / \sqrt{d'}\right); \quad (1) \qquad M_{i,j} = \frac{x_i'^\top x_j'}{\|x_i'\|_2 \|x_j'\|_2}; \quad (2)$$

where $X' = A(XW^V) \in \mathbb{R}^{n \times d'}$ and $x_i' \in \mathbb{R}^{d'}$ indicates the i-th row of X'.

We present analysis with A and M under two settings. First, we calculate them using complete set of patches X at the encoder as $E(X)$. This ideal setting offers the most accurate $\hat{c}$, suitable to analyze the features learned by MAE. As an alternative, they can be obtained from the practical setting of the decoder $D([E(X_v); m])$ which contains mask tokens. Since only the visible tokens are exploited to estimate $\hat{c}$, this setting would produce less accurate token relations compared to the former one.

3.2 What is Learned by MAE?

We investigate the distribution of patch relationships in the learned embedding space, using the last layer embeddings of $E(X)$ and $D([E(X_v); m])$ for 196 (14×14) patches of set-aside test images.

Qualitative Analysis. In Figure 2, we compare the patch representations among different models (MAE, MoCo [23], and ViT [15]). Figure 2a depicts the normalized pairwise cosine similarity matrix (M) for all 196×196 patch pairs for a test image. The MAE encoder shows more polarized values, *i.e.*, higher variance, indicating that patches are more clearly clustered. Figure 2b illustrates the cosine similarity between the mean of patches and all individual patches. In the examples in Figure 2, the background patches are majority, so the mean patch is closer to the background. Patches corresponding to the main object clearly show lower similarity to the mean (background), indicating that the MAE encoder has learned patch clustering based on visual patterns, *i.e.*, texture and color. Similar results in the projected latent space are provided in Appendix B. Figure 2c shows the attention scores of the class ([CLS]) token. As the class token is not updated during self-supervised training, it does not carry particularly meaningful information and therefore could be regarded as a random vector. As a result, the class token does not lean towards any specific patch under self-supervision, and thus the score is distributed similarly to the relationship with the mean patch in Figure 2b. In contrast, MoCo and ViT fail to clearly distinguish the patterns among the whole patches.

Despite the limited information X_v, the decoder also exhibits proficiency in grouping patches based on their patterns, albeit not as effective as the encoder.

Quantitative Analysis. We additionally measure the feature variance (${}_F$) and variance of the pairwise similarities (${}_S$), on the ImageNet-1K validation set:

$$ {}_F = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i}{\|x_i\|_2} - x\right)^2; \quad {}_S = \sum_{i=1}^{n}\sum_{j \neq i}\frac{(M_{i,j} - M)^2}{n(n-1)}; \tag{3} $$

where $x = \frac{1}{n}\sum_i x_i/\|x_i\|_2$ and $M = \frac{1}{n(n-1)}\sum_{i,j} M_{i,j}$. Higher ${}_F$ indicates patch embeddings are spread out more widely in the feature space, while higher ${}_S$ indicates stronger patch clustering.

In Table 1, the MAE encoder and decoder show significantly higher ${}_F$ and ${}_S$ compared to MoCo and ViT, suggesting that their patch embeddings are more diversely clustered in the embedding space rather than in a simpler alternative, *e.g.*, bi-partition. Given the significant utilization of high-frequency information (*e.g.*, pattern or texture) in MAE (Figure 7), we can quantitatively confirm that MAE effectively clusters patches based on their patterns. MoCo and ViT show significantly lower ${}_F$ and ${}_S$, as they tend to learn a simpler form of feature maps, aligned with [42]. To alleviate the concern that the large variance might be a result of a few extremely clustered features, instead of good separability, we additionally measure Normalized Mutual Information (NMI) [46] between queries and keys, which is an indicator of homogeneity in attention map [42]. As shown in Figure 6, we confirm with non-zero NMI that MAE does not collapse to a few extremely separated feature groups.

Table 1: Feature variance (${}_F$) and similarity variance (${}_S$).

Feature	E[${}_F$]	E[${}_S$]
MAE encoder	0.08	0.075
MAE decoder	0.11	0.059
MoCo [23]	0.01	0.003
ViT [15]	0.02	0.012

Summary. MAE learns patch-level clustering in an image based on their visual patterns. Operating only with visible tokens, the decoder learns a similar but less clearer trend than the encoder.

3.3 When Does MAE Learn Patch Clustering?

Given that the MAE learns patch clustering upon completion of pre-training, when does it start to learn them in pre-training? We answer this question by tracking the token relations of MAE.

Evolving Bi-partitioning across Training. We start with the simplest form of token clusters, *i.e.*, bi-partitioning. We cluster the patches into the two most prominent sub-groups by applying graph-cut to M from the final layer. Based on this clustering, we trace the mean of inter-cluster edge weights (${}_{inter}$) and mean of intra-cluster edge weights (${}_{intra}$) with M and A, across the training.

Figure 3a shows ${}_{inter}$ and ${}_{intra}$ measured with M and A. We observe two notable patterns regarding the gap ${}_{intra} - {}_{inter}$: 1) the gap tends to get larger along with the training steps, more prominently with the attention scores. 2) there is a clear margin between ${}_{intra}$ and ${}_{inter}$ from very early stages. The decoder also shows a similar but less prominent trend.

Convergence of Token Relations. Going beyond investigating token clusters, we directly track the gap between the distribution of token relations during the training *vs.* upon completion. Specifically, we consider the mean KL divergence ${}_i(j)$ in the i-th layer at the j-th epoch over a set of images D:

$$ {}_{i,j} = \frac{1}{|D|}\sum_{I \in D} D_{KL}(R_j^{(i)}(I)\|R_N^{(i)}(I)); \tag{4} $$

where N is total epochs and $R_j^{(i)} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{n \times n}$ is a function mapping an input image I to a token relation matrix (*e.g.*, M or A) computed with the i-th layer embeddings at the j-th epoch.

(a) Bi-partitioning performance

(b) KL divergence of token relations

Figure 3: MAE learns patch clustering from very early stage of training process. (a) MAE widens the gap ${}_{intra} - {}_{inter}$. (b) Token relations drastically converge at early epochs and then gradually level off. Numbers in the legend denote the layer i. More details are provided in Appendix B.

Figure 3b depicts ${}_i(j)$ for even-numbered layers up to 400 epochs, measured with M and A. It clearly shows that ${}_i$ monotonically decreases, converging quickly at early epochs, indicating that the patches begin to be clustered from early epochs. This result strongly implies that MAE learns the token relations at early epochs and progressively strengthens it along the rest of training. The decoder also shows a similar trend, but with less prominence.

Summary. MAE learns to cluster the patches from the early stage of training.

3.4 Operations of the Decoder

In previous experiments, we observe that the decoder $D([E(X_v); m])$ in the practical setting is still able to build complete token relation, which verifies that the decoder exploits the estimated shared information $\hat{c}$ conveyed from the encoder $E(X_v)$ to complement the missing information in masked-out tokens (X_m) and reconstruct them. Connecting this to our discovery in Section 3.2, we claim that the pattern-based patch clustering learned by MAE conceptually corresponds to this $\hat{c}$: If the encoder is trained sufficiently, its output embeddings X'_v for the visible tokens would convey the general context (i.e., $\hat{c}$) of the entire image. Then, through the decoding process, mask tokens are contextualized by selectively attending to X'_v, thereby possessing the essential information to represent the target patches X_m, originally derived from $\hat{c}$. Therefore, by reversing this process, we can assess if the encoder has been sufficiently trained to precisely associate the patches by quantifying $\hat{c}$ deployed in $D([E(X_v); m])$, which is estimated by $E(X_v)$. Based on this idea, we propose a novel metric to measure it during training, which will be the key to our proposed method in Section 4.

Exploitation Rate. We pose that the overall attention weight on mask tokens at the decoder is a good indicator to quantify the amount of $\hat{c}$ utilized by the decoder. Specifically, we define the *exploitation rate* of the mask tokens over the decoder layers using the attention score matrix A (Equation 1), which can be interpreted as the special case of attention rollout [1]. For the sets of token indices A and B, the exploitation rate $r^{(l)}_{A \to B} \in \mathbb{R}$ of the tokens in A to construct the tokens in B at the l-th layer is defined as the average attention weights relying on the tokens in A for the tokens in B:

$$r^{(l)}_{A \to B} = \frac{1}{|B|} \sum_{i \in B} \sum_{j \in A} A^{(l)}_{ij}; \quad (5)$$

where $l = 1, \ldots, L$ is the layer index, and $A^{(l)}$ is the attention score matrix at the l-th layer. For A and B, we are interested in the set of mask tokens $M = \{i : x^{(i)} \in X_m\}$, of visible tokens $V = \{i : x^{(i)} \in X_v\}$, and of all tokens $O = \{1, \ldots, n\}$. For example, $r^{(l)}_{M \to V} = 0.7$ indicates that the contextualized visible tokens consist of mask tokens (70%) and visible tokens (30%) on average.

Then, we recursively accumulate these ratio across all layers to get the overall exploitation rate of the mask tokens. Formally, the *accumulated exploitation rate* $R^{(l)}_{A \to B} \in \mathbb{R}$ of the tokens in a set A to construct the tokens in a set B up to the l-th layer is defined by

$$R^{(l)}_{A \to B} = r^{(l)}_{A \to B} \cdot R^{(l-1)}_{A \to A} + r^{(l)}_{B \to B} \cdot R^{(l-1)}_{A \to B}, \tag{6}$$

where $R^{(0)}_{A \to B} = 1$. At the l-th layer, tokens in B (denoted by B^l) consist of tokens in both A and B from the previous layer, A^{l-1} and B^{l-1}, with their respective ratios of $r^{(l)}_{A \to B}$ and $r^{(l)}_{B \to B}$. Thus, the left term $r^{(l)}_{A \to B} \cdot R^{(l-1)}_{A \to A}$ means the overall exploitation rate of tokens in A to construct those in B^l, coming through A^{l-1}, and similarly, $r^{(l)}_{B \to B} \cdot R^{(l-1)}_{A \to B}$ indicates that coming through B^{l-1}.

Finally, the proportion of information from the visible tokens ($R^{(l)}_{V \to O}$) and that from the mask tokens ($R^{(l)}_{M \to O}$) after l-th layer with masking ratio of m are given by

$$R^{(l)}_{V \to O} = m \cdot R^{(l)}_{V \to M} + (1-m) \cdot R^{(l)}_{V \to V}, \quad R^{(l)}_{M \to O} = m \cdot R^{(l)}_{M \to M} + (1-m) \cdot R^{(l)}_{M \to V}. \tag{7}$$

Empirical Analysis. We measure the exploitation rate of visible tokens ($R^{(l)}_{V \to O}$) and mask tokens ($R^{(l)}_{M \to O}$) in Figure 4. Surprisingly, $R^{(l)}_{M \to O}$ surpasses $R^{(l)}_{V \to O}$ after some moment, denoted by T. Heavy exploitation on the mask tokens in every decoder layer strongly indicates that they truly hold substantial amount of shared information estimated by the encoder, which is more valuable than simple interpolation of visible patches (before T epochs) to represent masked out patches. We observe $T \approx 50$, but this may differ depending on the model or dataset.

Figure 4: Exploitation rate.

Summary. When the encoder is sufficiently trained to cluster patches, the encoder outputs reflect the shared information, and they are utilized to constitute mask tokens in the decoder. This means the mask tokens possess this patch clustering information and start to be intensely exploited for reconstructing masked-out patches. Thus, we can conversely infer from a high exploitation rate of the mask tokens in the decoder that the mask tokens have patch clustering information conveyed from the encoder, sufficiently to cluster the patches. This process is verified via measuring the shared information learned by the encoder $E(X_v)$ during training by tracking the accumulated exploitation rate in Equation 7. Heavy exploitation of the mask tokens in decoder after $T \approx 50$ implies that the encoder is presently trained sufficiently to cluster the patches.

4 Self-Guided Informed Masking

In Section 3.2 and Section 3.3, we show that the MAE encoder learns patch clustering from an early stage, allowing us to appropriately bi-partition the image into two major token clusters and to mask out one of them. In other words, we can generate informed masks with MAE itself early in the pre-training phase and use these informed masks for the remainder of the training. To decide when exactly the MAE can properly cluster the patches, we use exploitation rate suggested in Section 3.4, which allows us to confidently generate informed masks at T epoch, ultimately leading to the design of our method.

From these observations in Section 3, we are motivated to leverage the patch relevance learned from the early-stage to expedite training, instead of relying on random masking. Random masking delays the learning of powerful patch-clustering, inefficiently revisiting easily separable patches already clustered in the early stage which reflects the *key dissimilarities* among the image tokens.

Based on this idea, we propose our *self-guided informed masking*, which internally injects the information about the learned *key dissimilarities* by intensively masking one of the top two well-

Figure 5: Examples of Self-guided Informed masking. More examples and detailed explanations on our method are displayed in Appendix A.

separated clusters. We emphasize that MAE is still trained in a single stage; after T epoch we begin generating informed masks and continue the training process without interruption.

Armed with our method, we can accelerate MAE to focus on learning less distinguishable patches instead of wasting time repeating to discover the most prominent patterns. As our method purely relies on inherent metrics during training, it is completely free from any external models or extra information. A more detailed reasoning can be found in Appendix A.

To achieve this, we need to 1) bi-partition the image, 2) properly design the informed masks, 3) select the attention layer to construct the informed masks, and 4) decide when to start informed masking.

Bi-partitioning. To bi-partition the image reflecting the learned key dissimilarities, we take Normalized Cut (Ncut) [44] to consider both dissimilarity between different clusters and similarity inside each cluster. We construct a fully-connected undirected image graph with M (Equation 2) which consists of the patches and similarity between them as nodes and edges, respectively. To partition the set of all node indices O into two disjoint sets A and B, we minimize the Ncut energy $S(A,B)/S(A,O) + S(A,B)/S(B,O)$, where $S(A,B) = \sum_{i \in A, j \in B} M_{i,j}$. As shown in [44], we can approximate the solution of this problem by calculating the second smallest eigenvector (y_1) of the eigensystem $(D - M)y = \lambda Dy$, where D is a diagonal matrix with $D_{i,i} = \sum_j M_{i,j}$ and $D - M$ is the Laplacian matrix. Finally, we bi-partition the graph by thresholding y_1 with its mean, $\overline{y_1}$, i.e., $A = \{i \mid y_1^i \geq \overline{y_1}\}$ and $B = \{i \mid y_1^i < \overline{y_1}\}$.

Object-centric Masking. Our approach stems from the observation that masking the entire image leads to learn patch clustering across the entire image, i.e., the reconstruction loss affects the whole image. To refine this process, we restrict the masking to object-centric regions. By narrowing the masking focus, our method guides the MAE to concentrate on learning patch clustering within the object regions; that is, the loss affects only the object-related parts, thereby accelerating the process of learning patch clustering in object region. In this context, we aim to mask out the cluster C containing the main object, letting the model to learn feature representations for the foreground faster. Since we do not have access to the label, we take an indirect approach: the token with the largest absolute element in y_1 tends to compose the main object. [55]

In reality, however, practical issues like imperfect bi-partitioning and varying cluster sizes complicating batch processing, arise. To address, we rank the tokens by the relevance to C and mask out a fixed ratio of the tokens based on the ranking. The relevance score S_i of the i-th patch is defined as

$$S_i = \frac{x_m^\top x^{(i)}}{\|x_m\|_2 \, \|x^{(i)}\|_2}, \quad \text{where } x_m = \frac{1}{|C|} \sum_{i \in C} x^{(i)}. \tag{8}$$

As x_m represents the majority of C which mainly consists of the object, S_i exhibits the relevance to the object by measuring the similarity of each patch to x_m. In this manner, we can robustly mask out the whole object even with the noisy bi-partitioning.

Though intensive masking on an object leads to expedited feature learning on it, reconstruction is fundamentally impossible if all tokens are masked, as there is no clue to construct high-level information about it. To prevent such a case, we add a few hint tokens, chosen either by uniformly randomly or proportional to S_i with decaying ratio, following [32]. Figure 5 illustrates our object-centric informed masking (4th row) based on bi-partitioning (3rd row). We provide minimum information for estimating shared information via the hint tokens (last row).

Appropriate Layer for Patch Clustering. We consider attention distance and Normalized Mutual Information (NMI) in Figure 6, to decide with which layer embeddings to compute the similarity matrix among the patches. To obtain suf ciently meaningful token relation, we discard the early layers (~4th layer) from the candidates as they show extremely low NMI, which indicates relatively homogeneous attention map. Then, we select the second last encoder layer with the highest attention distance, as they have the highest potential awareness of the global patterns.

Figure 6: MAE properties.

When to Start Informed Masking. Lastly, we need to determine when the model has learned patch clustering enough to generate high-quality informed masks. Since the *relative* relationship among the tokens is enough for bi-partitioning, we generate them when the mask tokens start to have comparable amount of information to the visible tokens. Based on Section 3.4, we quantify the shared information possessed by the mask tokens and start informed masking when it becomes comparable to the information in visible tokens ($R^{(L)}_{M\to O}$ ≈ $R^{(L)}_{V\to O}$), i.e., around T epochs in Figure 4.

5 Experiments

5.1 Experimental Settings

Baselines. We compare our model to the original MAE [22] and Attention-driven Masking and Throwing (AMT) [38], a recently enhanced MAE without requiring any external model or label information. We exclude other models requiring external pre-trained models [10, 32] or labeled data [28] for generating informed masks, since it is no longer a fair comparison.

Experimental Protocol. We pre-train all competing models for 400 epochs on ImageNet-1K [12], and ne-tune on 3 downstream tasks: image classi cation, object detection, and semantic segmentation. All experiments are conducted following the settings in original MAE [22], unless noted otherwise. We conduct experiments on 8 NVidia A6000 GPUs (48GB).

Datasets. We use CIFAR-100 [30], iNaturalist 2019 [48], and CUB200-2011 [49] for image classi cation. We ne-tune our model on COCO [36] for object detection, and on ADE20K [62] for semantic segmentation.

5.2 Performance on Downstream Tasks

Image Classi cation. The left side of Table 2 compares the image classi cation performance on various datasets. Our method outperforms the baselines on all datasets with both linear probing and ne-tuning, implying that the expedited training with our method leads to stronger feature representation after same epochs of training. We provide further analysis on our boosted performance in Section 5.4 and extended training results in Appendix F.

Object Detection and Segmentation. We ne-tune a Mask R-CNN model [24] end-to-end on COCO with a ViT backbone for 90K iterations, and evaluate with the average precision for bounding box (AP^{box}) and for segmentation mask (AP^{mask}). We also compare semantic segmentation performance on ADE20K using UperNet [57], in Mean Intersection over Union (mIoU). Our method outperforms baselines in all ne-grained tasks as shown in the right side of Table 2, indicating that ours better captures the ne details of the image with the same training session.

Table 2: Performance on downstream tasks. LP and FT stand for Linear probing and Fine-tuning, respectively. Det. indicates the Object Detection task.

Task	Image Classification					Det.	Segmentation	
Dataset	ImageNet-1K		iNat2019	CIFAR	CUB	COCO	COCO	ADE20K
Metric	LP	FT	FT	FT	FT	AP^{box}	AP^{mask}	mIoU
MAE [22]	61.4	82.5	78.7	89.3	81.8	43.0	38.9	45.0
AMT [38]	61.7	82.8	76.0	87.8	80.8	42.8	36.6	43.1
Ours	62.9	83.2	78.9	90.0	82.8	43.3	39.3	45.2

5.3 Ablation Studies

Table 3 compares linear-probing performance of our method on image classification, with various settings. We fix the masking ratio to 0.75 for all the experiments in this section. Ablation studies on more factors including masking ratio can be found in Appendix D. The first group verifies that the later layers of the encoder yield the most accurate token relations, aligned well with our analysis in Section 4. The second group verifies that the hint tokens proposed in Section 4 are essential. Also, better performance with uniform sampling (Random) than with S_i-based approach (Equation 8) indicates the importance of providing equal opportunity to all clusters to have visible tokens.

Table 3: Ablation studies. The default is highlighted in gray. Detailed analysis can be found in Appendix D.

Layer	Target cluster	Hint strategy	Linear probing
Enc 3	Object	Random	62.3
Enc 7	Object	Random	62.4
Dec 8	Object	Random	62.7
Enc 11	Object	S_i-based	62.5
Enc 11	Object	No hint	52.3
Enc 11	Object	Random	62.9

5.4 Analysis on the Learned Feature Space

We take a deeper look into our method for further insights on its improvements via various metrics with the ImageNet-1K validation set. We analyze with $m = 0$ unless noted otherwise.

Attention Distance [15]. We measure the weighted average distance of the attention operations between the query and key tokens within the image in Figure 7a. Since it can be interpreted as the size of the receptive fields in CNNs, higher attention distance of our method suggests that it has been better-trained at the same epoch, more globally capturing the image context.

Fourier Analysis [42]. Figure 7b shows the relative log amplitude of Fourier-transformed representations, which indicates the degree to which the model prioritizes either high-frequency (pattern) or low-frequency (shape) information. Our method utilizes high-frequency components more intensively than MAE does, implying more powerful pattern-based clustering.

Mask Token Variance. We report the variance of mask token embeddings along the decoder layers with $m = 0.75$ in Figure 7c. As they carry high-level semantics of each potential patch cluster, higher variance among them indicates that the latent variables responsible for estimated shared information in each individual cluster has been diversified, implying that patches are grouped into finer clusters. Consistently higher variance along the layers of ours manifests its further progressed patch clustering.

Qualitative Comparison. Figure 1 clearly shows that ours captures finer patch embeddings and tighter boundaries than MAE. See Appendix C for more examples and detailed explanations.

(a) Attention distance (b) Fourier analysis (c) Mask token variance

Figure 7: Metrics explaining our performance gain. Layers left on the red dotted line belong to the encoder, and the rest to the decoder.

6 Related Work

Masked Image Modeling (MIM). Inspired by Masked Language Modeling [13, 8], MIM has been widely applied in image [63, 11, 43, 7, 59, 22, 14, 4, 2, 58, 19, 33, 3, 18, 37, 51, 16, 5, 35] and video understanding [47, 52, 17, 53, 40, 6, 56, 60, 54, 20, 21, 26, 27, 25]. Context Encoder [43] established MIM with CNNs, focusing on the prediction of masked regions of an image. MLM also has been applied to ViT [15]; e.g., BEiT [7] leverages visual tokens from dVAE as reconstruction targets for masked image patches, and SimMIM [59] directly predicts raw pixel values as a regression problem. MAE [22] also regresses the raw pixels, adopting asymmetric encoder-decoder architecture.

Informed Masking. Recent researches have considered to arm MAE with an advanced masking strategy [34, 28, 45, 10, 32, 38]. MST [34] and AttMask [28] have pioneered information-guided masks, utilizing the attention maps of a supervised ViT. ADIOS [45] adopts adversarial training to get optimal masks. SemMAE [32] and AutoMAE [10] leverage the powerful knowledge from pre-trained self-supervised ViTs, showcasing the synergistic fusion of informed masking with MAE. Despite effectiveness, these methods are limited as they require an external model or rely on labels, which make these models no longer fully self-supervised. To address this critical issue, AMT [38] extracts attention maps directly from the model during pre-training and generates informed masks from them.

Analysis on MAE. Nam et al. [42] reports a comparative analysis between MIM and contrastive learning, highlighting how the encoder-decoder architecture of MAE empowers the properties of MIM, although their discovery is confined to token differentiation of MIM models. Another avenues introduce theoretical architectures to elucidate MAE's behavior under specific assumptions [9, 41, 61, 31]. Hierarchical latent variable model [29] aligns well with our primary observations.

7 Summary and Limitations

Unveiling the operation of Masked Autoencoder (MAE) which fundamentally learns pattern-based patch-level clustering, we expedite the MAE to learn patch clustering by incorporating informed mask derived from itself. Notably, our method does not require any other external models or additional information. Superior results on extensive experiments demonstrate the effectiveness of our method.

Limitations. Our method may show less significant improvement when training with excessively fragmented images, *e.g.*, some dataset for segmentation tasks. In detail, since there would be numerous clusters within each image, masking specific clusters with informed masking may yield similar masks to random masking.

Acknowledgements

This work was supported by Samsung Electronics Co., Ltd (IO230414-05943-01, RAJ0123ZZ-80SD), by Youlchon Foundation (Nongshim Corp.), and by National Research Foundation (NRF) grants (No. 2021H1D3A2A03038607 / 50%, RS-2024-00336576 / 10%, RS- 2023-00222663 / 5%) and Institute for Information & communication Technology Planning & evaluation (IITP) grants (No. RS-2024-00353131 / 25%, RS-2022-II220264 / 10%), funded by the government of Korea.

References

[1] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. *arXiv:2005.00928*, 2020.

[2] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022.

[3] S. Atito, M. Awais, A. Farooq, Z. Feng, and J. Kittler. MC-SSL0.0: Towards multi-concept self-supervised learning. *arXiv:2111.15340*, 2021.

[4] S. Atito, M. Awais, and J. Kittler. SiT: Self-supervised vision transformer. *arXiv:2104.03602*, 2021.

[5] R. Bachmann, D. Mizrahi, A. Atanov, and A. Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022.

[6] W. G. C. Bandara, N. Patel, A. Gholami, M. Nikkhah, M. Agrawal, and V. M. Patel. AdaMAE: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *CVPR*, 2023.

[7] H. Bao, L. Dong, S. Piao, and F. Wei. BEiT: Bert pre-training of image transformers. *arXiv:2106.08254*, 2021.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[9] S. Cao, P. Xu, and D. A. Clifton. How to understand masked autoencoders. *arXiv:2202.03670*, 2022.

[10] H. Chen, W. Zhang, Y. Wang, and X. Yang. Improving masked autoencoders by learning where to mask. *arXiv:2303.06583*, 2023.

[11] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

[14] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo. PeCo: Perceptual codebook for bert pre-training of vision transformers. In *AAAI*, 2023.

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

[16] Y. Fang, S. Yang, S. Wang, Y. Ge, Y. Shan, and X. Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. In *ICCV*, 2023.

[17] C. Feichtenhofer, Y. Li, K. He, et al. Masked autoencoders as spatiotemporal learners. In *NeurIPS*, 2022.

[18] P. Gao, T. Ma, H. Li, Z. Lin, J. Dai, and Y. Qiao. MCMAE: Masked convolution meets masked autoencoders. In *NeurIPS*, 2022.

[19] R. Girdhar, A. El-Nouby, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. OmniMAE: Single model masked pretraining on images and videos. In *CVPR*, 2023.

[20] A. Gupta, S. Tian, Y. Zhang, J. Wu, R. Martín-Martín, and L. Fei-Fei. MaskViT: Masked visual pre-training for video prediction. *arXiv:2206.11894*, 2022.

[21] A. Gupta, J. Wu, J. Deng, and L. Fei-Fei. Siamese masked autoencoders. *arXiv:2305.14344*, 2023.

[22] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[23] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[25] Z. Hou, F. Sun, Y.-K. Chen, Y. Xie, and S.-Y. Kung. MILAN: Masked image pretraining on language assisted representation. *arXiv:2208.06049*, 2022.

[26] B. Huang, Z. Zhao, G. Zhang, Y. Qiao, and L. Wang. MGMAE: Motion guided masking for video masked autoencoding. In *ICCV*, 2023.

[27] Z. Huang, X. Jin, C. Lu, Q. Hou, M.-M. Cheng, D. Fu, X. Shen, and J. Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv:2207.13532*, 2022.

[28] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, A. Bursuc, K. Karantzalos, and N. Komodakis. What to hide from your students: Attention-guided masked image modeling. In *ECCV*, 2022.

[29] L. Kong, M. Q. Ma, G. Chen, E. P. Xing, Y. Chi, L.-P. Morency, and K. Zhang. Understanding masked autoencoders via hierarchical latent variable models. In *CVPR*, 2023.

[30] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[31] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo. Predicting what you already know helps: Provable self-supervised learning. In *NeurIPS*, 2021.

[32] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng. SemMAE: Semantic-guided masking for learning masked autoencoders. In *NeurIPS*, 2022.

[33] X. Li, Y. Ge, K. Yi, Z. Hu, Y. Shan, and L.-Y. Duan. mc-BEiT: Multi-choice discretization for image bert pre-training. In *ECCV*, 2022.

[34] Z. Li, Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang, et al. MST: Masked self-supervised transformer for visual representation. In *NeurIPS*, 2021.

[35] F. Liang, Y. Li, and D. Marculescu. SupMAE: Supervised masked autoencoders are efficient vision learners. *arXiv:2205.14540*, 2022.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[37] Y. Liu, S. Zhang, J. Chen, K. Chen, and D. Lin. PixMIM: Rethinking pixel reconstruction in masked image modeling. *arXiv:2303.02416*, 2023.

[38] Z. Liu, J. Gui, and H. Luo. Good helper is around you: Attention-driven masked image modeling. In *AAAI*, 2023.

[39] N. Madan, N.-C. Ristea, K. Nasrollahi, T. B. Moeslund, and R. T. Ionescu. Cl-mae: Curriculum-learned masked autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2492–2502, 2024.

[40] J. Mun, M. Shin, G. Han, S. Lee, S. Ha, J. Lee, and E.-S. Kim. BaSSL: Boundary-aware self-supervised learning for video scene segmentation. In *ACCV*, 2022.

[41] J. Pan, P. Zhou, and S. Yan. Towards understanding why mask-reconstruction pretraining helps in downstream tasks. *arXiv:2206.03826*, 2022.

[42] N. Park, W. Kim, B. Heo, T. Kim, and S. Yun. What do self-supervised vision transformers learn? *arXiv:2305.00729*, 2023.

[43] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[44] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[45] Y. Shi, N. Siddharth, P. Torr, and A. R. Kosiorek. Adversarial masking for self-supervised learning. In *ICML*, 2022.

[46] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(12):583–617, 2002.

[47] Z. Tong, Y. Song, J. Wang, and L. Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022.

[48] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

[49] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[50] H. Wang, K. Song, J. Fan, Y. Wang, J. Xie, and Z. Zhang. Hard patches mining for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10375–10385, 2023.

[51] H. Wang, Y. Tang, Y. Wang, J. Guo, Z.-H. Deng, and K. Han. Masked image modeling with local multi-scale reconstruction. In *CVPR*, 2023.

[52] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao. VideoMAE v2: Scaling video masked autoencoders with dual masking. In *CVPR*, 2023.

[53] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, Y.-G. Jiang, L. Zhou, and L. Yuan. BEVT: BERT pretraining of video transformers. In *CVPR*, 2022.

[54] Y. Wang, Z. Pan, X. Li, Z. Cao, K. Xian, and J. Zhang. Less is more: Consistent video depth estimation with masked frames modeling. In *ACM MM*, 2022.

[55] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley, and D. Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022.

[56] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022.

[57] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[58] J. Xie, W. Li, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy. Masked frequency modeling for self-supervised visual pre-training. *arXiv:2206.07706*, 2022.

[59] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu. SimMIM: A simple framework for masked image modeling. In *CVPR*, 2022.

[60] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. VideoGPT: Video generation using vq-vae and transformers. *arXiv:2104.10157*, 2021.

[61] Q. Zhang, Y. Wang, and Y. Wang. How mask matters: Towards theoretical understandings of masked autoencoders. In *NeurIPS*, 2022.

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

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

Appendix

A Method Elaboration

Detailed Reasoning for Our Method. As discussed in Section 2, the true shared information c exists for the entire token set X, which is equivalent to statistical dependency among the patches in X. With training, MAE learns to estimate this high-level latent variable $\hat{c}$, which reflects the context of the entire image. Let us denote by s_m and s_v for information specific to masked out patches X_m and visible patches X_v respectively, *e.g.*, positional embeddings.

Since MAE cannot access X_m during training, the decoder is forced to reconstruct X_m via 1) simple interpolation using visible tokens, or 2) estimated statistical dependency among the entire tokens, *i.e.*, $\hat{c}$. As shown in Figure I, simple interpolation means reconstructing X_m mainly with X_v and s_v, which is not directly related to X_m, leading to poor reconstruction result. However, due to the reconstruction loss, MAE is forced to improve the reconstruction quality, establishing high-level information $\hat{c}$ and performing the reconstruction based on it. As a result, at some moment, the encoder starts to map the visible tokens X_v to estimated shared information $\hat{c}$ for the whole token set X, and decoder exploits this hierarchical information to reconstruct the low-level information, *i.e.*, the raw RGB pixels of X_m. This process is verified in Figure 4 in the main manuscript.

Figure I: Hierarchical latent variable model framework [29]. Assuming high-level shared information c exists among the whole tokens, MAE encoder learns to estimate $\hat{c}$ from X_v to reconstruct raw pixels of X_m. Here, shared information is equivalent to statistical dependency inside X. s_m and s_v stand for information specific to X_m and X_v, respectively. Dotted line indicates potential dependency.

Moreover, connecting this logic to our discovery in Section 3.2, we claim that this unknown c conceptually corresponds to pattern-based patch clustering information. In other words, considering the pattern-based patch clustering in MAE (as verified in Section 3), it suggests that MAE clusters the patches and builds corresponding high-level variable containing $\hat{c}$ for each patch cluster.

In summary, MAE learns to construct the latent variables for each potential patch cluster. However, considering the fact that MAE learns *relevance* among the patches from the extremely early stages in pre-training process (Section 3.3), it can be inferred that MAE with naive random masking is actually revisiting *key dissimilarities* in X, which exists between easily separable patches, every epoch wasting large portion of its training resources. Especially, when it comes to bi-partitioning (which is the simplest form of *key dissimilarities*), MAE learns it from the very early epochs as verified in Fig. 3a.

Based on this reasoning, we can enforce MAE to focus on learning hardly distinguishable tokens by guiding MAE to skip revisiting *key dissimilarities* by injecting the information about it as input. We can inject this information via informed masks, which possess *key dissimilarities* by intensively masking one of the bi-partitioned clusters, leading MAE to assign most of the training resource to learning relatively vague patch clusters in masked out patch sets.

Qualitative Analysis. As discussed in Section 4, our method generates informed masks by itself without using any external model or requiring additional information. Recall that MAE generates informed masks after $T = 50$ epochs of training. Figure II compares our informed masking with and without the hint tokens to the random masking. It also illustrates the bi-partitioned clusters extracted from MAE itself after 51 epochs, which are used for the internal generation of the informed masks. We observe in these examples that our relevance-score-based masking (Section 4) guarantees to fully mask out the target cluster even when the bi-partitioning is not perfect. For example, the target cluster in (e) consists of the portion of *house* and *sky*, but our method fully masks out the patches composing the *house* in the image. Similar results can be found in (j) and (k). Also, even when the foreground

Figure II: Qualitative examples of informed masking on ImageNet training set. Based on our method, informed masks are generated after 51 epochs of pre-training with a hint ratio of 0.05. Results clearly show that MAE in early training steps provides appropriate bi-partitioning information and successfully creates informed mask without using external models or additional information. We also note that, our similarity-score-based masking strategy yields robust informed mask even in the case when the bi-partitioning is imperfect.

is not clearly distinguished due to the barely discernible patterns as in (i) and (l), we see that our approach still fully masks out the object. The success of relevance score strongly indicates that patch vectors are hierarchically clustered based on their visual patterns, as they are masked out in the order of pattern similarity with the mean patch vector.

We confirm from the examples that even in early epochs, MAE is able to appropriately bi-partition the image, which means it has already learned to discriminate the image into two clusters. We also find that most of the examples are bi-partitioned into foreground and background, since the similarity edges between these two groups tend to have the weakest values. In summary, although MAE in the early epochs does not promise to provide perfectly discriminated object-centric cluster from the image, our proposed approach robustly builds object-centric masks through the introduction of the relevance score.

B Token Relations

Patch Clustering in Projected Latent Space. Figure III illustrates the patch clusters on a few examples and their t-sne plots. We consider a graph $G = (V; E)$ for the given image, where V and E correspond to patches and edges between them weighted by M in Eq. (2), respectively. From this graph, we repeatedly apply Normalized Cut [44] to remove edges with the lowest relevance until the graph is split into a predefined number (K) of clusters. We clearly see that tokens with similar visual patterns (color, texture) are 1) grouped together as the same patch cluster (2nd row) and 2) embedded closely in the latent space (last row). Apparent discrimination in the representation space supports the patch-level clustering.

Enhanced Feature Separability of MAE with Our Method. Based on our analysis on embedding space suggested in Equation 3, we compare the vanilla MAE and our method in the aspect of feature separability with 800 epochs of training in Table I. The results indicate that our method shows more diversified feature space via higher feature variance and similarity variance, aligned well with the analysis in Section 5.4.

Table I: Feature variance ($_F$) and similarity variance ($_S$).

Feature	E[$_F$]	E[$_S$]
MAE [22]	0.082	0.075
Ours	0.096	0.079

KL Divergence of All Layers in MAE. We additionally provide KL divergence (KLD) of token relations for all layers in MAE as an extension of Section 3.3. For the decoder, we use token relations with the intact input, i.e., $D([E(X)])$, for the criterion distribution in the KLD (Equation 4). In other

Figure III: Illustrations of patch clusters learned by MAE. (a) Input images. (b) Similarity-based patch clusters. (c) t-sne plots of the patch embeddings.

words, we compare the token relations from each epoch with masked inputs to the token relations from the last epoch with intact inputs. Due to this setting, KLD with decoder does not converge to zero at the final epoch in Figure IV.

As shown in Figure IV, all layers but the first one in the decoder drastically converge at the early epochs with both of M than A. Encoder ($E(X)$) layers are much stabler and converge faster than decoder ($D([E(X_v); m])$) layers due to the difference in the amount of given information. Also, since the cosine similarity scores M directly compare the similarity among the tokens, strong convergence of M supports the observation that MAE intrinsically learns the patch-level clustering.

KLD of the attention scores in the first encoder layer is low at the first epoch, which implies that it learns homogeneous attention map rather than random values as discussed in Section 5.4. The first layer of the decoder shows high KLD with the attention scores along with the training, because 1) the mask tokens are not contextualized yet (that is, mask token vectors does not represent the masked out patches at all), and 2) the index of each mask token is randomly selected for every epoch. On the other hand, KLD with the similarity scores decreases along the epochs, because the similarity score matrix is calculated after the contextualization. This suggests that even a single first layer in decoder has ability to properly exploit it from the encoder to discriminate the patches although it is weaker than the later layers.

Figure IV: KL divergence of the token relations between the final and intermediate epochs. Layer numbers are displayed in the legend. All the layers but the first one in decoder show drastic decrement of (a) similarity score and (b) attention score at early epochs. The convergence speed and the final converged values vary in layers.

Figure V: Bi-partitioning performance of various models. MAE, MoCo and ViT show different trends of bi-partitioning performance in both of (a) similarity score and (b) attention score.

Further Experiments on ViT [15] and MoCo [23]. We provide bi-partitioning performance and KL divergence of token relations of ViT and MoCo for better understanding on our metrics in Figure V. We display the result of MAE encoder together for comparison. Before delving into the analysis, we note that the result of this experiment with ViT and MoCo is irrelevant to our main claims since ViT and MoCo do not learn patch clustering.

As MoCo yields homogeneous attention map [42] resulting in simple form of embedding space, *e.g.*, main object cluster and background cluster, the result of MoCo in Figure V indicates that the last epoch of MoCo has provided properly bi-partitioned patch groups. Consistent gap between mean inter-cluster ($_{inter}$) and mean intra-cluster ($_{intra}$) edge weights of similarity score matrix M and attention score matrix A of MoCo supports this claim.

Unlike MAE or MoCo, embedding space of ViT does not guarantee to provide appropriate bi-partitioning results. As a result, in Figure V, although the similarity score matrix M enlarges the gap between $_{inter}$ and $_{intra}$, the attention score matrix A increases the $_{inter}$ rather than $_{intra}$. This hardly interpretable pattern implies that the pseudo-ground truth for bi-partitioned patch groups generated at the last epoch is unstable or even incorrect.

In summary, only MAE explicitly shows its ability to clearly recognize key dissimilarities among the tokens, *i.e.*, bi-partitioning information, from the extremely early stage of pre-training, and consistently escalates the gap between $_{inter}$ and $_{intra}$.

Figure VI: KL divergence of token relations of various models. MoCo and ViT show weaker convergence of token relations in both of (a) similarity score and (b) attention score.

Figure VI shows the KL divergence of token relations from ViT and MoCo. Compared to the result of MAE in Figure IV, both ViT and MoCo reveal gradual convergence of token relations and some layers exhibit their unstable convergence. Again, as ViT and MoCo do not learn patch-clustering, the experiment results of ViT and MoCo are off-topic to the main stream of our work.

Figure VII: Qualitative comparison on ImageNet validation set. Patches are discriminated in more ne-grained manner with our method. More diverse and ner patch clusters constructed in foreground verify our hypothesis that intensive masking on speci c cluster leads to establish more diverse high-level latent variables.

C Qualitative Results

We provide more qualitative examples of patch clustering compared to vanilla MAE in Figure VII, where we see that images are segmented into K clusters in unsupervised manner. Successful segmentation from our recursive graph-cut suggests that features are hierarchically discriminated in the embedding space. Our method clearly shows more accurately clustered patches based on their pattern and also yields tighter boundary between the clusters for various types of images, i.e., object-centered images and those containing higher portion of background.

D Analysis on Ablation Studies

As displayed in Table II, our ablation study on layer selection for embedding extraction veri es the hypothesis on it (See Section 3), while showing the minor effect on model performance relative to other factors. Especially, the last layer of the decoder shows higher performance than the early or intermediate layers of the encoder. Since the decoder possesses the patch cluster information constructed through the entire encoder layers, it may have more appropriate bi-partitioning quality than using a few early encoder layers, e.g., layer 3 or layer 7.

To analyze the reason for the minor effect on layer selection, we display the examples of informed masks generated with bi-partitioned patch cluster from each layer in Figure VIII.

Table II: Ablation studies on various factors. The default is highlighted in gray.

Layer	Target cluster	Hint strategy	Masking ratio	Linear probing
Enc 3	Object	Random	0.75	62.3
Enc 7	Object	Random	0.75	62.4
Dec 8	Object	Random	0.75	62.7
Enc 11	Object	S_i-based	0.75	62.5
Enc 11	Object	No hint	0.75	52.3
Enc 11	Background	Random	0.75	61.1
Enc 11	Alternate	Random	0.75	61.6
Enc 11	Object	Random	0.6	62.9
Enc 11	Object	Random	0.9	61.4
Enc 11	Object	Random	0.75	62.9

In Figure VIII, we nd that the later layer of the encoder provides the most accurate bi-partitioning result compared to others. However, in spite of the improper patch clustering, each layer can build plausible informed mask (and often proper) based on our similarity-score-based masking strategy. With a simple image, e.g., (a), all layers are able to properly bi-partition the image leading to fully mask out the main object. With more complex images like (b), (c), (e) and (f), whether 1) the bi-partitioned cluster contains a mixture of foreground and background or 2) only some patches of

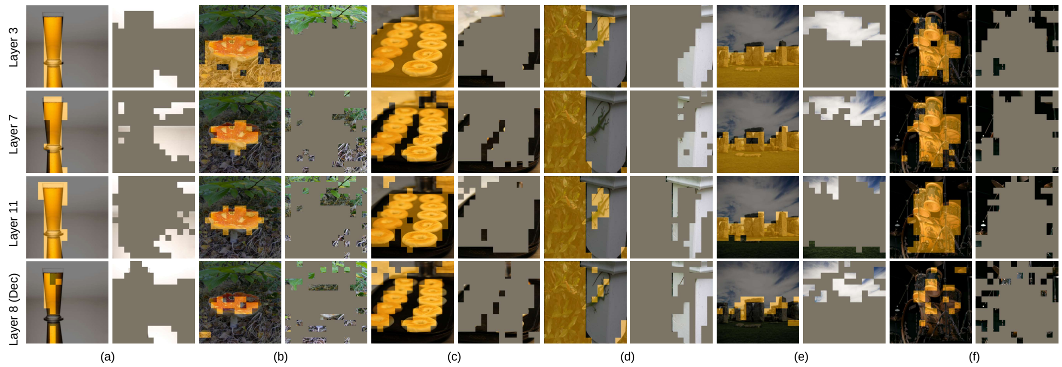

Figure VIII: **Comparison of the Quality of the informed masks generated from different layers.** Each example is denoted by the index of the original image in Figure II. Although early layers of the encoder and the last layer of the decoder yield inappropriate bi-partitioning result, our similarity-score-based masking strategy robustly alleviates this issue, leading to minor difference in performance in the layer selection for generating informed mask.

the foreground are discriminated, our method stably constructs the proper informed mask, aligning with the result in Figure II. In example (d), when the layer 7 is used, we observe that object-centric masks are successfully generated since the pattern of *lizard* is similar to that of *plants*, despite a failure in bi-partitioning where the discriminated foreground captures only the *plants*, missing the *liazard*. Also, although the decoder hardly captures the entire shape of the foreground, it precisely discriminates the salient patches belonging to the main objects, as expected to generate more accurate informed mask than the early or intermediate encoder layers. We also note that using the last layer of the encoder yields similar performance to our default setting (*i.e.*, using the second last layer of the encoder) which could be preferred for its simplicity.

As shown in the second group in the Table II, it is essential to provide hint tokens for successful training. As displayed in Table III, the training process results in too high loss without the hint tokens, while it is appropriately alleviated with them. This is because it is fundamentally impossible for the model to reconstruct the whole foreground without any visible tokens belong to it. In the aspect of patch clustering, MAE would lose an opportunity to construct high-level latent variables *i.e.*, shared information, for the clusters specific to the foreground when trained without hint tokens.

Table III: **Reconstruction loss (MSE)** with 400 pre-training epochs according to each training method.

Model	MAE [22]	Ours (no hint)	Ours (with hint)
Loss	0.41	0.64	0.56

In addition to the ablation studies in Table 3, we also consider 1) the target cluster to be masked out and 2) masking ratio in the third and fourth group in Table II, respectively. Object-centric informed masking leads to better performance compared to background-centric masking or alternately masking foreground and background along the epochs, supporting our choice of object-centric masking strategy in Section 4. For the masking ratio, although masking less regions (0.6) yields the same linear probing performance to the default one (0.75), it is recommended to set masking ratio to 0.75 for more efficient training cost.

E Comparison with Various MAEs

We compare the performance across various MAEs in Table IV, grouping the models based on the incorporation of external training costs. For models that we were able to reproduce the results (AMT [38], HPM [50]), we report the reproduced results. For SemMAE [32], we refer the performance as reported in the paper. For CL-MAE [39], we report only the training time, as its reproduction is difficult due to the high training cost, and the results reported in the respective paper are not directly comparable due to different experimental settings.

Table IV: **Comparison with additional MAEs in terms of linear probing performance.** MAEs that utilize external resources or additional parameterized modules are highlighted in gray, indicating that they are not included as baselines for a fair comparison. The training time is reported as a relative value to the MAE, training for 400 epochs. The results show that our method matches the performance of other MAEs with the same or even less training cost. Our method requires only about one more step to generate masks, which empirically increases the pre-training time about 0.25% for training 400 epochs, *i.e.,* 1.0025 training time compared to the vanilla MAE.

Method	# of params.	Pre-train epochs	Linear probing	Training time
Baselines using external pre-trained model				
SemMAE [32]	112M	800	68.7	6.3
MAE [22]	112M	800	63.8	2
MAE [22]	112M	1600	68.0	4
Ours	112M	800	**65.9**	2
Ours	112M	1600	**68.7**	4
Baselines using additional module				
HPM [50]	138M	400	63.2	1.5
CL-MAE [39]	148M	400	-	6
*Baselines **without** external resource or additional module*				
AMT [38]	112M	400	61.7	1
MAE	112M	400	61.4	1
Ours	112M	400	**62.9**	1

F Extended Training

We conduct extended pre-training sessions and report linear probing performance on ImageNet-1K along the training epochs in Table V. Our method consistently brings sustained performance gain after considerable length of training, *i.e.*, for 1600 epochs.

Table V: Linear probing with ImageNet-1K

Pre-training epochs	200	400	800	1600
MAE [22]	53.9	61.4	63.8	68.0
Ours	**54.4**	**62.9**	**65.9**	**68.7**

G Computing Resources

We conduct experiments on 8 NVidia A6000 GPUs (48GB) and it takes ~2.5 days on pre-training for 400 epochs. For 1600 epochs of pre-training, it takes about 10 days.

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: Our main contribution is discovery of the internal operations of MAE and a novel method to expedite its training based on the discovery. These points are clearly mentioned in the abstract and introduction.

   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: We mention limitations of our work in Section 7.

   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: We do not claim any theoretical result or contribution.

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: We provide the detailed algorithm and hyperparameter settings in Section 4 and Section 5.

   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: We plan to publicly provide our code used in this paper.

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: Experimental settings and details are elaborated in Section 5.

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 due to the expensive computational cost.

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 provide computational cost for pre-training in Appendix G.

   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: We strictly follow the ethics guideline with this paper.

   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: [NA]

    Justification: This paper mainly studies the internal operations conducted by the Masked Autoencoder model, and proposes how to expedite its training. As far as we see, there is no direct negative societal impacts from this work.

    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: Our proposed method is mainly about expediting an existing self-supervised model, Masked Autoencoder. Besides general risks of machine learning models, *e.g*, trained on biased data, this work does not impose any high risk for misuse as far as we concern.

    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: We respectfully cited all assets we use, following the license and terms of use properly.

    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: [Yes]

    Justification: We provide detailed documentation regarding the code and trained models to be released.

    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: Our work does not involve any human subject or crowdsourcing.

    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: Our work does not involve any human subject.

    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.