447

Appendix

A Method Elaboration

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

Since MAE cannot access $\mathcal{X}_m$ during training, the decoder is forced to reconstruct $\mathcal{X}_m$ via 1) simple interpolation using visible tokens, or 2) estimated statistical dependency among the entire tokens, *i.e.*, $\hat{\mathbf{c}}$. As shown in Figure I, simple interpolation means reconstructing $\mathcal{X}_m$ mainly with $\mathcal{X}_v$ and $\mathbf{s}_v$, which is not directly related to $\mathcal{X}_m$, leading to poor reconstruction result. However, due to the reconstruction loss, MAE is forced to improve the reconstruction quality, establishing high-level information $\hat{\mathbf{c}}$ and performing the reconstruction based on it. As a result, at some moment, the encoder starts to map the visible tokens $\mathcal{X}_v$ to estimated shared information $\hat{\mathbf{c}}$ for the whole token set $\mathcal{X}$, and decoder exploits this hierarchical information to reconstruct the low-level information; *i.e.*, the raw RGB pixels of $\mathcal{X}_m$. This process is verified in Figure 4 in the main manuscript.
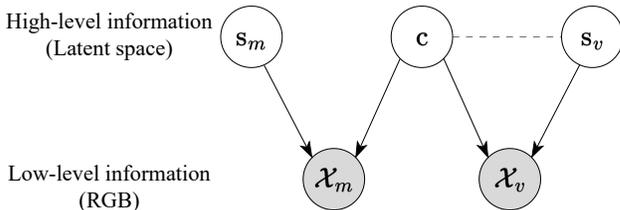
Figure I: **Hierarchical latent variable model framework [29].** Assuming high-level shared information $\mathbf{c}$ exists among the whole tokens, MAE encoder learns to estimate $\hat{\mathbf{c}}$ from $\mathcal{X}_v$ to reconstruct raw pixels of $\mathcal{X}_m$. Here, shared information is equivalent to statistical dependency inside $\mathcal{X}$. $\mathbf{s}_m$ and $\mathbf{s}_v$ stand for information specific to $\mathcal{X}_m$ and $\mathcal{X}_v$, respectively. Dotted line indicates potential dependency.

Moreover, connecting this logic to our discovery in Section 3.2, we claim that this unknown $\mathbf{c}$ 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 $\hat{\mathbf{c}}$ 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 $\mathcal{X}$, 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 $T \approx 50$ 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

i
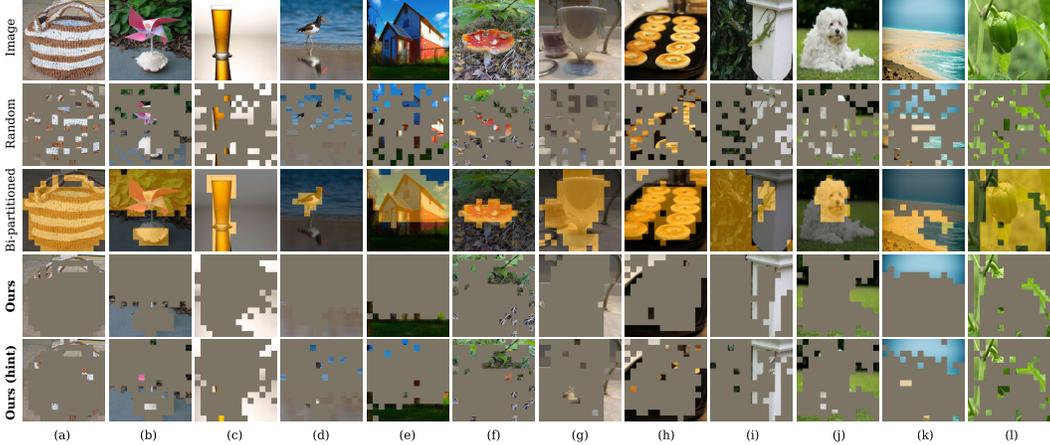
Figure II: **Qualitative examples of informed masking on ImageNet training set.** Based on our method, informed masks are generated after 51 epochs of pre-training with a hint ratio of 0.05. Results clearly show that MAE in early training steps provides appropriate bi-partitioning information and successfully 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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for the given image, where $\mathcal{V}$ and $\mathcal{E}$ correspond to patches and edges between them weighted by $\mathbf{M}$ in Eq. (2), respectively. From this graph, we repeatedly apply Normalized Cut [42] 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.

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.*, $D([E(\mathbf{X})])$, for the criterion distribution in the KLD (Equation 4). In other 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 $\mathbf{M}$ than $\mathbf{A}$. Encoder ($E(\mathbf{X})$) layers are much stabler and converge faster than decoder ($D([E(\mathbf{X}_v); \mathbf{m}])$) layers due to the difference in the amount of given information. Also, since the cosine similarity scores $\mathbf{M}$ directly compare the similarity among the tokens, strong convergence of $\mathbf{M}$ supports the observation that MAE intrinsically learns the patch-level clustering.

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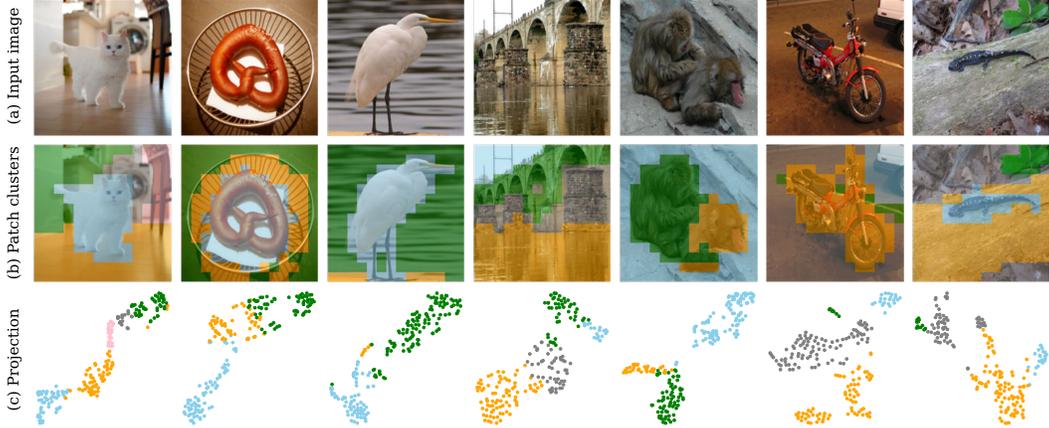
Figure III: **Illustrations of patch clusters learned by MAE.** (a) Input images. (b) Similarity-based patch clusters. (c) t-sne plots of the patch embeddings.

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 ĉ from the encoder to discriminate the patches although it is weaker than the later layers.

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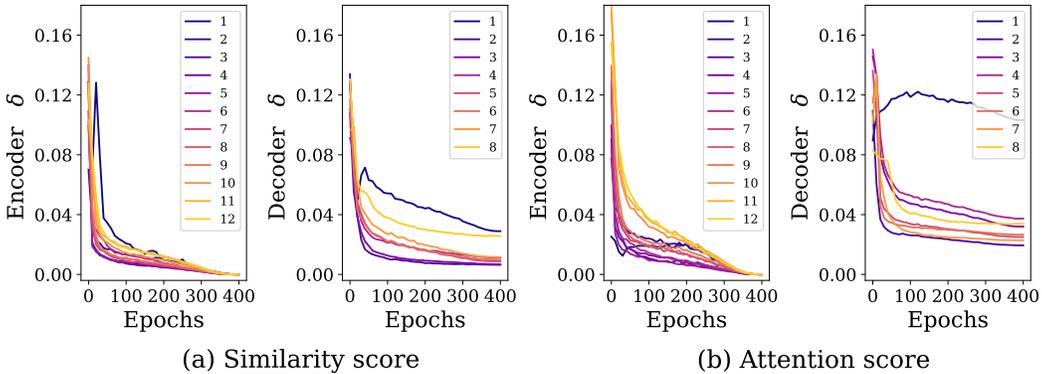
Figure IV: **KL divergence of the token relations between the final and intermediate epochs.** Layer numbers are displayed in the legend. All the layers but the first one in decoder show drastic decrement of (a) similarity score and (b) attention score at early epochs. The convergence speed and the final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

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 [40] 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 (μ_{inter}) and mean intra-cluster (μ_{intra}) edge weights of similarity score matrix $\mathbf{M}$ and attention score matrix $\mathbf{A}$ of MoCo supports this claim.

iii

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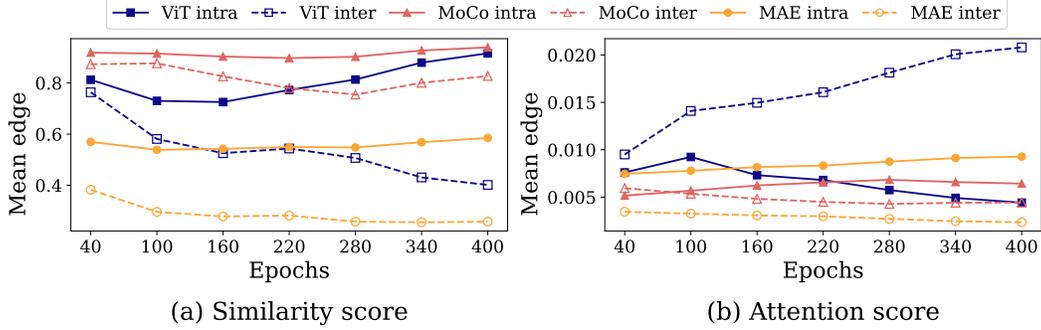
Figure V: **Bi-partitioning performance of various models.** MAE, MoCo and ViT show different trends of bi-partitioning performance in both of (a) similarity score and (b) attention score.

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 $\mathbf{M}$ enlarges the gap between μ_{inter} and μ_{intra}, the attention score matrix $\mathbf{A}$ increases the μ_{inter} rather than μ_{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 μ_{inter} and μ_{intra}.

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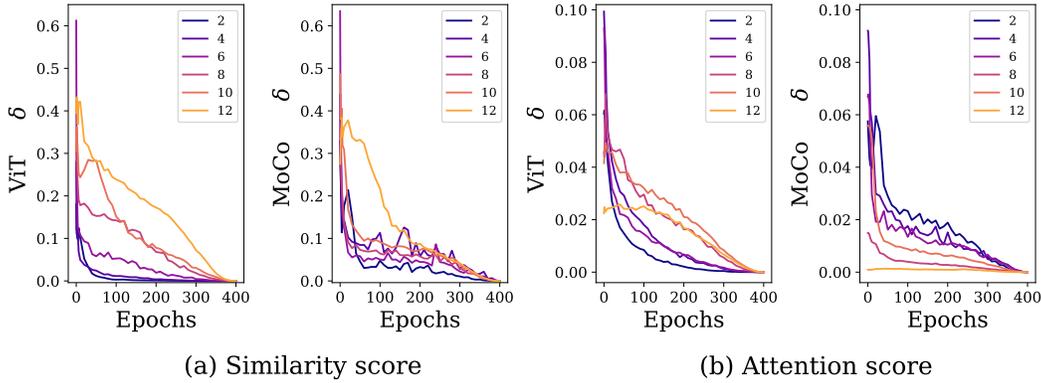
Figure VI: **KL divergence of token relations of various models.** MoCo and ViT show weaker convergence of token relations in both of (a) similarity score and (b) attention score.

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.

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.

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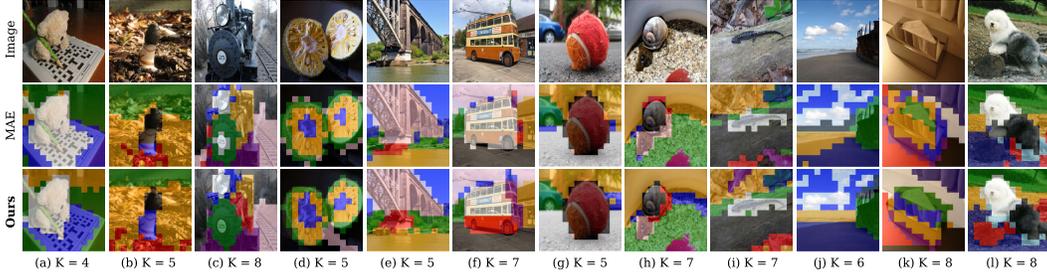
Figure VII: **Qualitative comparison on ImageNet validation set.** Patches are discriminated in more fine-grained manner with our method. More diverse and finer patch clusters constructed in foreground verify our hypothesis that intensive masking on specific cluster leads to establish more diverse high-level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

D Analysis on Ablation Studies

As displayed in Table I, our ablation study on layer selection for embedding extraction verifies 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 I: **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	Background	Random	61.1
Enc 11	Alternate	Random	61.6
Enc 11	Object	S_i based	62.5
Enc 11	Object	No hint	52.3
Enc 11	Object	Random	**62.9**

In Figure VIII, we find 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 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.

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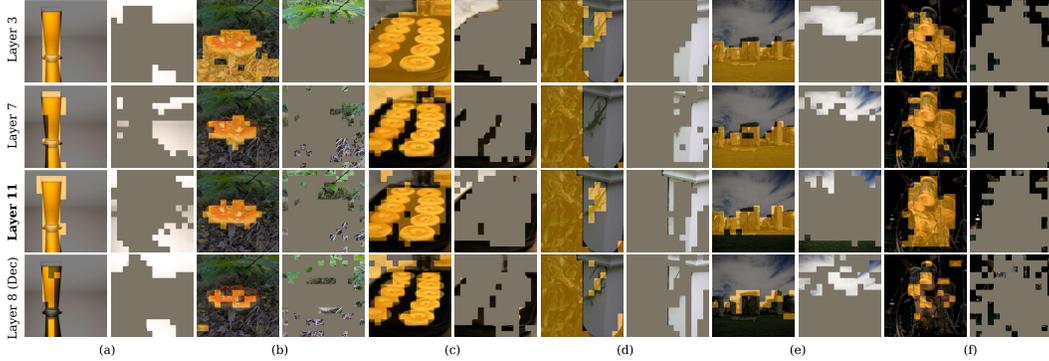
Figure VIII: **Comparison of the Quality of the informed masks generated from different layers.** Each example is denoted by the index of the original image in Figure II. Although early layers of the encoder and the last layer of the decoder yield inappropriate bi-partitioning result, our similarity-score-based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

E   Extended Training

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

Table II: 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**

F   Compute 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.

vi