Table 7. Fine-tuning setting.

config	value
optimizer	AdamW [39]
base learning rate	$5e-4$
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$ [9]
layer-wise lr decay [2, 13]	0.65
batch size	1024
learning rate schedule	cosine decay [38]
warmup epochs [24]	5
training epochs	100
cutmix [63]	1.0
drop path	0.1
mixup [64]	0.8
weight decay	0.05
label smoothing [53]	0.1
augmentation	$\mathrm{RandAug}(9, 0.5)$ [14]

A. Implementation details of DPPMask

Suppose we add i into the subset $Y_g \cup \{j\}$. From Eq. 10, we have

$$\begin{bmatrix} V & 0 \\ c_j & d_j \end{bmatrix} c_i'^{\top} = L_{Y_g \cup \{j\}, i} = \begin{bmatrix} L_{Y_g, i} \\ L_{ji} \end{bmatrix}, \tag{13}$$

where

$$c_i' = \begin{bmatrix} c_i & (L_{ji} - \langle c_j, c_i \rangle) / d_j \end{bmatrix} \doteq \begin{bmatrix} c_i & e_i \end{bmatrix}. \tag{14}$$

For updating d_i, we have

$$\begin{aligned} d_i'^2 &= L_{ii} - \|c_i'\|_2^2 \\ &= L_{ii} - \|c_i\|_2^2 - e_i^2 \\ &= d_i^2 - e_i^2. \end{aligned} \tag{15}$$

With Eq. 14 and Eq. 15, we can update d incrementally.

B. ImageNet-100 setting

We follow the original MAE [26] experiment setting, except for the learning rate of fine-tuning task. We list our fine-tuning parameters in Tab. 7. For iBOT [67], we train a ViT-small backbone with 100 epochs. We change the block mask strategy with random, and set masking ratio to 70% with 5% variation. We list our fine-tuning parameters in Tab. 8

C. Greedy approximation performance

We examine the approximation performance with respect to original DPPs. We run each setting with one entire epoch and report the mean time cost of each iteration. As Tab. 9 shows, the greedy approximation achieved 10x faster than original DPPs, which makes it possible to fit into a GPU training loop.

Table 8. iBOT pre-train setting.

config	value
learning rate	$5e-4$
teacher momentum [2]	0.996
teacher temp	0.07
warmup teacher temp epochs [2]	30
out dim	8192
local crops number	10
global crops scale	$[0.25,\ 1]$
local crops scale	$[0.05,\ 0.25]$
mask ratio	0.7
mask ratio var	0.05
masking prob	0.5

Table 9. Training efficiency of DPPMask

Method	DPP	DPP (Greedy approximation)	Random
Training speed	2.7442s/it	0.2609 s/it	0.2129 s/it

D. More discussion about relative works

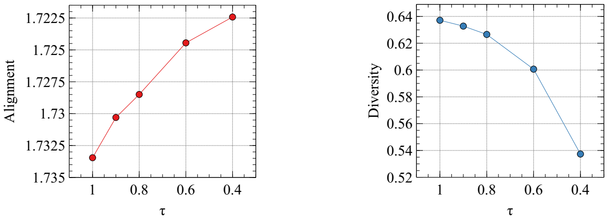
Our method is different from recent masking strategies. Recent strategies can roughly divide into two lines of work, learning-based and attention-based.

Learning-based strategies include ADIOS [52] and SemMAE [35]. They both need extra learning parameters. ADIOS trains a network to propose masks adversarially, in order to find out more meaningful masks for MIM tasks. However, the semantic meaningful masks proposed by the network are hard to predict, which is less explainable. SemMAE separates the mask learning process from pre-training and makes the training process into two stages. The type of masks they learned are more like semantic parts, such as heads, arms, etc. However, as the semantics varies in images, the number of classes is hard to define, therefore weakening the application of such methods.

Attention-based strategies include AttMask [29] and AMT [25]. This line of work selects patches according to the attention map. Despite different policies to manipulate the attention map, both intend to retain some patches with high attention scores to give a ”hint” to the model as such patches are more likely to have more semantics. Note this policy aligns with our ideas. However, they do not associate it with misalignment problems in MIM, thus leading to an inferior policy. Furthermore, attention maps require an extra forward pass to compute, which brings more computation.

E. Alignment versus diversity

As we discussed in 3.1, DPPMask aims to purge the training pairs that are polluted by misalignment problems. However, the network also needs irrelevant information between different masks to perform feature learning which can be measured as the variance of sampled masks. Here, we



(a) Alignment

(b) Diversity

Figure 7. Alignment and diversity of different τs, $\tau = 1$ indicate random masking.

show that DPPMask achieves the adjustment between alignment and diversity. We first obtain the original semantic representation by feeding the network with unmasked images and saving the *cls* token. Then, we obtain the masked semantic representation by saving the *cls* tokens of masked images under different masking strategies. We compute the L2 distance between masked semantics and original semantics to illustrate the alignment of different masking strategies. For masked semantic representation, we run 5 independent trails and compute the L2 distance between each trail to illustrate the diversity of different masking strategies. As shown in 7, random masking has the most diversity, but it also suffers from the misalignment problem, i.e. the farthest distance between masked semantics and original semantics 7a. As the τ decrease, the distance between masked semantics and original semantics has been reduced, indicating more alignment to the original semantics. However, a lower τ cause less diversity of different masks, which can purge some useful training pairs and is not helpful for feature learning 7b.

F. Broader impact

Despite the eye-catching performance of MIM algorithms, what makes a good mask for MIM tasks still remains unclear. DPPMask provides a possible answer to this question. By analog to the *InfoMin* principle of contrastive learning. We conclude two properties of MIM. Masked images should retain the original semantics while minimizing shared information from different masks. Minimizing shared information can be achieved by setting a high mask ratio, while how to retain the original semantics is a non-trivial problem. Furthermore, DPPMask also models the probability-of-co-occurrence of each patch and thus can serve as a potential tool to study the relationship between such two properties.

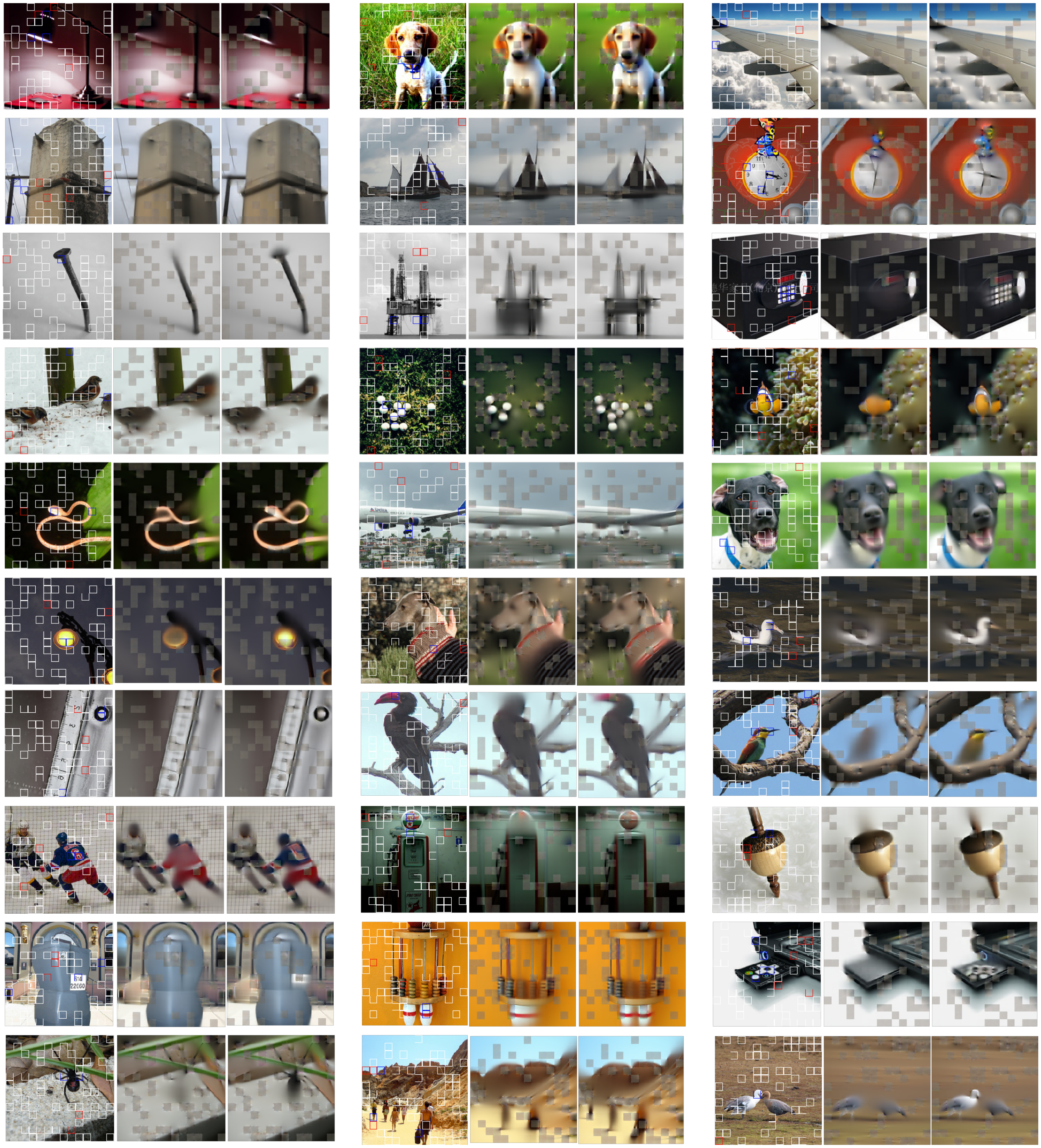


Figure 8. More qualitative samples. Each triplet indicates the original image (right), reconstruction result with random sampling (middle), and DPPs sampling (left).