MIXMASK: REVISITING MASKED SIAMESE SELF-SUPERVISED LEARNING IN ASYMMETRIC DISTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in self-supervised learning integrate *masked modeling* and *Siamese Networks* into a single framework to fully reap the advantages of both the two techniques. However, these approaches simply inherit the default loss design from previous siamese networks and ignore the distance change after employing *masking operation* in the frameworks. In this paper, we propose a filling-based masking strategy called *MixMask* to prevent information loss in vanilla masking method due to the randomly erased areas in an image. We further introduce a dynamic loss function design with soft distance to adapt the integrated architecture and avoid mismatches between transformed input and objective in *Masked Siamese ConvNets (MSCN)*. The dynamic loss distance is calculated according to the mix-masking scheme. Extensive experiments are conducted on various datasets of CIFAR-100, Tiny-ImageNet and ImangeNet-1K. The results demonstrate that the proposed framework can achieve better accuracy on linear evaluation and semi-supervised learning, which outperform the state-of-the-art MSCN by a significant margin. We also show the superiority on downstream tasks of object detection and segmentation. Our source code will be publicly available.

1 INTRODUCTION

Self-supervised learning is a widely used paradigm to learn representations from input data without human annotated labels. In computer vision domain, it has shown superior performance in many tasks, such as classification, detection, segmentation, etc. A popular self-supervised learning framework is a *Siamese Network* with two branches. A similarity loss (Grill et al., 2020; Chen & He, 2021), contrastive loss (Oord et al., 2018; He et al., 2020; Chen et al., 2020a) or distillation loss (Caron et al., 2021) is employed to calculate the distance of the two branches. Recently, *masked image modeling* (He et al., 2022; Bao et al., 2021; Assran et al., 2022) has emerged and proven to be an effective approach to learn useful representation. To fully leverage the advantages from both the *masked design* and *Siamese Networks*, Masked Siamese Networks (Assran et al., 2022; Jing et al., 2022) have been proposed for this purpose.

However, there are two main drawbacks in the *Masked Siamese Networks* that are usually overlooked. One is the default symmetric distance loss in the siamese networks. It brings relatively low-influence when both of the branches are masked. However, it causes a negative effect inevitably when only one branch is masked in the architecture. Consequently, a dynamic loss design with soft distance is crucial to reflect the true distance between the two asymmetric branches in siamese networks. Another drawback in such an architecture is the information loss from erasing-based masking operation. It is straightforward that vanilla masking will drop semantic information from the input data and cannot be recovered by post-processing. For instance, if we mask 25% of image areas, 25% of information will be lost during training and it will encumber the training efficiency. This is different from masked image modeling (MIM) methods that will recover the masked regions by image reconstruction loss. Further, a masking scheme in MIM can reduce the computational cost, while in masked siamese networks this is absent. Thus, a better masking strategy is necessary to encourage models to learn better representations.

To address these two drawbacks, in this work we propose to use a filling-based masking strategy to avoid information loss by erasing operation. Specifically, instead of erasing random areas in an image solely, we will randomly select another image and fill its pixels to the masked areas.

Compared to the erasing-based strategy, this masking method contains more semantic information for the input space and the siamese networks can also exploit more to enhance the representation during training. Our ablation experiments clearly show that this is a better solution for the *Masked Siamese Networks*. We further propose a soft dynamic loss term to evaluate the similarity of two branches in Masked Siamese Networks adaptively. The soft distance is calculated by comparing the masking ratios of two branches and the resulting distance essentially reflects the semantic similarity between them. Since the masking strategy changes across different iterations, the corresponding soft similarity loss also changes accordingly for different iterations.

Comprehensive experiments are conducted on CIFAR-100, Tiny-ImageNet, and ImageNet-1K datasets. We integrate our method in multiple *Siamese ConvNets* such as MoCo, BYOL, SimCLR, and SimSiam. Our method improves the various baselines by remarkable margins on all datasets. We also examine our learned models on semi-supervised learning and downstream tasks object detection and segmentation with consistent improvement.

Our contributions in this work are as follows:

- We proposed a dynamic loss function design with soft distance to adapt the integrated *masking* and *Siamese* architecture, as well as to avoid mismatches between transformed input and objective in *Masked Siamese Networks*.
- We introduce a simple yet effective filling-based masking strategy to prevent information loss from input space for self-supervised Siamese ConvNets. The proposed masking is mutually complementary to the soft similarity loss.
- Extensive experiments are conducted on various datasets and *Siamese* frameworks to demonstrate the effectiveness of the proposed method. We also verify our approach on semi-supervised learning and object detection/segmentation downstream tasks.

2 Related Work

Self-supervised Learning. Self-supervised learning is a popular technique for representation learning whose key ingredient is the usage of huge amounts of unlabeled data. First self-supervised approaches in vision were based on pre-text tasks (Noroozi & Favaro, 2016; Gidaris et al., 2018; Sermanet et al., 2018; Zhang et al., 2016). A milestone in SSL was a simple contrastive learning framework, introduced by (Chen et al., 2020b), which utilized a siamese architecture together with an InfoNCE (Oord et al., 2018) contrastive loss. MoCo (He et al., 2020) employed a memory bank to store negative samples. Several methods indicated that the presence of negative pairs is not necessary. BYOL (Richemond et al., 2020) used asymmetric architecture with EMA where the online network is trying to predict the representations of the target network. SimSiam (Chen & He, 2021) provided a simple siamese framework which applied stop-gradient operation on one branch and an additional predictor step on the other. Recently self-supervised learning adopted the use of vision transformers (Dosovitskiy et al., 2020). DINO (Caron et al., 2021) utilized knowledge distillation together with vision transformers in a Siamese framework. (He et al., 2022) explored how Masked Autoencoders can be used for the problem of self-supervised representation learning in vision.

Masked Image Modeling. Masked Image Modelling (MIM) is a task whose goal is to learn useful representations by trying to reconstruct a masked image to its original view. Masked Autoencoder is a common model which can be used in a MIM framework. (He et al., 2022), propose a simple transformer-based masked autoencoder architecture that tries to reconstruct the original image using MSE loss. (Feichtenhofer et al., 2022) extended MAE for the case of video data. (Huang et al., 2022) introduced an approach that combines contrastive learning and masked autoencoder under the unified framework. Several works (Girdhar et al., 2022; Bachmann et al., 2022; Geng et al., 2022) have studied how to make masked autoencoders in the case of multimodal data.

Masked Siamese Networks. Advances in masked image modeling and siamese self-supervised learning posed a natural question of finding a way to combine these two techniques. Masked Siamese Networks were proposed in (Assran et al., 2022). MSN generates two views, an anchor and a target, and applies masking operation on the anchor branch. It further uses prototype cluster assignment to assign masked anchor representations to the same cluster as the unmasked target. Vision Transformer has been chosen as an encoder, which greatly combines with masking operation due to it underlying design which splits the original image into patches of a smaller size. In contrast, CNNs are less suitable for working with masked images as they operate on the pixel level instead of the patch level, and, thus, masking operation destroys the image continuity that CNNs heavily rely on. In (Jing et al., 2022) authors explore and provide guidelines on how to make a CNN-based siamese self-supervised learning framework compatible with masking.

3 APPROACH

In this section, we first introduce each component in our framework elaborately, including: (i) a filling-based masking strategy; (ii) the proposed dynamic loss formulation with soft distance to match the mix-masking scheme; (iii) permutation strategies if incorporated with other approaches. Then, we provide an overview of the proposed architecture comparing to the basic model. Finally, we discuss in detail the design principles, insights, and differences from other approaches.

3.1 PERSPECTIVE ON MASK STRATEGY

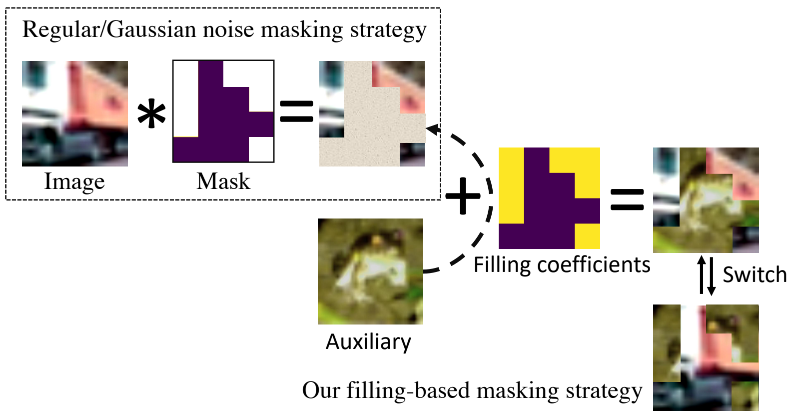

Figure 1: Illustration of the proposed filling-based masking strategy. The dashed box shows the regular and Gaussian noise masking strategies. A formal definition of a switch image in the case of inverse permutation is given in Equation 2.

Masking Scheme: Erasing or Filling? In Fig. 1, the regular and Gaussian noise masking strategies are illustrated inside the dashed box. Different from Masked Autoencoders (MAE) which is to reconstruct the masked contents for learning good representation, the Masked Siamese Networks will not predict the information in removed areas so erasing will only lose information and is not desired in the learning procedure. In contrast to erasing-based masking, our filling-based strategy will repatch the removed areas using an auxiliary image, as shown in the right part of Fig. 1. After that, we will *switch* the content between the main image and auxiliary image to generate a new image for information completeness of two input images. For a given original image I_i we define its mixture as mix of the pair (I_i, I_{n-i}) and *switch* image of I_i as mixture of the pair (I_{n-i}, I_i). Our soft objective calculation is designed to fit the format of such masked images in Siamese networks, as described in the next section.

3.2 PERSPECTIVE ON DISTANCE IN SIAMESE NETWORKS

Objective Calculation: Inflexible or Soft? It is obvious that different pretext and data processes (e.g., masking, mixture) will change the semantic distance of two branches in the Siamese networks, hence the default symmetric loss will no longer be aligned to reflect the true similarity of the latent representations. It so far has not attracted enough attention for such a problem in this area. In this work, we introduce a soft objective calculation method that can fit the filling-based masking strategy in a better way. To calculate the soft distance, we follow the protocol defined in (Shen et al., 2022). Firstly, we start by generating a binary mask with a fixed grid size that will later be used to mix a batch of images, denoted as $\boldsymbol{I}$, from a single branch. In case when we use an inverse permutation to obtain the mixture, each image in the batch with index i is mixed according to the mask with the image in the same batch but with index $n - i$ as described in Equation. 1:

$$\text{mix}_i := \text{mix}(\boldsymbol{I}_i, \boldsymbol{I}_{n-i}) = mask \odot \boldsymbol{I}_i + (1 - mask) \odot \boldsymbol{I}_{n-i}, \tag{1}$$

$$\text{switch}_i := \text{mix}(\boldsymbol{I}_{n-i}, \boldsymbol{I}_i) = mask \odot \boldsymbol{I}_{n-i} + (1 - mask) \odot \boldsymbol{I}_i, \tag{2}$$

where $\boldsymbol{n}$ is batch size. The mixed image contains parts from both $\boldsymbol{I}_i$ and $\boldsymbol{I}_{n-i}$ whose spatial locations in the mixture are defined by the contents of the binary mask. For the reverse permutation switch mixture will interact with two images, but for random permutation, it will involve an additional image. Furthermore, we calculate a mixture coefficient λ which is equal to the ratio of the masked area to the total area of the image using the Equation 3:

$$\lambda = \frac{\sum\limits_{x,y} \mathbb{1}[mask(x, y) = 1]}{width \cdot height}, \tag{3}$$

where $\mathbb{1}$ is the indicator function that measures masked area of an image. The mixture part of the loss consists of the two terms: loss calculated on the mixed images in the original and reversed order as each image in the batch appears in the mixture exactly two times. Thus, mixture loss can be simply defined by introducing two loss terms calculated between mixed images in original (↑)/reverse (↓) order in a mini-batch:

$$\mathbb{L}_{\text{MixMask}} = \lambda \cdot \mathbb{L}_{\uparrow} + (1 - \lambda) \cdot \mathbb{L}_{\downarrow}. \tag{4}$$

The final loss defined as a summation of the original loss and mixture loss:

$$\mathbb{L} = \mathbb{L}_{\text{Orig}} + \mathbb{L}_{\text{MixMask}}. \tag{5}$$

Permutation Strategies: Incremental or Replacing? We also provide insights on how our method can be combined with Un-Mix. Namely, we explore how different permutation strategies used to produce an image mixture affect the final result of the model trained when Un-Mix and MixMask are applied together. Recall, that by default in Un-Mix a reverse permutation is used to generate the image mixture, i.e. the image with index i is mixed with the image with index $n - i$. We empirically show that to get the best performance for the model which is trained both with Un-Mix and MixMask we need to use a different permutation on the MixMask branch. In this case, we can generate more diverse set of mixed images as different pairs of images are mixed in Un-Mix and MixMask branches respectively. This permutation strategy ultimately leads to a better generalization ability of the model. A detailed illustration of the strategy is shown in Fig. 4.

3.3 Framework overview

Our framework overview is shown in Fig. 2. In this figure, the left is the conventional Masked Siamese ConvNets (MSCN), right is our proposed MixMask with asymmetric distance loss. The motivation behind this design is that directly erasing regions will lose a significant proportion of information in the Siamese ConvNets, which cannot be recovered by post-training. This is quite different from the mechanism of Masked Autoencoders (MAE) that predict masked areas to learn good representations. According to this, we propose a filling-based scheme to overcome the drawback. The soft distance loss is designed to fit the true similarity of the two branches. Despite its conceptual simplicity, we show empirically that with the integrality of mix-masking and objective, we can learn more robust and generalized representations from the masked input.

Loss Function. Here we take a contrastive loss as an example, the MixMask objective will be:

$$\mathbb{L}_{\text{MixMask}} = -\left(\lambda \cdot \log \frac{\exp(q_{\uparrow} \cdot k/\tau)}{\sum_{i=0}^{K} \exp(q_{\uparrow} \cdot k_i/\tau)} + (1 - \lambda) \cdot \log \frac{\exp(q_{\downarrow} \cdot k/\tau)}{\sum_{i=0}^{K} \exp(q_{\downarrow} \cdot k_i/\tau)}\right), \tag{6}$$

where $q_{\uparrow}$ and $q_{\downarrow}$ are normal and reverse orders of mixed queries in a mini-batch respectively, k is the unmixed single key, λ is calculated according to the Equation 3 and τ is the temperature.

3.4 Discussions

Mask Pattern: Blocked or Discrete? Masking strategies determine the difficulty for the network to generate representations in siamese networks, even if the masking ratio is the same, the representation will still be different for different masking patterns. It will directly influence the information of

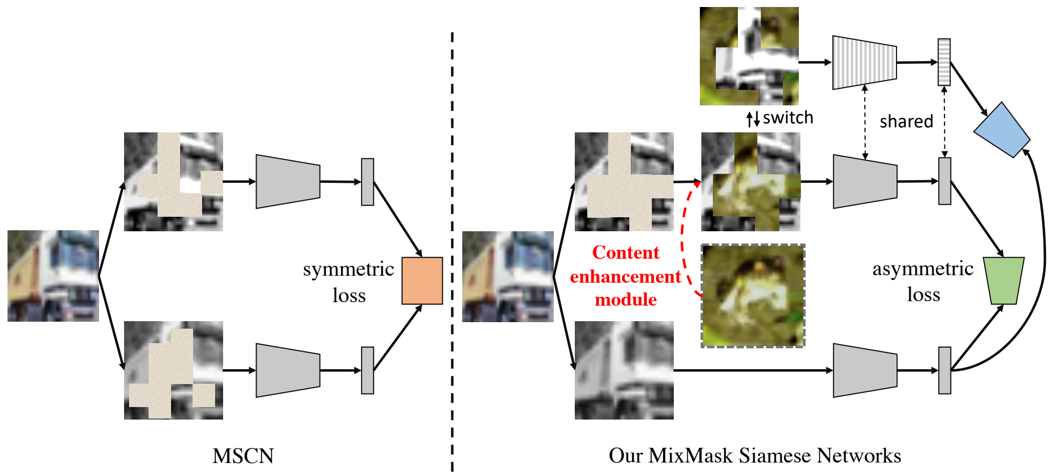

Figure 2: Illustration of the Masked Siamese ConvNets (left) and our proposed framework (right). MixMask branch incorporates asymmetry into the loss function design by generating images with different rates of similarity to the images in the original branch. In MixMask branch image of the truck is presented twice with different levels of similarity to the image in the original branch due to the regions masked with contents of another image.

input, to further affect the latent representations. *Our observation on Masked Siamese Networks is opposite to that in MIM methods which found discrete/random is better.* From our empirical experiments, on CIFAR-100, blocked and discrete masking patterns achieve similar accuracy and discrete is slightly better, however, on Tiny-ImageNet and ImageNet-1K, blocked mask clearly shows superiority over discrete/random. We explain this as that if the input size is small, the mask shape is not so necessary since the semantic information of the object is still preserved. On larger datasets like Tiny-ImageNet and ImageNet-1K, discrete/random masking will entirely destroy the completeness of the object in an image, as shown in Fig. 3 (d, e), while, this is crucial for ConvNet to extract a meaningful representation of the object. Therefore, blocked masking shows superior ability on the MSCN and is a better choice than random masking.

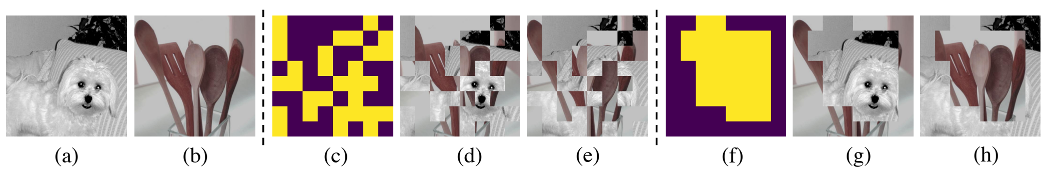

Figure 3: Illustration of the different mask patterns with mask grid size of 8. (a) and (b) are input images. (c) is the discrete/random masking pattern, and (d) and (e) are mixed images using this mask. (f) is the blocked mask pattern, and (g) and (h) are mixed images with a blocked mask.

Symmetric or Asymmetric losses in *Siamese Networks*? There are several paradigms in Siamese networks on input and objective spaces: (i) *Input symmetric* + *objective symmetric* (regular Siamese models). (ii) *Input asymmetric* + *objective asymmetric* (Un-Mix (Shen et al., 2022)) (iii) *Input asymmetric* (slightly) + *objective symmetric* (MSCN (Jing et al., 2022)). Here, *input symmetric* indicates that we use the same probability of data augmentations to generate two *different* views of samples from the same image. *Input asymmetric* means one view/branch contains extra data augmentations like Mixup or CutMix. *Objective symmetric* presents that regular contrastive loss or similarity loss is used, and *objective asymmetric* indicates that the similarity will be calibrated by a soft coefficient. Generally, both input and objective are asymmetric and can force the model to learn subtler and fine-grained information, thus the representation is more robust for downstream tasks.

Relationship to Counterparts in Self-supervised Learning. Un-Mix introduced Mixup and CutMix into the Siamese networks for self-supervised learning. Our MixMask is basically a generalized Un-Mix with arbitrary shape of mask areas. Thus, if considering the self-supervised learning scenario, Un-Mix's region-level mixture can be regarded as a special case of our MixMask scheme. MixMask itself has shown strong representation learning ability in our empirical sturdy, while, it is

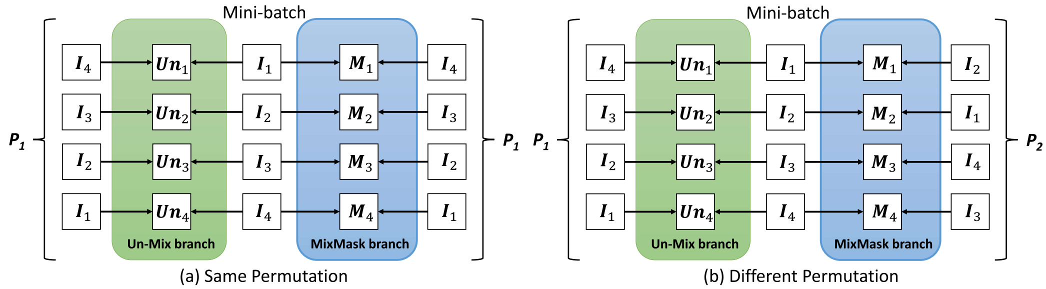

Figure 4: Illustration of different permutation strategies when Un-Mix and MixMask used together. I_i denotes an image in mini-batch, Un_i denotes a mixture image obtained using Un-Mix and M_i denotes a mixture image generated using MixMask. In (a), we use the same inverse permutation P_1 for both Un-Mix branch and MixMask branch, and thus we mix the same pair of images twice. In (b), we use different permutations P_1 and P_2 thus different pairs of images are mixed in different branches yielding more diverse set of training data.

interesting to see that MixMask is also compatible with Un-Mix and can be employed together to further improve the performance.

4 EXPERIMENTS

Base Models. In our experimental section, we use base models including: MoCo V1&V2 (He et al., 2020), Un-Mix (Shen et al., 2022), SimCLR (Chen et al., 2020a), BYOL (Richemond et al., 2020) and BYOL (Richemond et al., 2020). A detailed introduction for each of them is in Appendix.

Datasets and Training settings. We conduct experiments on CIFAR-100 (Krizhevsky et al., 2009), Tiny-ImageNet (Le & Yang, 2015) and ImageNet-1K (Deng et al., 2009) datasets. For CIFAR-100 and Tiny-ImageNet we train each framework for 1000 epochs with ResNet-18 (He et al., 2016) backbone, for ImageNet-1K we pretrain ResNet-50 for 200 epochs and then finetune a linear classifier on top of the frozen features for 100 epochs and report Top-1 accuracy. We use MoCo (and MoCo V2 for ImageNet-1K) as a base framework unless stated otherwise. We report Top1-accuracy on linear evaluation, except the case of MoCo on CIFAR-100 and TinyImageNet for which we provide KNN accuracy. * indicates that we build our method upon Un-Mix. We provide full training configurations and additional KNN evaluation in Appendix.

4.1 ABLATION STUDY

In the below we use a blocked masking strategy and masking ratio of 0.5 if not stated otherwise.

Masking Strategies. We first explore how different masking strategies affect the final result. We consider three different parameters: grid size, grid strategy, and masking ratio. To make our experiments more reliable we consider three datasets with different image sizes: CIFAR-100: 32×32, Tiny-ImageNet: 64×64 and ImageNet-1K: 224×224.

The first parameter grid size specifies the granularity of the $n \times n$ square grid of the mask. We consider the following values of $n = 2, 4, 8, 16, 32, 48$. However, we have different upper bounds for different datasets depending on the spatial size of the images, from our experiments we can conclude that a very large grid size completely disrupts semantic features of the image and leads to poor performance. Our results indicate that optimal grid size increases proportionally to the input size of the image. We obtain the optimal grid size for CIFAR-100 is 2, for Tiny-ImageNet is 4, and for ImageNet-1K is 8. The masks with very low and very large grid sizes show bad performance. We believe this happens because a small grid size does not provide enough variance in the mask structure whilst a large grid size destroys the important semantic features of the image.

We consider two different strategies for the random mask generation, namely discrete/random mask and blocked mask. We generate a blocked mask according to the algorithm described in (Bao et al., 2021). A discrete mask does not have any underlying structure, whilst a blocked mask is generated in a way to preserve global spatial continuity and, thus, more suitable for capturing global semantic

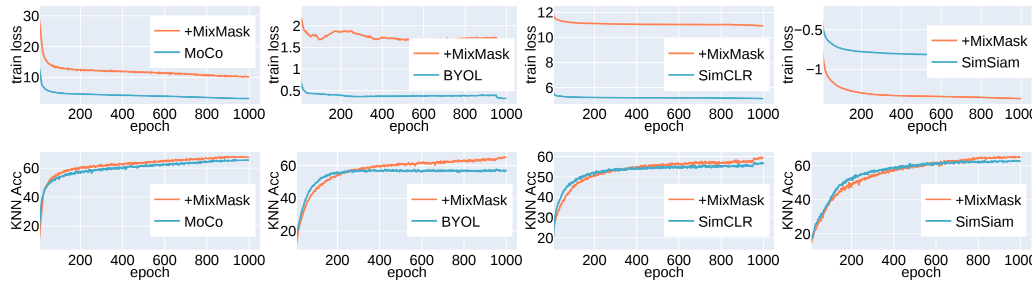

Figure 5: Training losses (top row) and KNN evaluation accuracy (bottom row) on CIFAR-100 were plotted against the number of epochs for different self-supervised frameworks. MixMask (red) outperforms vanilla baseline (blue) on all frameworks by a significant margin.

features. For CIFAR-100 we observe a negligible difference between blocked and discrete masks. On the other hand, for Tiny-ImageNet and ImageNet-1K which have larger spatial sizes of the image, blocked mask performs better than discrete. This result is different from MAE, where a discrete mask achieves better performance and highlights the importance of maintaining global features when generating an image mixture as opposed to the case when erasing parts of the image by performing a vanilla masking operation.

Table 1: Ablation study on masking strategies using MoCo V1 and V2 (ImageNet-1K) on various datasets with only MixMask branch.

	CIFAR-100	**Tiny-ImageNet**	**ImageNet-1K**
Input size	32×32	64×64	224×224
		Grid Size	
2×2	**68.11**	46.4	68.44
4×4	67.55	**47.44**	68.95
8×8	67.51	46.48	**69.18**
16×16	–	46.10	68.54
32×32	–	–	68.65
48×48	–	–	68.79
		Masking Strategy	
blocked	67.78	**47.44**	**69.18**
discrete	**68.11**	46.40	68.44
		Masking Ratio	
0.25‖0.75	66.78	45.60	68.53
0.5	**68.11**	**47.44**	**69.18**
uniform(0, 1)	67.56	45.46	68.71
uniform(0.25, 0.75)	67.78	47.04	68.85

For the masking ratio we consider two cases with constant values of 0.5 and $0.25/0.75$ (as there is no difference between these two due to the loss function design) and two cases when the value is sampled from the uniform distribution with different bounds. We obtain the best results for masking ratio 0.5 under different settings, conjecturing that this value causes the blocked mask to generate consistent global views for both of the images being mixed. Sampling masking ratio from a uniform distribution with bounds 0.25 and 0.75 yields better results than using 0 and 1 as bounds. We believe this shows that extreme values of masking ratio close to either 0 or 1 generate a mixture where one image heavily dominates over the other when in the optimal mixture areas of each image should be roughly proportional to each other.

Training Budgets. We test our method with different training budgets, including 200 epochs; 400 epochs; 600 epochs; 800 epochs; 1000 epochs on CIFAR-100 using MoCo, SimSiam, BYOL, and SimCLR. The results are shown in Fig. 5. We can observe our method achieves consistent improvement over various frameworks.

Compatibility with Un-Mix. We examine the question of whether our method can be applied together with Un-Mix at the same time. We consider two cases: in the first case, both Un-Mix and MixMask branches use the same permutation to generate the mixed images. In the second case mixtures for Un-Mix and MixMask branches are generated using different permutations. Our results highlight the importance of using different permutations when mixing images on Un-Mix and

MixMask branches respectively. Clearly, when using the same permutations, even though Un-Mix and MixMask perform different operations on images, there is still some redundancy and duplicated information in the produced mixtures, as each pair of images is being mixed twice. On contrary, when different permutations are used, different sets of pairs are generated in different branches. Thus, we produce a more diverse and richer set of mixed training samples yielding better performance. Furthermore, we consider the effect of using two branches together on the parameter of the probability of global mixture in Un-Mix. In (Shen et al., 2022) it has been shown that optimal parameter for the global mixture on ImageNet-1K is $P = 0$. We challenge this assumption in the setting with two branches as we believe adding MixMask branch can change the optimal value of P in Un-Mix. Indeed we yield the best performance for $P = 0.5$, theorizing that CutMix is essentially the special case of MixMask applied with a blocked mask, which enables the opportunity for the model to benefit from MixUp part of Un-Mix.

Table 2: Results using different permutation strategies when Un-Mix and MixMask are applied together. We compare two different strategies: using the same or different permutations on Un-Mix and MixMask branches. Applying different permutations produces the best performance.

	CIFAR-100			**Tiny-ImageNet**		
Permutations	MoCo	BYOL	SimCLR	MoCo	BYOL	SimCLR
Same	69.24	71.84	**70.05**	47.42	50.78	49.70
Different	**69.95**	**72.14**	70.04	**47.90**	**53.72**	**50.72**

Table 3: Results on ImageNet-1K for the optimal value of probability for global mixture (P) for Un-Mix when it is used together with MixMask. We report Top-1 accuracy for the linear evaluation protocol for 100 epochs.

$P = 0$	$P = 0.5$	$P = 1.0$
69.09	**69.51**	68.72

Table 4: Comparison of erase-based masking with MixMask on ImageNet-1K. MixMask shows better performance.

Method	Top-1 acc
Baseline	67.5
Erase-Based	68.23
MixMask	**69.18**

Results for different base frameworks. We consider generalizability of our method by applying it on top of four different self-supervised learning frameworks. In this set of experiments we use masking ratio of 0.5, blocked mask and set grid size for each dataset according the optimal values highlighted in Table 1. When applying our method on top of Un-Mix we yield superior performance in all cases. We can also observe really good performance of plain MixMask is some cases as well.

Table 5: Ablation on different base framework approaches.

	CIFAR-100				**Tiny-INet**		**INet-1K**
	MoCo	**SimSiam**	**BYOL**	**SimCLR**	**MoCo**	**BYOL**	**MoCoV2**
Vanilla	65.67	66.71	66.8	67.11	42.76	51.4	67.50
Un-Mix (Shen et al., 2022)	68.62	70.02	70.02	69.77	45.32	52.92	68.50
MixMask (Our)	68.36	69.83	71.88	69.22	47.44	50.78	69.18
MixMask* (Our)	**69.95**	**70.56**	**72.14**	**70.04**	**47.90**	**53.72**	**69.51**

4.2 Comparison with erase-based masking

We further show that our approach is superior to erase-based masking which is used in (Jing et al., 2022). For that, we compare our method against baseline and erase-based masking. It is not straightforward how to compare erase-based masking and our proposed method as in our case the mixture loss contains two terms because each image appears in the mixture twice, whilst the resulting image with vanilla masking contains only one source image. To make the comparison fair for erase-based masking we still keep two terms in the loss function with coefficients λ and $1 - \lambda$. In this way erase-based masking will receive the same amount of discriminative signal as MixMask.

Erase-based masking provides an improvement over the baseline showing the usefulness of an additional branch. MixMask further solidifies this improvement by proposing a way to fill the erased regions in a meaningful way.

4.3 Comparison to state-of-the-art approaches

We provide a comparison with state-of-the-art approaches in Table 5. For CIFAR-100 and Tiny-Imagenet we report KNN accuracy obtained from our experiments. For ImageNet-1K we report

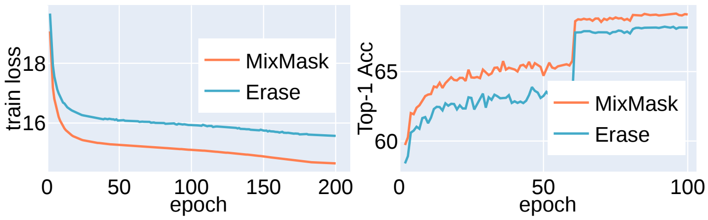

Figure 6: Comparison of training loss (left) and Top-1 accuracy (right) curves for erase-based masking and MixMask on ImageNet-1K. The latter outperforms erase-based masking by almost 1%.

Top-1 accuracy either from our experiments or from other papers. When reporting values for Un-Mix and MixMask we use MoCo as an underlying framework. MixMask shows very competitive results across all the datasets. As indicated in Table 5 it is able to generalize well across different self-supervised learning frameworks.

4.4 Results on semi-supervised evaluation

We perform a semi-supervised evaluation on 1% and 10% labels on top of the frozen weights. We follow the same configuration setting as in (Caron et al., 2020). Please note that we are using checkpoints obtained for 200 epoch training.

Table 6: Results on semi-supervised finetuning on ImageNet-1K with 1% and 10% labels.

	1%		**10%**	
	Top-1 Acc	Top-5 Acc	Top-1 Acc	Top-5 Acc
Un-Mix	34.23	60.34	62.73	85.36
MixMask	34.36	60.72	63.02	85.69
MixMask*	**36.59**	**62.94**	**63.93**	**86.25**

4.5 Results on object detection and segmentation

We test our method on the downstream task of object detection and segmentation. For that we fine-tune a Faster-RCNN (Ren et al., 2015) and Mask-RCNN (He et al., 2017) models implemented in Detectron2 (Wu et al., 2019) library on Pascal VOC 2007 (Everingham et al., 2010) and MS COCO 2017 (Lin et al., 2014) datasets. For Pascal VOC 2007 we follow the standard evaluation protocol defined in (He et al., 2020) with 24k training iterations. As in semi-supervised evaluation we use checkpoints from epoch 200.

For COCO we use `mask_rcnn_R_50_C4_1x` configuration from (Wu et al., 2019). In this experiment, we replaced `Res5ROIHeadsExtra` with `Res5ROIHeadsExtraNorm` and added `SyncBN` to avoid a loss divergence issue which was reported by others as well. This may be the reason why our result may slightly differ from the result reported in the Detectron2 library.

Table 7: Object detection and segmentation results on COCO. MixMask and MixMask* in general perform better than Un-Mix.

	MS COCO 2017												**Pascal VOC 2007**		
	Object detection						**Segmentation**						**Object detection**		
	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75
Un-Mix	36.25	55.57	38.89	**19.40**	40.72	50.68	31.88	**52.47**	33.81	**13.62**	35.23	49.20	57.69	82.98	64.53
MixMask	36.42	55.56	**39.27**	19.33	40.92	50.86	**31.96**	52.18	**34.06**	13.38	**35.26**	49.34	57.56	82.83	64.00
MixMask*	**36.50**	**55.87**	39.18	19.29	**40.97**	**50.88**	31.95	52.40	33.70	13.07	34.97	**49.67**	**57.73**	**83.22**	**64.61**

5 Conclusion

In this work, we have revisited the conventional loss function in the *Masked Siamese ConvNets* and proposed a dynamic and asymmetric loss design for the mix-masking distance calculation. To avoid the information loss in the vanilla masking scheme, we further proposed a filling-based masking strategy MixMask that is stronger for learning good representations in self-supervised learning. Extensive experiments are conducted on CIFAR-100, Tiny-ImageNet, and ImageNet-1K across different frameworks and downstream tasks. Our method outperforms the state-of-the-art baselines by a significant margin.

REFERENCES

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *arXiv preprint arXiv:2204.07141*, 2022.

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

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

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022.

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

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018.

Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Omnimae: Single model masked pretraining on images and videos. *arXiv preprint arXiv:2206.08356*, 2022.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv preprint arXiv:2207.13532*, 2022.

Li Jing, Jiachen Zhu, and Yann LeCun. Masked siamese convnets. *arXiv preprint arXiv:2206.07700*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

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

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video, 2018.

Zhiqiang Shen, Zechun Liu, Zhuang Liu, Marios Savvides, Trevor Darrell, and Eric Xing. Un-mix: Rethinking image mixtures for unsupervised visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.

APPENDIX

Here we provide details that were not included in the main part of the manuscript, namely:

• Section A ”Base Models & Datasets”: includes the description of the self-supervised learning frameworks and datasets which are used in the experiments.

• Section B ”Training configurations”: we provide training configuration details for the experiments.

• Section C ”Additional KNN-evaluation results”: contains additional KNN-evaluation results for base framework and permutation experiments.

A BASE MODELS & DATASETS

In this section we provide the description of self-supervised learning frameworks and datasets that we used in experiments. To test our method we tried to select a diverse set of frameworks that incorporate different mechanisms to avoid model collapse and follow different design paradigms, i.e. vanilla contrastive learning with negative pairs vs knowledge distillation.

A.1 BASE MODELS

MoCo V1&V2 (He et al., 2020) Momentum Contrast is a self-supervised contrastive learning framework that employs a memory bank to store negative samples. MoCo V2 is an extension of the original MoCo which introduces a projection head and stronger data augmentations.

Un-Mix (Shen et al., 2022) Un-Mix is an image mixture technique with state-of-the-art performance for unsupervised learning, which uses CutMix and MixUp at its core. It smooths decision boundary and reduces overconfidence in model predictions by introducing an additional mixture term to the original loss value which is proportional to the degree of the mixture.

SimCLR (Chen et al., 2020a) Simple Contrastive siamese framework with two branches uses contrastive loss to attract positive and repel negative instances using various data augmentations.

BYOL (Richemond et al., 2020) BYOL is a self-supervised learning technique that does not use negative pairs. It is composed of two networks, an online and a target. The task of an online network is to predict the representations produced by the target network. EMA from the online network is used to update the weights of the target.

A.2 DATASETS

CIFAR-100 (Krizhevsky et al., 2009) CIFAR-100 consists of 32×32 images with 100 classes. There are 50000 train images and 10000 test images, 500 and 100 per class respectively.

Tiny-ImageNet (Le & Yang, 2015) This is a dataset containing 64×64 colored natural images with 200 classes. The test set is composed of 10000 test images, whilst the train contains 500 images per category totaling 100000 images.

ImageNet-1K (Deng et al., 2009) It has images with a spatial size of 224×224. 1,281,167 images span the training which includes 1K different classes, whilst the validation set includes 50K images.

SimSiam (Chen et al., 2020a) The authors examined the effect of the different techniques which are commonly used to design siamese frameworks for representation learning. As a result, they proposed a simple framework with two branches that relies on the stop gradient operation on one branch and an extra prediction module on the other.

B TRAINING CONFIGURATIONS

In this section we provide hyperparameter configurations for training on CIFAR-100 and Tiny-ImageNet. In all these experiments we use ResNet-18 as a backbone and train for 1000 epochs with batch-size 512 on a single GPU. Probability of global mixture in Un-Mix is set to 0.5.

Table 8: Training settings on CIFAR-100 and Tiny-ImageNet. Comma separated values correspond to CIFAR-100 and Tiny-ImageNet respectively.

MoCo		SimCLR & BYOL		SimSiam	
hparam	value	hparam	value	hparam	value
backbone	resnet18	backbone	resnet18	backbone	resnet18
optimizer	SGD	optimizer	Adam	optimizer	SGD
lr	0.06	lr	0.003, 0.002	lr	0.03
batch size	512	batch size	512	batch size	512
opt momentum	0.9	proj layers	2	opt momentum	0.9
epochs	1000	epochs	1000	epochs	1000
weight decay	5e-4	weight decay	5e-4	weight decay	5e-4
embed-dim	128	embed-dim	64, 128	embed-dim	128
moco-m	0.99	Adam l2	1e-6	warmup epochs	10
moco-k	4096	proj dim	1024	proj layers	2
unmix_prob	0.5	unmix_prob	0.5	unmix_prob	0.5
moco-t	0.1	byol tau	0.99		

C ADDITIONAL KNN-EVALUATION RESULTS

In this section we provide supplementary KNN-evaluation results for testing our method on different frameworks and permutation strategies. The obtained results are consistent with the result from linear evaluation.

Table 9: Ablation of framework designs using different base approaches on CIFAR-100 and Tiny-ImageNet. * indicates that we build our method upon Un-Mix. KNN-evaluation is reported.

	CIFAR-100			**Tiny-INet**
	SimSiam	**BYOL**	**SimCLR**	**BYOL**
Vanilla	63.16	57.77	57.16	35.67
Un-Mix (Shen et al., 2022)	66.45	62.17	60.53	37.64
MixMask (Our)	65.26	**65.30**	59.94	36.04
MixMask* (Our)	**66.69**	65.01	**61.48**	**40.02**

Table 10: Results for KNN evaluation on CIFAR-100 and Tiny-ImageNet using different permutation strategies when Un-Mix and MixMask are applied together. Applying different permutations produces the best performance in almost all cases.

	CIFAR-100		**Tiny-ImageNet**	
Permutations	BYOL	SimCLR	BYOL	SimCLR
Same	64.42	**61.82**	36.48	34.9
Different	**65.01**	61.48	**40.02**	**34.96**