Masking Improves Contrastive Self-Supervised Learning for ConvNets, and Saliency Tells You Where
– *Supplementary Materials* –

Zhi-Yi Chin[1*] Chieh-Ming Jiang[1*] Ching-Chun Huang[1] Pin-Yu Chen[2] Wei-Chen Chiu[1]
[1] National Yang Ming Chiao Tung University
[2]IBM Research
{joycenerd.cs09, nax1016.cs10, chingchun}@nycu.edu.tw, pin-yu.chen@ibm.com, walon@cs.nctu.edu.tw

1. Appendix

In this appendix, we firstly provide the summarization of our contributions with respect to our two main baselines MSCN [12] and ADIOS [20] (cf. Section 1.1 and Section 1.2 respectively) as well as our contribution in terms of saliency masking (cf. Section 1.3). Furthermore, we show the efficacy of our three different masking strategies (i.e., high-pass filtering, strong blurring, and mean filling) with more experiments as well as discuss the computational cost: In Section 1.5, we provide detailed experimental setups and conduct various downstream tasks (i.e., classification, object detection, and semantic segmentation) in different contrastive SSL frameworks (i.e., MoCov2 [4] and SimCLR [3]); While in Section 1.6, we compare the computational cost of three different masking strategies with MSCN [12] and ADIOS [20].

1.1. Emphasis upon our contribution compared to baseline MSCN [12]

Here we would like to emphasize again that our contributions stand out from the ones of MSCN [12] as it includes: 1) *Saliency masking* *with various masking strategies* (their benefits are shown in Table 4 and 1 of the main manuscript), in which MSCN does not adopt saliency-guided masking but applies random masking, and its masking strategy based on high-pass filtering constrains the setting of downstream tasks (since the input for the downstream tasks needs to be firstly high-pass-filtered as well, i.e. having the prior knowledge upon how the pre-training of feature extractor is done, c.f. lines 360-372 in our main manuscript). Our proposed strong blurring and mean-filling masking strategies are novel and practical as they do not have such constraints, thus being more flexible; 2) Based on the explicit analysis of *variance manipulation*, our proposed method applies masking solely on the query branch of the siamese framework and is shown to consistently improve the performance for all masking strategies (c.f. Table 6 of the main manuscript); 3) Generating the *hard negative samples* easily by masking only the foreground patches with the help of saliency (cf. Table 8 of the main manuscript for the improvement based from such design).

1.2. Emphasis upon our contribution compared to baseline ADIOS [20]

We would like to emphasize that our contributions stand out from the ones of ADIOS [20] as it includes: 1) *Efficiency in obtaining (partially) semantic masks*. While both our proposed method and ADIOS employ a localization network to address the “where to mask” issue, our approach achieves a more favorable trade-off between obtaining (partially) semantic masks and computational effort. Notably, the localization network we utilize remains frozen during feature extractor training, whereas ADIOS requires joint training of the localization network (UNet-based segmentation model) alongside the feature extractor; 2) *Variance manipulation in single branch*. ADIOS masks a *single view* while both views (i.e. masked and unmasked) will go through both query and key branches (as indicated in their source code). In comparison, our design shows that incorporating variance manipulation through masking only the query branch has a positive impact on the Siamese network. Our method differs from ADIOS in terms of both operation and motivation (i.e. variance manipulation). Further details of our investigation and discussion can be found in lines 744-807, and while corresponding ablation studies can be found in Tables 6 and 7.

1.3. Our contribution in saliency masking

As described in lines 101-113 in our main manuscript, and we would like to clarify again here: most existing studies of adopting masking operations (together with self-reconstruction objective) to realize self-supervised learning are based on the *transformer backbone* thanks to the tokenized input (where the masking is simply to block out some tokens), and the prior works (e.g. SemMAE [13], MST [14], BEiT [2], iBOT [24], MAE [10], and SimMIM [22]) are designed for transformers as well. In a recent study, MSN [1] introduces a clustering-based self-supervised learning method. Their approach involves assigning the query view (where random masking is applied) and the key view from the two branches of the Siamese network to the same cluster. To prevent the adverse impact of masked patches on the model, they opt for ViT as the backbone network and remove the masked patches in the query view during training. Notably, this technique is applicable specifically once while ViT (i.e. vision transformer) serves as the backbone. In contrast, we aim to apply masking for *convolutional neural networks*, which is actually nontrivial due to the unwanted edges caused by masking (and that is exactly why MSCN [12] needs to introduce the high-pass filtering at first). Moreover, even there exists some transformer-based prior works adopting the saliency operations as well, the ways of their applying saliency masking are also different from ours: For instance, SemMAE [13] requires a two-stage training process to determine where to apply the mask, while our approach achieves the same goal with a single feature extractor and end-to-end training; MST [14] also aims to avoid masking important objects, while our method of explicitly distributing masked patches across foreground and background empirically leads to better performance.

Additionally, in the absence of our proposed saliency-guided masking, creating effective hard negative samples through masking can be challenging. Strategic masking of salient patches allows for the generation of impactful hard negative samples without the need to mask a large number of patches, considering the potential detrimental effects of masking on ConvNets.

1.4. Implementation Detail

Here, we introduce how the parameters are set in our saliency masking approach. We set the penalty ratio, denoted as ρ, for hard negative samples to 2, in which such setting is identified to effective as well in [7]. For our strong blurring masking strategy, we establish a Gaussian blur kernel of size of 31×31 with setting its standard deviation to be 10. The parameters for the high-pass filter, specifically the radius and standard deviation, are adopted from MSCN [12]. Regarding the positive masking ratio, its ranges among $\mathcal{U}(0.05, 0.25)$ of the total patches, while for the hard negative masking ratio, the range among $\mathcal{U}(0.4, 0.7)$ of the salient patches is adopted. It's worth noting that we do not find an optimal fixed masking ratio, and instead, we determined these ratios by extending the search range, ensuring that performance did not significantly deteriorate.

1.5. More Experimental Results

In this section, we provide the results for all the downstream tasks with three different masking strategies (i.e., high-pass filtering, strong blurring, and mean filling) and baselines (i.e., MSCN [12] and ADIOS [20]) based on two contrastive SSL frameworks (i.e., MoCov2 [4] and SimCLR [3]). In the pretraining stage, we train the feature encoder (under MoCov2 and SimCLR frameworks) with using ResNet-50 as the backbone on the ImageNet-100 [19] dataset for 200 epochs. We conduct experiments on three datasets (i.e., ImageNet-100, Caltech-101 [6], and Flower-102 [16]) for downstream classification tasks and supervisedly train a linear classifier while the feature encoder is kept fixed/frozen for 100 epochs. We conduct experiments on VOC07+12 [5] and COCO [15] datasets for downstream detection tasks, where the COCO dataset is also used for the downstream instance segmentation task. For the VOC07+12 dataset, we adopt the Faster R-CNN [17] model with C4 backbone which is finetuned for 24k iterations; while for the COCO dataset, we adopt the Mask R-CNN [11] model with C4 backbone which is finetuned for 180k iterations (using 1× learning rate schedule).

MoCov2 Results. First of all, please note that most of the results based on MoCov2 framework have been provided in our main paper, here we particularly include them again for the purpose of having better and more complete overview. For MoCov2, we set the batch size to 128 and the base learning rate to 0.015 and use SGD [18] as the optimizer during pretraining. When training the linear classifier, we set the base learning rate to 30.0 and adopt a learning rate schedule that decreases the learning rate by 0.1 at epochs 60 and 80. All MoCov2's downstream classification results are reported in the upper half of Table 1, while all the downstream detection and instance segmentation results are reported in the upper half of Table 2. Our method outperforms the fundamental contrastive SSL framework (i.e., MoCov2, which has no masking involved) and two baselines (i.e., MSCN and ADIOS) in all the downstream tasks.

SimCLR Results. For SimCLR, we set the batch size to 256, the base learning rate to 0.3, and use LARS [23] as the optimizer during pretraining. When training the linear classifier, we set the batch size to 256, the base learning rate to 1.0, and adopt a cosine learning rate schedule. All the SimCLR's downstream classification results are reported in the lower half of Table 1, while all the downstream detection and instance segmentation results are reported in the lower half of Table 2. Our method outperforms the fundamental contrastive SSL framework (i.e., SimCLR, which has no masking involved) and two baselines (i.e., MSCN and ADIOS) in all the classification tasks; but ADIOS slightly outperforms our method in the detection and instance segmentation tasks, where we attribute this to two reasons. Firstly, according to ablation studies conducted in [21], ma-

Method	ImageNet-100	Caltech-101	Flowers-102
Supervised	82.72	21.99	20.29
MoCov2	68.22	81.87	88.39
+ MSCN [12]	70.28	**84.13**	90.10
+ ADIOS [20]	62.76	79.83	88.39
+ OURS (High-pass filtering)	**73.80**	**84.91**	**90.95**
+ OURS (Strong blurring)	**72.50**	83.95	90.59
+ OURS (Mean filling)	70.84	82.68	**90.83**
SimCLR	69.77	78.20	85.21
+ MSCN [12]	77.18	**86.99**	**91.08**
+ ADIOS [20]	71.12	81.96	87.53
+ OURS (High-pass filtering)	**77.90**	**87.04**	90.71
+ OURS (Strong blurring)	**77.78**	83.41	**91.93**
+ OURS (Mean filling)	77.36	83.55	90.83

Table 1. Linear evaluation results on ImageNet-100, Caltech-101 and Flowers-102. The best and second-best results on each dataset with different constrastive SSL frameworks (i.e., MoCov2, SimCLR) are marked in orange and blue respectively.

Method	VOC07+12 detection			COCO detection			COCO instance segmentation		
	AP_{all}	AP_{50}	AP_{75}	AP^{bb}_{all}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}_{all}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised	44.30	73.47	46.50	37.84	57.09	40.67	33.14	53.95	35.31
MoCov2	50.27	76.68	54.76	38.52	57.62	41.67	33.75	54.70	35.86
+ MSCN	50.27	76.99	54.70	38.80	58.09	**42.20**	33.89	54.78	**36.36**
+ ADIOS	45.85	73.44	48.45	38.12	57.38	41.29	33.38	54.25	35.63
+ OURS (High-pass filtering)	**50.89**	**77.66**	**55.44**	**39.16**	**58.62**	**42.45**	**34.22**	**55.28**	36.30
+ OURS (Strong blurring)	**50.76**	**77.29**	54.75	38.90	**58.13**	42.11	**33.93**	54.77	**36.53**
+ OURS (Mean filling)	50.59	76.97	**55.30**	**38.93**	58.08	42.17	33.92	**54.86**	36.27
SimCLR	40.34	69.86	40.96	36.30	55.55	38.80	31.99	52.28	33.80
+ MSCN	43.50	73.18	**45.04**	37.88	57.44	40.68	33.36	54.15	35.57
+ ADIOS	**43.83**	**73.42**	**45.01**	**38.76**	**58.35**	**41.96**	**33.94**	**54.96**	**36.23**
+ OURS (High-pass filtering)	**43.76**	**73.43**	44.90	**38.45**	**57.79**	**41.58**	**33.90**	**54.70**	**35.93**
+ OURS (Strong blurring)	43.20	73.15	44.27	37.44	56.80	39.96	32.92	53.73	35.00
+ OURS (Mean filling)	43.20	72.54	44.79	37.27	56.46	40.10	32.68	53.35	34.54

Table 2. Transfer learning results on VOC07+12 and COCO detection tasks, and COCO instance segmentation task. Performances in terms of AP_{all}, AP_{50} and AP_{75} metrics are reported, and the best and second-best results on each task of different contrastive SSL frameworks (i.e., MoCov2, SimCLR) are marked in orange and blue respectively.

nipulating variance across branches in symmetric encoders (i.e. SimCLR) does not improve as much as that in asymmetric encoders (i.e., MoCov2), limiting improvement in our three masking strategies. Secondly, more detailed semantically meaningful masks of ADIOS are learnt in its pretraining stage, which yield better performance for the downstream detection and instance segmentation tasks (as both detection and instance segmentation can be seen as more detailed recognition tasks than classification). However, noting that an occlusion module needs to be trained jointly with the main SSL objective to learn such masks for ADIOS (thus being believed to require more computational efforts). In contrast, our saliency masking utilizes a pretrained localization network before masking (where the resultant masks are less detailed than the ones in ADIOS but no additional joint learning is required) and still contributes to the comparable results with ADIOS.

Supervised Baseline Results. To establish a solid foundation, we create a supervised baseline. In this baseline, we train an image classification model using ResNet-50 as the feature extractor. Our training setup involves using a batch size of 256, a base learning rate of 0.1, and a learning rate decay of 10 every 30 epochs, and employing SGD as our optimizer when training on the ImageNet-100 dataset. For downstream classification tasks involving Caltech-101 and Flowers-102, we follow our SSL approaches. In these cases, we kept the ResNet-50 feature which is trained on ImageNet-100 fixed, and train a linear classifier with hyperparameters similar to our SSL approaches. All classification tasks undergo 100 epochs of training. Regarding downstream detection and instance segmentation tasks, we utilize settings similar to those used in our SSL methods. The top row of Table 1 presents the results for the supervised baseline classification, while the top row of Table 2 showcases the results for downstream detection and instance segmentation. Despite achieving the highest accuracy in ImageNet-100 classification, the supervised baseline exhibits the poorest transferability. Both transfer classification tasks achieve only 20% accuracy, a result we attribute to the distribution differences between the ImageNet-100 and Caltech-101/Flowers-102 datasets.

Monocular Depth Estimation Downstream Task Results In addition to the commonly addressed downstream tasks of classification, detection, and instance segmentation in most SSL previous works, we have extended the evaluation of our approach to include monocular depth estimation. To achieve this, we adopt Monodepth2 [9] as our reference and substitute its feature encoder with our pretrained ResNet-50, which remains frozen during training. We maintain identical hyperparameters to those used in Monodepth2 and conduct our evaluation on the KITTI 2015 dataset [8]. The results are presented in Table 3. Notably, whereas Monodepth2 trains all model components, we exclusively train the depth decoder and the pose network while keeping the feature encoder fixed. Our mean filling masking strategy produces results on par with the original Monodepth2, and all our settings outperform baseline methods (MoCov2, MoCov2+MSCN, MoCov2+ADIOS). Furthermore, our approach's learned features demonstrates the capacity to generalize to tasks beyond the scope of traditional classification, detection, and instance segmentation.

1.6. Computational Cost

We compare the computational cost of ADIOS [20], MSCN [12], and our three masking strategies (i.e., high-pass filtering, strong blurring, and mean filling) using MoCov2 as the SSL framework. Training time (in minutes) per epoch in ImageNet-100 for each method is measured. Serving as the base SSL framework of all methods, MoCov2 takes 5.5 minutes to train one epoch. In order to alleviate the parasitic edges caused by masking operation in ConvNets, MSCN [12] adopts a high-pass filter and applies random masking (including channel-wise and focal masking) on input images, which in results takes 7 minutes per epoch. Instead of randomly masking, ADIOS [20] proposes an UNet-based occlusion module to adversarially learn along with the feature encoder to determine the regions to be masked, which is called masking slot. The memory and computational cost will increase linearly as the number of masking slots increases. 10 minutes are needed to train one epoch with 6 masking slots in ADIOS. In order to determine where and how to mask in an easier way, our three masking strategies consist of saliency computation and different image processing. In saliency computation, two forward passes through the localization network are needed to produce saliency maps for positive and (hard) negative samples. Compared to MSCN, although it takes 2 minutes longer per epoch due to the saliency constraint in our high-pass filtering strategy, we achieve better performance on various downstream tasks. While in mean filling and strong blurring strategies, mean value and strong blurred patches are filled in the masked regions to make those edges caused by masking less visible, in total each epoch takes 7.5 and 10.5 minutes respectively for their training. The strong blurring strategy spends more time than other strategies, in which the bottleneck is attributed to the GPU I/O. Since our saliency masking procedure is done on GPU, for our strong blurring strategy, we need to move both the standard augmented images and strong blurred images onto the GPU. The data transfer time will be twice that of our other two strategies (i.e., high-pass filtering strategy only moves images onto GPU after high-pass filtering, while mean filling strategy only moves the images onto GPU after standard data augmentation). We will keep improving the overall GPU I/O procedure for our proposed strategies. Furthermore, we test the accuracy of our high-pass filtering method against MSCN on ImageNet-100, with matching pre-training times. MSCN achieves its highest accuracy 70.28% in 197 epochs, while around the same time our method based on high-pass filtering masking strategy reaches 131 epochs but results to have 71.66% accuracy, which is already 1.4% higher than MSCN. To sum up, our high-pass filtering strategy strikes a better balance between efficiency and efficacy than MSCN and ADIOS.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 2

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 2

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incre-

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [9]	**0.132**	**1.053**	**5.159**	**0.211**	**0.846**	**0.949**	0.976
MoCov2	0.140	1.112	5.430	0.218	0.826	0.943	**0.977**
+ MSCN	0.140	1.104	5.416	0.219	0.826	0.944	**0.977**
+ ADIOS	0.139	1.080	5.355	0.217	0.830	0.946	0.976
+ OURS (High-pass filtering)	0.138	1.074	5.331	0.216	0.828	0.945	**0.977**
+ OURS (Strong blurring)	**0.135**	1.098	5.357	0.214	0.838	0.947	**0.977**
+ OURS (Mean filling)	**0.132**	**1.043**	**5.263**	**0.210**	**0.842**	**0.950**	**0.979**

Table 3. Transfer depth estimation results wih modified Monodepth2 through monocular training on KITTI 2015 [8] utilizing the Eigen split. Metrics presented in red cells denote that lower values are preferred, whereas those in blue cells suggest that higher values are desirable. The best results are marked in orange, while the second-best results are marked in blue.

mental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2004. 2

[7] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *NeurIPS*, 2021. 2

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 4, 5

[9] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, October 2019. 4, 5

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[12] Li Jing, Jiachen Zhu, and Yann LeCun. Masked siamese convnets. *ArXiv:2206.07700*, 2022. 1, 2, 3, 4

[13] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *Advances in Neural Information Processing Systems*, 2022. 2

[14] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 2021. 2

[15] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[16] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008. 2

[17] 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. 2

[18] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 2

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[20] Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *ICML*, 2022. 1, 2, 3, 4

[21] Xiao Wang, Haoqi Fan, Yuandong Tian, Daisuke Kihara, and Xinlei Chen. On the importance of asymmetry for siamese representation learning. In *CVPR*, 2022. 2

[22] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 2

[23] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 2

[24] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. In *ICLR*, 2022. 2