SAMCLR: Contrastive pre-training on complex scenes using SAM for view sampling

Benjamin Missaoui *
benjamin.missaoui@gmail.com
Georgia Institute of Technology

Chongbin Yuan
chongbinyuan@gmail.com
National University of Singapore

Abstract

In Computer Vision, self-supervised contrastive learning enforces similar representations between different views of the same image. The pre-training is most often performed on image classification datasets, like ImageNet, where images mainly contain a single class of objects. However, when dealing with complex scenes with multiple items, it becomes very unlikely for several views of the same image to represent the same object category. In this setting, we propose SAMCLR, an add-on to SimCLR which uses SAM to segment the image into semantic regions, then sample the two views from the same region. Preliminary results show empirically that when pre-training on Cityscapes and ADE20K, then evaluating on classification on CIFAR-10, STL10 and ImageNette, SAMCLR performs at least on par with, and most often significantly outperforms not only SimCLR, but also DINO and MoCo.

1 Introduction

Self-Supervised Learning (SSL) is now a well-established and reliable way to train neural networks to produce robust image representations, without ever providing labels. By first pre-training the model with a pretext task such as contrastive learning [4, 12, 3], clustering [1, 2], pseudo-labeling [10, 5] or more recently masked image modeling [13, 22], SSL approaches have been able to produce features that compete or even outperform their supervised counterparts in many downstream tasks like image classification, object detection or image segmentation. This paper focuses on contrastive learning.

Motivation Pre-training with a contrastive learning objective is most often performed on an image classification dataset, and ImageNet [8] is the de-facto choice of the research community. While ImageNet is indeed a high-quality, large-scale and highly curated dataset, its popularity in contrastive learning also comes from the fact that it is an object-centric (in opposition with scene-centric) dataset. In the contrastive learning scheme, models are trained to output representations that are invariant to various image augmentations. If an image only contains one main object, it is highly likely that two random views from this image will also contain at least part of this object. However, contrastive pre-training on scene-centric datasets is known to yield models with less discriminative power, which ultimately underperform on downstream classification tasks [19].

Meanwhile, several foundation models for image segmentation have recently emerged, like SAM [17] and SEEM [25], which both enable to divide any image into meaningful semantic regions. Our main idea is to use these segmentation models to perform object-level contrastive learning regardless of the the type of dataset used for pre-training. For a given scene, we propose to first use SAM to identify the different regions of interest in the image, then select one of those regions and sample the views inside of it, rather than the whole image (see Figure 1). This simple trick enforces that

*Corresponding author

NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice

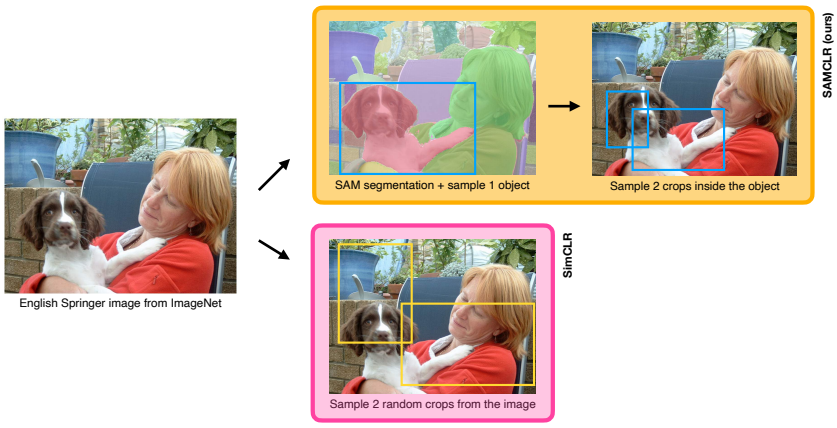


Figure 1: View sampling process for SimCLR (bottom) and SAMCLR (ours, top). Using SAM to segment out the image avoids sampling views which represent different object categories, which in turn helps denoising the contrastive learning procedure.

the different views are part of the same object. This idea derives from SimCLR [4], which repels the views coming from different images. By doing so, SimCLR gives every image a different label, and since it is typically trained on ImageNet, it can be seen as giving a different label to each object. Thus, we aim at generalizing the SimCLR framework to complex scenes, by performing object-level contrastive learning regardless of the number of objects originally present in the scenes.

Contribution Based on these observations, we propose **SAMCLR** (**SAM** + Sim**CLR**), a variation of SimCLR which first leverages SAM to segment the input image into semantic regions, then samples the views from one of these regions. By doing so, SAMCLR ensures that different views from the same image represent the same object, which helps denoising the learning process on complex scenes with multiple objects. When pre-training on datasets like Cityscapes [7] and ADE20K [23], and evaluating the features on image classification on CIFAR-10 [18], STL10 [6] and ImageNette [14], SAMCLR performs at least on par with, and most often significantly outperforms not only SimCLR, but also other SSL baselines, like DINO [3] and MoCo [12].

2 Method

In this section, we first briefly review SAM and SAMCLR, and then detail our method which uses SAM to sample the views for SimCLR to learn from.

2.1 Preliminaries

SAM Let $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ be an image where H, W and C are the height, width and number of channels respectively. SAM first passes $\mathbf{x}$ though a Vision Transformer encoder [9] f_θ in order to get an image embedding $z_{img} = f_\theta(\mathbf{x})$. Then, a prompt encoder embeds the input prompts (clicks, box, text, mask...), we denote z_{pr} this embedding. Finally, a transformer decoder $g_{\theta'}$ predicts a binary segmentation mask $\hat{y} = g_{\theta'}(z_{img}, z_{pr})$, with $\hat{y} \in \mathbb{R}^{H \times W}$. When prompted with a regular grid of points, SAM will output a list of binary masks denoting different regions of the input image $\mathbf{x}$. However, these regions are *not* mutually exclusive and in most cases a few coarse regions will encompass all of the others. In our case, we are only interested in having a rough idea of the different elements of the scene. Thus, we only focus on the coarsest regions and we drop the others as they represent higher level of granularity. Then, we randomly select one of those regions (to get one object), and sample 2 views from it.

SimCLR SimCLR attempts to enforce consistent representations between two augmentations from the same image, and does so by performing contrastive learning between images from the same batch. Given a batch of n images $X = \{\mathrm{x}^{(1)}, \dots, \mathrm{x}^{(n)}\}$, SimCLR applies 2 random cropping and color jittering operations to each image. We denote $\tilde{\mathrm{x}}_1^{(i)}$ and $\tilde{\mathrm{x}}_2^{(i)}$ the views from image $\mathrm{x}^{(i)}$ and X_{aug} the

resulting augmented batch $X_{aug} = \{\tilde{\mathrm{x}}_1^{(1)}, \tilde{\mathrm{x}}_2^{(1)}, \ldots, \tilde{\mathrm{x}}_1^{(n)}, \tilde{\mathrm{x}}_2^{(n)}\}$, with $|X_{aug}| = 2n$. Then, a feature extractor f_θ and a MLP head $g_{\theta'}$ project each view to a feature vector $z_k^{(i)} = g_{\theta'}(f_\theta(\mathrm{x}_k^{(i)}))$. Finally, the NT-Xent loss [20] brings closer the feature vectors of views from the same image, and repels the representations from the other images. Equation 1 gives the loss for a pair of positive embeddings z_i, z_j.

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2n} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_j)/\tau)}, \text{ with } \mathrm{sim}(u, v) = \frac{u^T v}{||u|| \cdot ||v||} \tag{1}$$

2.2 SAMCLR: Instance-level contrastive learning

Our idea is very simple and is explained in Figure 1. Contrarily to SimCLR which samples 2 random views anywhere in the image, we first segment the image with SAM, then select randomly one of the semantic regions, and sample 2 views from it. This ensures that the two views represent the same object. Then, as in SimCLR, these 2 views undergo color jittering before being fed to the feature extractor, and before computing the distance to the embeddings of the other views from the batch.

However, the randomness in SimCLR's view sampling process has the advantage of producing very diverse views. Limiting the sampling to views inside of objects would prevent our method from picking crops near object borders, which hinders variety. Additionally, padding around objects provides context that could help understanding. Thus, before selecting one object region, we multiply the height and width of each region produced by SAM by a constant factor c. We set $c = 1.3$ for the rest of the paper.

3 Experiments

In this section, we evaluate the effectiveness of our method by pre-training SAMCLR, as well as three other SSL baseines (SimCLR, DINO and MoCo) on either CityScapes or ADE20K, then evaluating the resulting model on three classification datasets: CIFAR-10, STL10 and ImageNette. For the downstream classification task, we adopt two standard testing protocols, i.e. linear probing and KNN classification.

Implementation details The pre-training of all SSL baselines is done with a ResNet18 [11] with batch size 256. We use the implementations of SimCLR, DINO and MoCo available in Lightly [21], with default optimizers and hyperparameters. Our SAMCLR's implementation follows the one from SimCLR, and incorporates our view sampling procedure (see Sec. 2.2). In practice, we perform the segmentation with HQ-SAM [16], a variant of SAM which produces slightly more precise and accurate masks. In order to speed up the view sampling process, we pre-compute and store the segmentation masks for all images of all datasets which will be used for pre-training. We also drop all regions with an area smaller than 1000px to avoid sampling views that would be too small. The temperature for SAMCLR is set to $\tau = 0.07$ as in MoCo. For SimCLR, SAMCLR and MoCo, the views are all resized to 128x128px. For DINO, the 2 global crops and 6 local crops are resized to 128x128px and 64x64px respectively. We run all experiments using a single RTX 4090 GPU with 24G memory.

Pre-training on Cityscapes We start with Cityscapes, which contains images taken on the roads of 50 different cities. The images usually contain a lot of information, like other vehicles, pedestrians, traffic signs or buildings. We pre-train all SSL methods for 1600 epochs, using the 2975 training images and we evaluate on image classification.

Table 1 sums up the results. We first note that the scores for all models are low overall. We believe this to be due to two main reasons. First, and as suggested before, contrastive pre-training is expected to fail on datasets which contain complex images (with more than 1 object), as views from the same image are no longer guaranteed to represent the same subject. On the other hand, the image distribution in Cityscapes is vastly different from the datasets used for evaluation. For example, ImageNette contains images of golf balls, parachutes and French horns, which are more than unlikely to appear in Cityscapes. Nevertheless, SAMCLR systematically improves over SimCLR by large

Table 1: KNN Top-1 accuracy and Linear probing for various SSL methods when pre-trained on Cityscapes and evaluated on image classification on CIFAR-10, STL10 and ImageNette. In this setting, SAMCLR systematically and significantly improves SimCLR and the other baselines.

	KNN			Linear		
	CIFAR-10	STL10	ImageNette	CIFAR-10	STL10	ImageNette
MoCo	44.0	41.2	45.4	43.2	40.1	46.8
DINO	38.8	36.7	40.8	37.1	34.7	40.4
SimCLR	50.5	46.9	50.5	42.5	40.8	42.7
SAMCLR (ours)	**63.9**	**57.2**	**66.3**	**63.4**	**56.0**	**67.5**

Table 2: KNN Top-1 accuracy and Linear probing for various SSL methods when pre-trained on ADE20K and evaluated on image classification on CIFAR-10, STL10 and ImageNette. SAMCLR systematically improves SimCLR but looses to MoCo on STL10 and ImageNette.

	KNN			Linear		
	CIFAR-10	STL10	ImageNette	CIFAR-10	STL10	ImageNette
MoCo	63.1	**61.9**	**71.8**	60.7	**60.7**	**73.5**
DINO	52.4	55.8	65.4	48.9	51.7	62.8
SimCLR	60.6	59.1	68.8	57.0	54.5	66.5
SAMCLR (ours)	**65.5**	59.3	69.1	**65.6**	59.8	70.8

margins (+21.9, +15.2 and +24.8 points on CIFAR-10, STL10 and ImageNette linear probing accuracy respectively).

Pre-training on ADE20K We then follow the same procedure to pre-train on ADE20K, which contains 20,210 training scenes from everyday life. We set the number of epochs to 250. The results are shown in Table 2. Once again, SAMCLR brings significant performance improvements over SimCLR (+8.6, +5.3 and +4.3 points on CIFAR-10, STL10 and ImageNette linear probing accuracy respectively). These gains make it outperform MoCo on CIFAR-10, and close the gap on the other benchmarks. We note that this time, all models perform better compared to the pretraining on Cityscapes. This is easily explained by ADE20K having a higher volume of images (20K compared to 3K), and a distribution closer to the downstream tasks. For example, ADE20K contains airplane images, which are very unlikely to appear in Cityscapes whereas *airplane* is actually one of the class labels for both CIFAR-10 and STL10.

4 Future work

This paper presents preliminary results on the use of SAM (and foundation models more broadly) to support the pretraining of other unsupervised methods. While we introduced SAMCLR as an add-on for SimCLR, it would be interesting to see if other contrastive learning approaches also benefit from it. Since SAMCLR only affects the view sampling process, it would be easily to integrates it into other baselines, especially MoCo which is very similar to SimCLR. Furthermore, pre-training SSL approaches on scene-centric datasets has shown to yield better feature extractors for downstream dense prediction tasks, like object detection or image segmentation [19]. It would be interesting to see if SAMCLR follows this trend and outperforms the other baselines in such benchmarks.

5 Conclusion

In this paper, we introduced a simple view sampling strategy, coined SAMCLR, that can be plugged into SimCLR to improve its learning capability on datasets with complex scenes. SAMCLR relies on SAM to ensure coherence and consistency between the views sampled from the same image. We validated the effectiveness of our method through experiments on Cityscapes and ADE20K, two datasets originally designed for segmentation tasks. We showed empirically that in these settings, SAMCLR systematically improves over SimCLR, and most often by a significant margin.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

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

[3] 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 International Conference on Computer Vision (ICCV)*, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[6] Adam Coates, A. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, 2011. URL `https://cs.stanford.edu/~acoates/stl10/`.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] 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*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[14] Jeremy Howard. Imagenette. URL `https://github.com/fastai/imagenette/`.

[15] Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *CoRR*, abs/2011.11765, 2020. URL `https://arxiv.org/abs/2011.11765`.

[16] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv:2306.01567*, 2023.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

[19] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[20] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. NIPS'16, 2016.

[21] Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, and Malte Ebner et al. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

[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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

[24] Wentao Zhu, Jingya Liu, and Yufang Huang. Hnssl: Hard negative-based self-supervised learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4778–4787, 2023. doi: 10.1109/CVPRW59228.2023.00506.

[25] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once, 2023.

A Supplementary Material

A.1 Pre-training on an object-centric dataset (ImageNette)

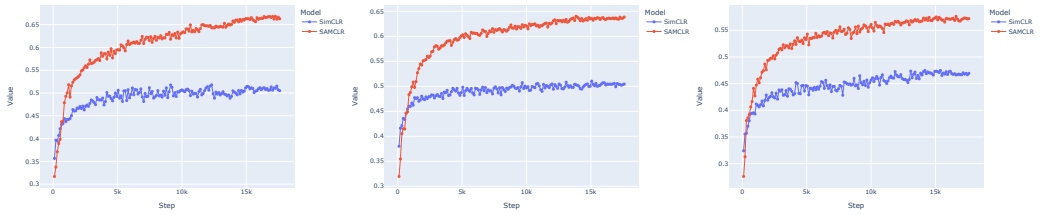
While we aim at designing a SSL approach that enables pre-training on scene-centric datasets, it remains interesting to see how SAMCLR performs when pre-trained on an image classification dataset. Thus, we pre-train and evaluate both SAMCLR and the other SSL baselines on ImageNette with the same experimental setting as in Section 3. We set the number of epochs to 800. Table 3 shows the results. We notice that SAMCLR underperforms in this setting. With an image classification dataset, we believe that our view sampling method hinders the variety of possible views fed to the contrastive loss, preventing the model from generalizing.

Table 3: KNN Top-1 accuracy and Linear probing for various SSL baselines when pre-trained and evaluated on ImageNette.

	KNN	Linear
SwaV	**89.5**	**91.9**
DINO	85.9	88.6
SimCLR	89.0	88.5
SAMCLR	85.1	85.8

A.2 SimCLR vs SAMCLR - Learning curves

To give a better idea of how SAMCLR fares against SimCLR and gauge the influence of our view sampling process, we display in Figure 2 the evolution of KNN classification accuracy as a function of the number of steps for both SAMCLR and SimCLR. Both models are pretrained on Cityscapes. The corresponding final results are given in Table 1. SAMCLR basically outperforms SimCLR throughout the whole learning process.



(a) (b) (c)

Figure 2: KNN accuracy vs optimization steps for SimCLR (blue) and SAMCLR (ours, red), pre-trained on Cityscapes and evaluated on ImageNette (a), CIFAR-10 (b) and STL10 (c).

We also note that the accuracy for SAMCLR is still seemingly going up at the end of training (1600 epochs). We thus attempt to train it for longer (3200 epochs) and observe an about 1.5 point increase on ImageNette KNN accuracy. We explain SAMCLR's appetite for longer training by the fact that, for each image, we sample the views from one object. Thus, over one epoch, the model will only see a single object of each image. It will therefore take longer for the model to see all objects of all images. SimCLR, on the other end, samples the views randomly, and thus has more chances to sample views with several objects, and it may even sample large crops that would reveal a large part of the original image.

A.3 Related works

Self-supervised learning of image features. Many different approaches have been proposed for learning good representations from an unlabeled set of images. This paper draws inspiration from the contrastive learning literature [4, 12, 3, 10, 5], which enforces similar representations between two or more views of the same image. Every method brings its own novelty, like BYOL [10] which does

not use negative samples. Other approaches include clustering, like [1, 2]. In particular, SwaV [2] performs online clustering, making it one of the few clustering methods to be easily trainable at scale.

Sample selection and choice of dataset Recent work [19] shows that the performance of current SSL schemes heavily depends on the dataset used for pretraining. For example, a MoCo [12] trained on MSCOCO leads to less discriminative power and lower classification accuracy than when pretrained on MSCOCO Boxes (a dataset they obtain by cropping the annotated bounding boxes from COCO). This observation motivated the current paper. While some other recent work has been tackling the view sampling process in contrastive learning [15, 24], we believe that the recent development of foundational segmentation models opens up new possibilities in the field.