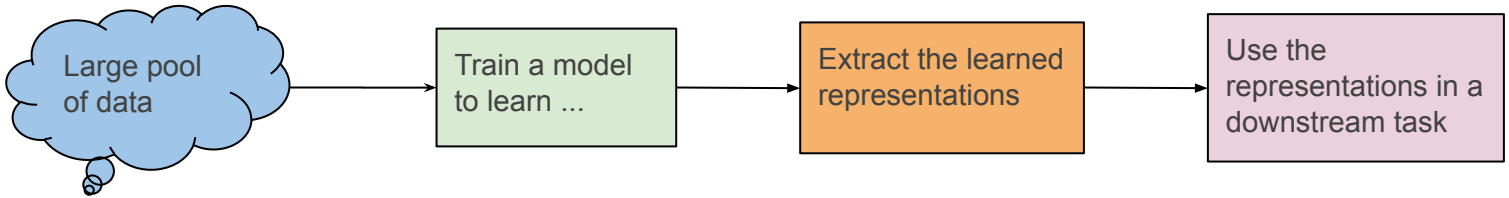
G-SimCLR: Self-Supervised Contrastive Learning with Guided Projection via Pseudo Labelling

Souradip Chakraborty*, Aritra Roy Gosthipaty*, Sayak Paul*

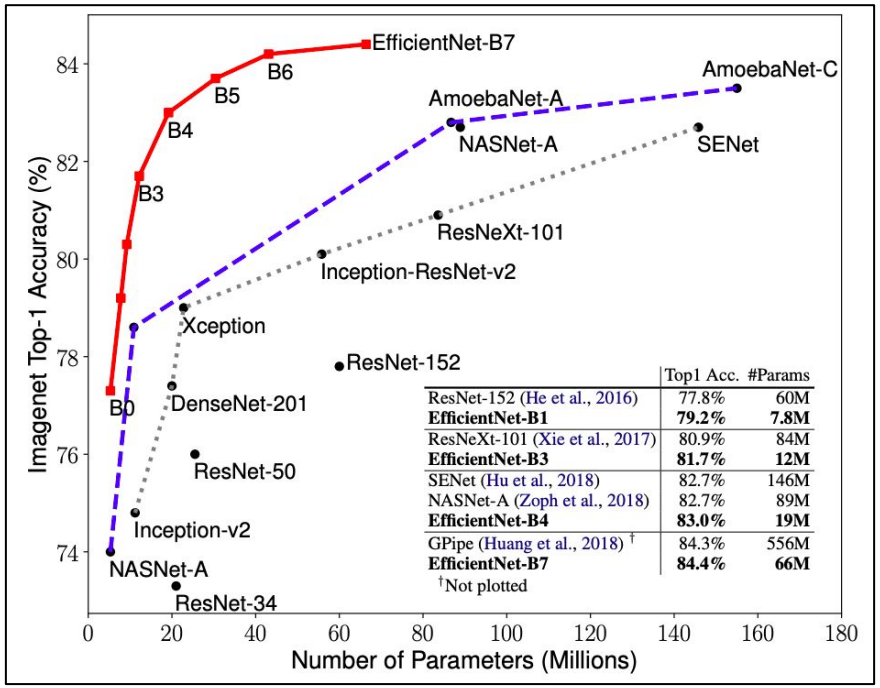
* Equal Contribution

Representation learning as a framework

Training models to learn representations for tasks like image classification, object detection, semantic segmentation, and so on.



The unreasonable success of supervised representation learning

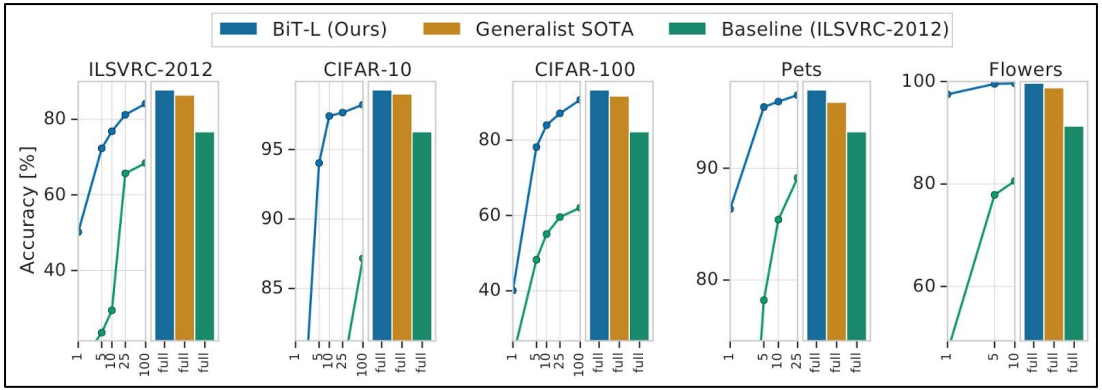


Source: EfficientNet; Tan et al. (2019)

But ...

Often, this success is constrained by

- The amount of labeled data for pre-training

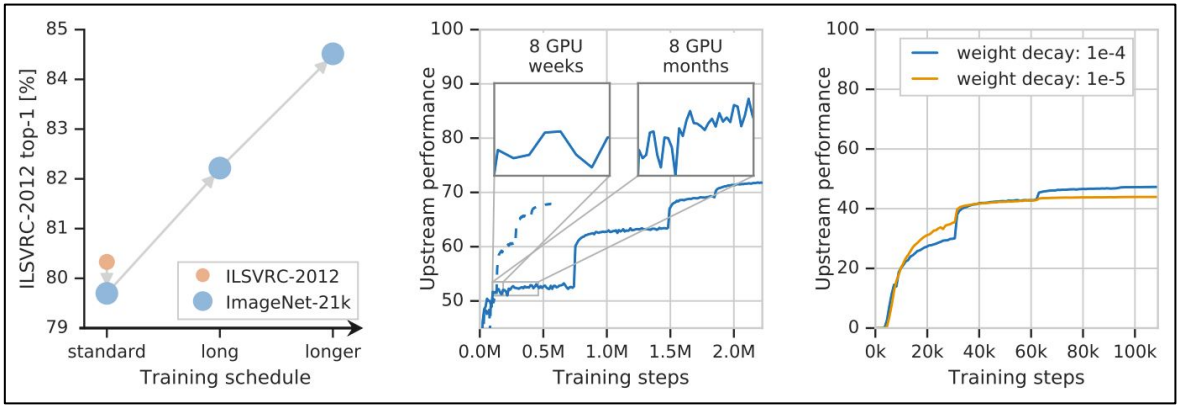


Source: Big Transfer (BiT); Kolesnikov et al. (2020)

But ...

Often, this success is constrained by

- The amount of labeled data for pre-training
- Length of training time



Source: Big Transfer (BiT); Kolesnikov et al. (2020)

But ...

Gathering large amount of labeled data

- Is costly
- Can be faulty

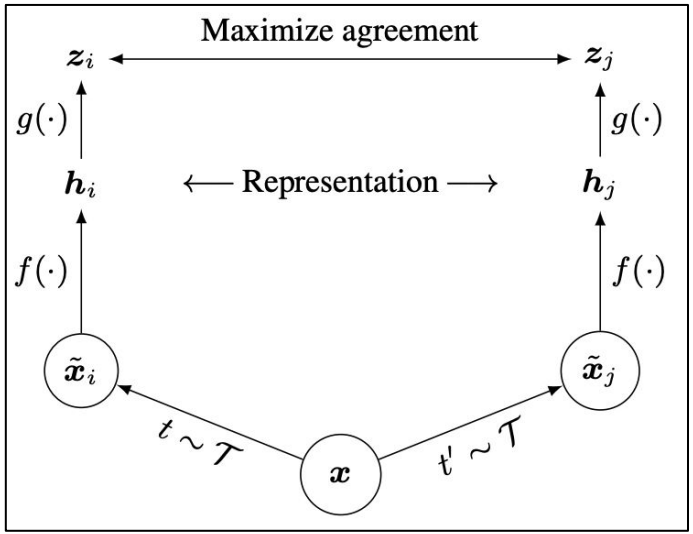
Self-supervised learning - Intro

- Constructing a supervised signal from ***unlabeled*** data
- Training models on that signal
- Using representations learned by these models for downstream tasks
- These supervised formulations are known as ***pretext tasks***

Pretext task - Examples

- Predicting the next word from a sequence of words
- Predicting a masked word in a sentence
- Predicting the angle of rotations in images
- Predicting the next frame in a video
- Filling out missing pixels in images

SimCLR

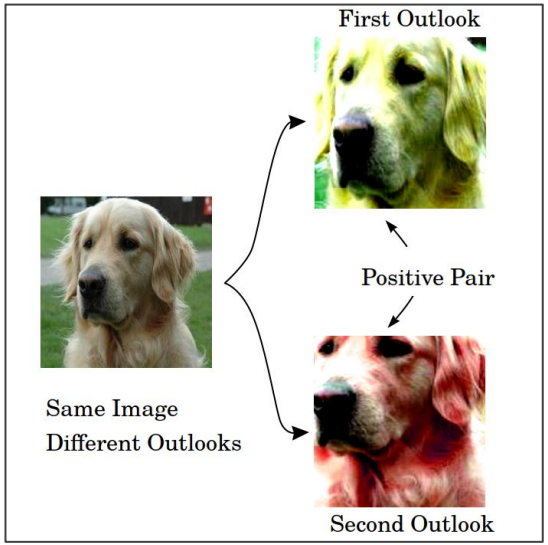


Source: SimCLR; Chen et al. (2020)

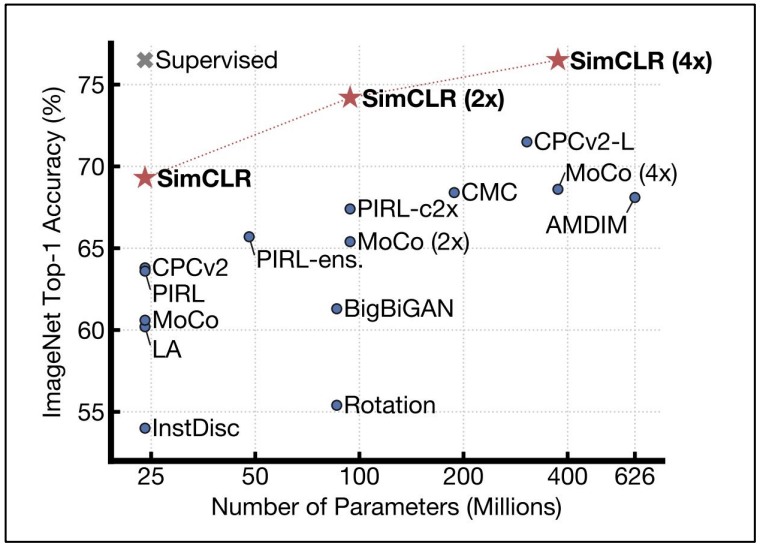
- Form different views with data augmentation techniques.
- Contrast different views of images with ***NT-Xent loss***.
- Pull together the views coming from the same images.

SimCLR and its prowess

Contrasting between different views of the same images works very well!



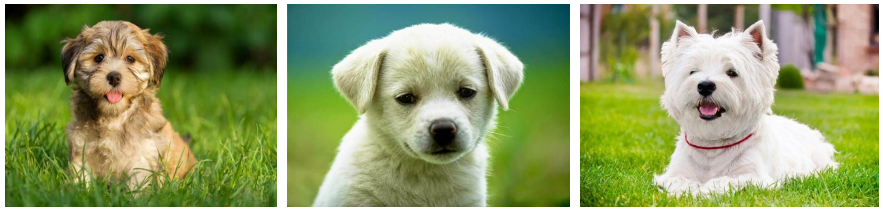
Source: [MoCo-V2 in PyTorch]



Source: SimCLR; Chen et al. (2020)

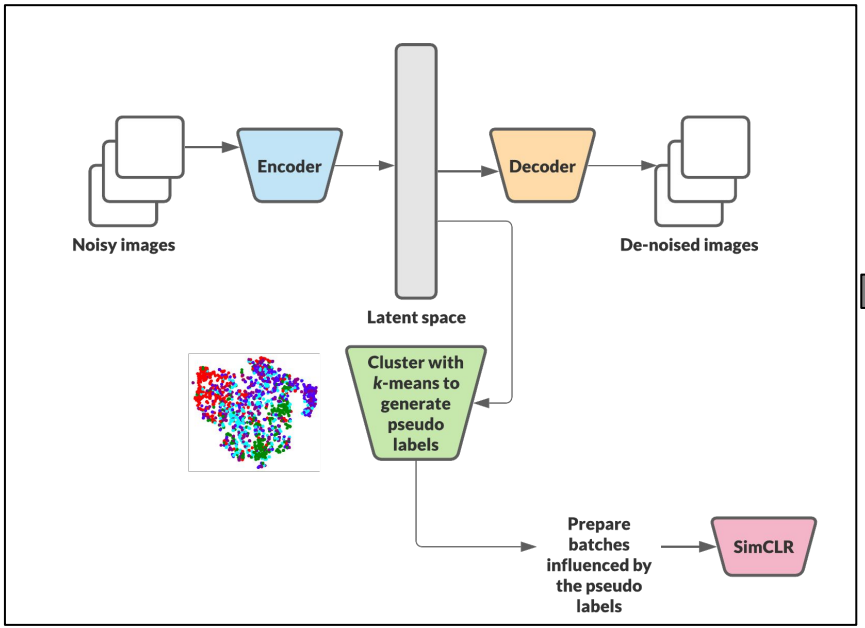
Potential drawbacks

- Relies on pretty large batch size for negative samples.
- Does not rigorously ensure two similar images would be closer in the feature space.
- Images from the same category might be in the same batch and treated as different image.
- Computationally intensive.



Images from the same category (Dog) are treated as different semantic images in the same batch, as per SimCLR.

G-SimCLR framework (ours)



In our work, we present an additional step on top of the SimCLR framework to ensure two semantically similar images do not get treated differently.

1. First, we train a denoising autoencoder (fully convolutional) on the given dataset.
2. In the second step, we take the encoder-learned robust representations and use k-means to get initial cluster assignments of those representations. We refer to these cluster assignments as pseudo labels for the given dataset.
3. Finally, we use these pseudo labels to prepare the batches and use them for SimCLR training.

Denoising autoencoder

- In the first step of our methodology, a denoising autoencoder is used which enhances the robustness of the latent feature representation when compared to a vanilla autoencoder.
- This is done by distorting the input image (x) to get a semi-distorted version of the same by means of Gaussian noise and the corrupted input is then mapped, as in case of a vanilla autoencoder, to a latent representation.
- The latent encoder representation now helps in reconstructing the distortion-free image input in the decoding phase and the average reconstruction loss is minimized to learn the parameters of the denoising autoencoder.

K-Means for pseudo-labels

- By applying k-means clustering algorithm on the latent space representations of images, we retrieve the cluster labels for each image.
- The choice for the number of clusters is 64. Empirically, we found out that it is helpful to have a cluster number that is large enough to capture the implicit marginal distribution of the given dataset.
- One could also use domain knowledge or some informative priors to determine this value.

Creating batches for G-SimCLR

- In the final step of our methodology, instead of randomly sampling the mini-batch of samples, we use the pseudo labels obtained by k-means clustering on the autoencoder representations to sample the images in each of the batches.
- In our current implementation, we have kept the batch size and cluster number to be the same so, while preparing the training batches, we perform random sampling stratified by the pseudo labels.
- k-means clustering does not ensure equal-sized clusters which means that some batches will have more from one cluster and less from others. This ensures that our method does not provide a hard constraint on the images of the batches to belong to different and discrete classes only.

Experimental results

- In our approach, we cluster the latent space representations of images into 64 clusters.
- The image representations close to each other fall into the same cluster, while the ones that are far away, belong to different clusters.
- From the t-SNE projections of the latent space representations (as learned by the denoising autoencoder), it is evident that the representations of similar images cluster together.

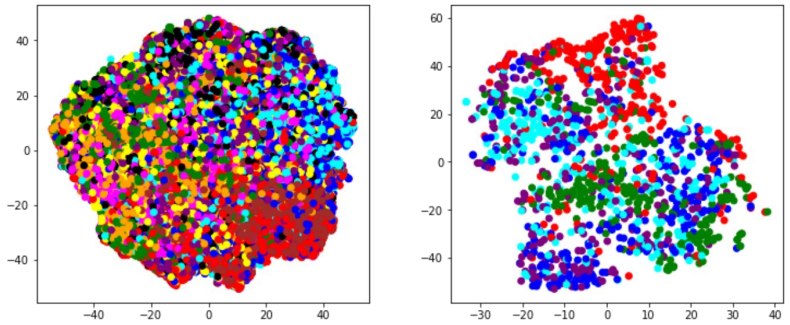


Figure : *t-SNE projections of the latent space representations learned by the denoising autoencoder on the CIFAR10 and ImageNet Subset datasets.*

Experimental results

- To compare G-SimCLR with our variant of SimCLR, we run two downstream supervised classification tasks on the visual representations that each of them learns.
- We first run linear evaluation on the classification task and later fine-tune the models with 10 % data and then evaluate again.

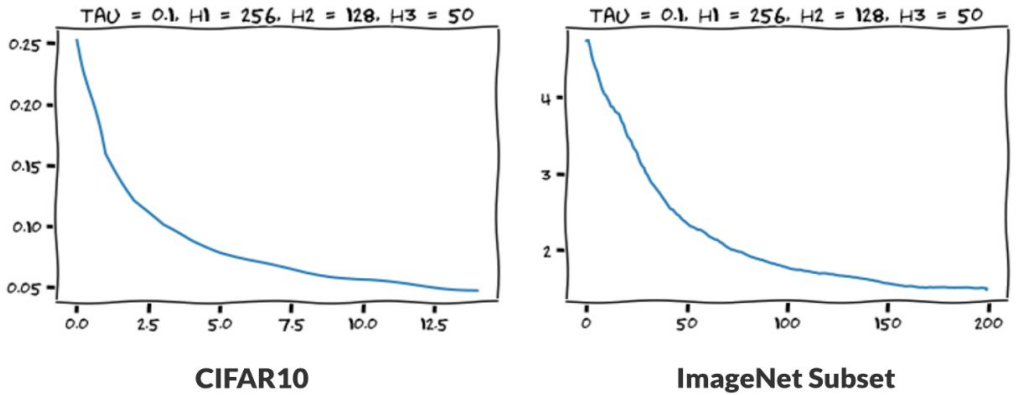


***Figure** : Loss (NT-Xent) curves as obtained from the G-SimCLR training with the CIFAR10 and ImageNet Subset datasets.*

Experimental results

For linear evaluation we report the validation accuracy for different levels of the feature backbone network and the projection head denoted as P1, P2 and P3.

Linear evaluation			
		CIFAR10	ImageNet Subset
Fully supervised		73.62	67.6
SimCLR with minor modifications	P1	37.69	52.8
	P2	39.4	48.4
	P3	39.92	52.4
G-SimCLR (ours)	P1	**38.15**	**56.4**
	P2	**41.01**	**56.8**
	P3	**40.5**	**60.0**

Figure *: Performance of the linear classifiers trained on top of the representations (kept frozen during training linear classifiers) learned by G-SimCLR.*

P1 : denotes the feature backbone network + the entire non-linear projection head - its final layer

P2 : denotes the feature backbone network + the entire non-linear projection head - its final two layers

P3 : denotes the feature backbone network only

Fine-tuning (10% labeled data)		
	CIFAR 10	ImageNet Subset
SimCLR with minor modifications	42.21	49.2
G-SimCLR (ours)	**43.1**	**56.0**

Figure *: Performance of weakly supervised classifiers trained on top of the representations learned by G-SimCLR. We fine-tuned the representations with only 10\% labeled data*

Future scope

1. More rigorous experimentations to study the effect of the number of number of clusters on the contrastive learning task and further downstream tasks has been left as a scope for future work.
2. Finding an optimal connection between the number of clusters and batch-size for better and robust performance of the system.
3. Developing a mechanism to include the clustering step in the Self-supervised learning framework for optimal performance and bias reduction.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[2] T.Chen, S.Kornblith, M.Norouzi, and G.Hinton,“A simple framework for contrastive learning of visual representations,” 2020.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.

[5] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” 2019

[6] T. H. Trinh, M.-T. Luong, and Q. V. Le, “Selfie: Self-supervised pretraining for image embedding,” 2019.