



**HAL**  
open science

# Contrastive Learning using Random Walk Laplacian Matrix

Ilyass Moummad, Bastien Pasdeloup, Nicolas Farrugia

► **To cite this version:**

Ilyass Moummad, Bastien Pasdeloup, Nicolas Farrugia. Contrastive Learning using Random Walk Laplacian Matrix. Graph Signal Processing Workshop, 2023, Oxford, United Kingdom. hal-04639769

**HAL Id: hal-04639769**

**<https://imt-atlantique.hal.science/hal-04639769>**

Submitted on 9 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Contrastive Learning using Random Walk Laplacian Matrix

Ilyass Moummad  
*IMT Atlantique, Lab-STICC,*  
*UMR CNRS 6285, F-2923*  
 Brest, France  
 ilyass.moummad@imt-atlantique.fr

Bastien Padeloup  
*IMT Atlantique, Lab-STICC,*  
*UMR CNRS 6285, F-2923*  
 Brest, France  
 bastien.padeloup@imt-atlantique.fr

Nicolas Farrugia  
*IMT Atlantique, Lab-STICC,*  
*UMR CNRS 6285, F-2923*  
 Brest, France  
 nicolas.farrugia@imt-atlantique.fr

## I. CONTEXT

In recent years, Self-Supervised Learning (SSL) has gained in popularity due to the availability of unlabeled data. SSL consists in training a neural network encoder capable of representing data in a low-dimensional space efficiently, *i.e.*, capable of extracting useful features from data (the one it is trained on, as well as new data). This problem of learning good representations is generally called *representation learning*. To this end, SimCLR [1] is a popular contrastive method that optimizes an encoder to output similar representations for different views of the same data (positive pairs), and different representations for views of different data (negative pairs).

In this work, we treat the representation learning problem as a random walk on a graph of data representations in the latent space (vertices) and their similarities (edges). This can be formulated as an optimization problem that maximizes the transition probability between views of the same data, and minimizes the transition probability between views of different data. This problem has been approximated in [2], which proposes a solution minimizing the sum of euclidean distances between positive pairs, and adds a decorrelation term to avoid representation collapse (*i.e.*, convergence to a trivial solution in which representations are constant). In this work, we propose a simpler loss function, that leverages the random walk Laplacian matrix directly. We benchmark our approach on the CIFAR10 dataset using standard data augmentations from the literature to create different views of data [3], and compare our results to SimCLR.

## II. METHOD

We model a graph of data augmentations using an adjacency matrix  $\mathbf{W} \in \mathbb{R}^{N \times N}$  between  $N$  vertices, each representing an augmented view of a sample in a given dataset to learn from<sup>1</sup>. Weights in  $\mathbf{W}$  denote positive similarities between samples, defined as :

$$\mathbf{W}_{i,j} = \begin{cases} \exp(f_\theta(\mathbf{x}_i)^\top f_\theta(\mathbf{x}_j)), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

This work was co-funded by the AI at IMT ANR project, as well as the company OSO AI.

<sup>1</sup>Data augmentations can be diverse, depending on the task and considered dataset. Those we consider in this work are detailed in Section V.

where  $f_\theta$  is a Neural Network of parameters  $\theta$  that encodes an image into an abstract representation in a lower dimensional space. From  $\mathbf{W}$ , we define the degrees matrix  $\mathbf{D} = \text{diag}\left(\left[\sum_{j=1}^N \mathbf{W}_{i,j}\right]_i\right)$  and the random walk Laplacian matrix  $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ . Here,  $\mathbf{P}_{i,j}$  represents the transition probability from view  $\mathbf{x}_i$  to view  $\mathbf{x}_j$ . The rationale behind our approach is as follows : we want  $\mathbf{P}_{i,j} = 1$  when  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are two views of the same data (noted  $\mathbf{x}_i \sim \mathbf{x}_j$ ), and  $\mathbf{P}_{i,j} = 0$  when they are views of different data (noted  $\mathbf{x}_i \not\sim \mathbf{x}_j$ ), similar to the philosophy of contrastive methods [1].

## III. TRAINING LOSS

Within a batch  $\mathcal{B}$ , we define the loss function  $\mathcal{L}$  between a view  $\mathbf{x}_i \in \mathcal{B}$  and a view  $\mathbf{x}_j \in \mathcal{B}$  as :

$$\mathcal{L}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{1}_{[i \sim j]}(1 - \mathbf{P}_{i,j}) + \mathbf{1}_{[i \not\sim j]}\mathbf{P}_{i,j}, \quad (2)$$

where  $\mathbf{1}_{[\cdot]}$  is the indicator function, equal to 1 when condition is verified and 0 otherwise. The batch loss  $\mathcal{L}_{\mathcal{B}}$  is the sum of losses  $\mathcal{L}$  computed for all pairs in the batch :

$$\mathcal{L}_{\mathcal{B}} = \sum_{\substack{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{B} \\ i \neq j}} \mathcal{L}(\mathbf{x}_i, \mathbf{x}_j). \quad (3)$$

Fig. 1 illustrates our method visually. In this small example, a batch of 4 images is used to generate a two-view batch of  $N = 8$  views  $\mathcal{B} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\}$ , where a same color represents a same original data from which augmentation is made. We define the random walk matrix  $\mathbf{P}$  of these views, then the loss  $\mathcal{L}_{\mathcal{B}}$  is computed using Eq. 3 to optimize the parameters  $\theta$  of  $f_\theta$ .

## IV. COMPARISON TO SIMCLR

SimCLR uses the normalized temperature-scaled cross entropy loss  $\mathcal{L}_{\text{SimCLR}}$  [1] defined as :

$$\mathcal{L}_{\text{SimCLR}}(\mathbf{x}_i, \mathbf{x}_j) = -\log \left( \frac{\exp(f_\theta(\mathbf{x}_i)^\top f_\theta(\mathbf{x}_j)/\tau)}{\sum_{k=1}^N \mathbf{1}_{[k \not\sim i]} \exp(f_\theta(\mathbf{x}_i)^\top f_\theta(\mathbf{x}_k)/\tau)} \right), \quad (4)$$

where  $\tau$  is a temperature parameter helping to discriminate hard examples.

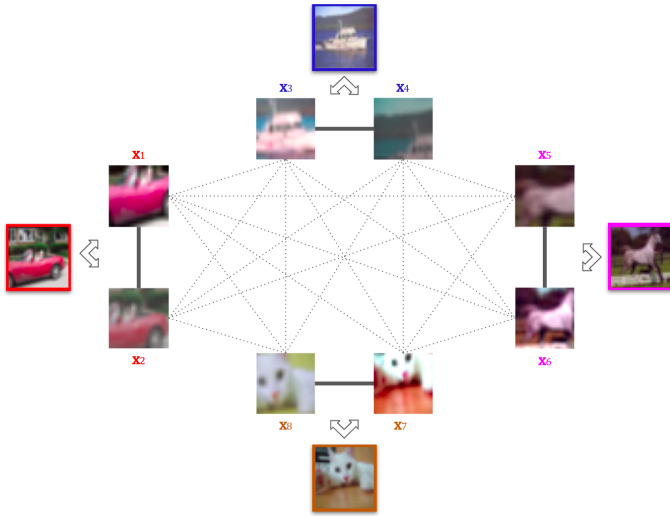


Fig. 1. Overview of the framework. Each image is augmented twice to produce two views. The edges represent probability transitions of pair of views. Thick edges represent high probability transitions (positive pairs) that are maximized toward 1, while dotted lines correspond to low probability transitions (negative pairs), minimized toward 0.

It is interesting to notice that our loss introduced in Eq. 2 is similar to  $\mathcal{L}_{SimCLR}$ , as it can be reformulated as follows using the cosine similarity used in the definition of  $\mathbf{W}$  in Eq. 1:

$$\mathcal{L}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{1}_{[i \sim j]} \left( 1 - \frac{\exp(f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_j))}{\sum_{k=1}^N \exp(f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_k))} \right) + \mathbf{1}_{[i \not\sim j]} \left( \frac{\exp(f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_j))}{\sum_{k=1}^N \exp(f_{\theta}(\mathbf{x}_i)^{\top} f_{\theta}(\mathbf{x}_k))} \right). \quad (5)$$

Due to the mutual exclusion of indicator functions, only one term of the sum is used for each pair  $(\mathbf{x}_i, \mathbf{x}_j)$ . The other main difference with  $\mathcal{L}_{SimCLR}$  is the presence of a log. Also, we do not use a temperature parameter in our similarity measure. Adding such a parameter may help and will be investigated in future work.

## V. EXPERIMENTS

For learning representations, we train the CIFAR variant of a ResNet18 backbone, followed by a 2-layer MLP, on CIFAR10, without labels, for 100 epochs. We use the following data augmentations: resized crop, horizontal flip, color jitter, and grayscale conversion. These are the same augmentations as [3], where hyperparameters can be found for the interested reader. We train our model using SGD optimizer with base learning rate = 0.06 and a cosine schedule for 100 epochs, weight decay = 0.0005, momentum = 0.9, and batch size = 512. After training the encoder, we freeze its parameters, and we evaluate the classification performance on CIFAR10 classes using a logistic regression, trained using SGD optimizer for 100 epochs with base learning rate = 0.1, momentum = 0.9, without weight decay and with batch size = 256.

Table I shows the performance obtained by our method compared to SimCLR in the settings described above. Our

TABLE I  
RESULTS OF LINEAR EVALUATION OF OUR METHOD COMPARED TO SIMCLR AND [2] ON CIFAR10. RESULTS ON [2] ARE THOSE REPORTED IN THEIR ARTICLE DUE TO UNAVAILABILITY OF THE CODE AND MISSING INFO ON TRAINING SETTINGS.

Method	Epochs	Accuracy
SimCLR (reproduced)	100	80.17 ± 0.20
Lee [2] (reported)	800	92.24
Ours	100	84.03 ± 0.29

approach shows competitive results against SimCLR [1], but seems to be outperformed by [2]. However, their results are reported from their article, and are not reproducible due to missing information or code. Besides, they train on 800 epochs, while we train on 100. We plan to extend our simulations to similar settings by the time of the conference.

## VI. PERSPECTIVES

Our first experiments show promising results in the integration of a new loss based on the random walk Laplacian matrix, on CIFAR10. In addition to the points already mentioned – studying addition of a parameter  $\tau$  and training for more epochs – we want to perform experiments on more advanced datasets, and explore new choices of similarity measures. Finally, we are interested in evaluating the ability of the representations learned using our loss function to be used in transfer learning for downstream tasks.

## REFERENCES

- [1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, 2020.
- [2] Kuilin Chen and Chi-Guhn Lee, “Unsupervised few-shot learning via deep laplacian eigenmaps,” *arXiv preprint arXiv:2210.03595*, 2022.
- [3] Xinlei Chen and Kaiming He, “Exploring simple siamese representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15750–15758.