Group Meeting - February 26, 2021

Paper review & Research progress

Truong Son Hy [*]

[*]Department of Computer Science
The University of Chicago

Ryerson Physical Lab

Content

- Literature review:
  - **A Simple Framework for Contrastive Learning of Visual Representations** (ICML 2020), https://arxiv.org/abs/2002.05709 (2D image)
  - **MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks**, https://arxiv.org/abs/2102.10056 (molecular graphs)
  - **Graph Contrastive Learning with Augmentations** (NeurIPS 2020), https://arxiv.org/pdf/2010.13902.pdf (molecular graphs)
  - **PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding**, https://arxiv.org/pdf/2007.10985.pdf (3D point cloud)

Paper 1

A Simple Framework for Contrastive Learning of Visual Representations (ICML 2020)

Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton

`https://arxiv.org/abs/2002.05709`

Overview

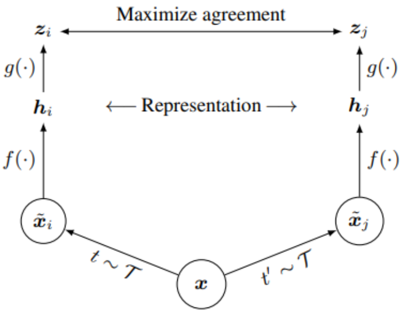
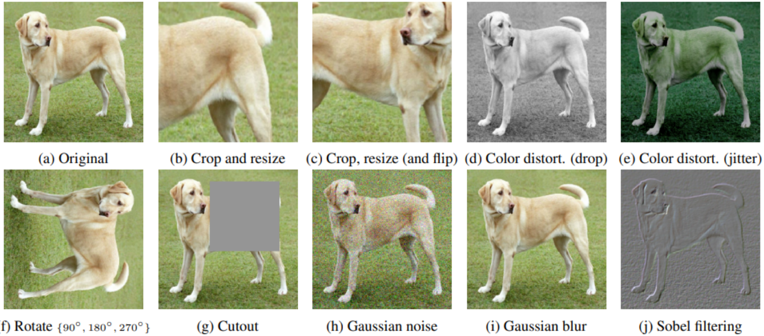


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

Augmentation



(a) Original

(b) Crop and resize

(c) Crop, resize (and flip)

(d) Color distort. (drop)

(e) Color distort. (jitter)

(f) Rotate $\{90^\circ, 180^\circ, 270^\circ\}$

(g) Cutout

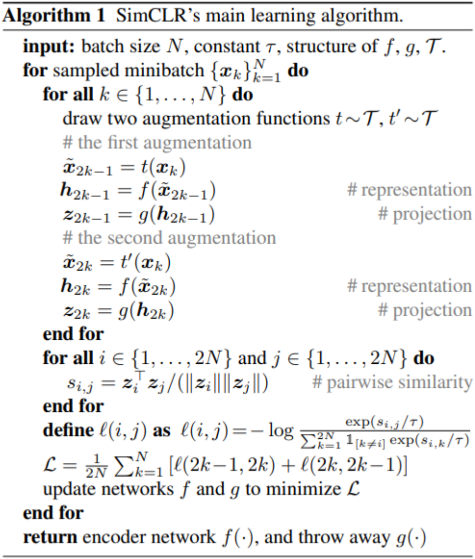
(h) Gaussian noise

(i) Gaussian blur

(j) Sobel filtering

Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

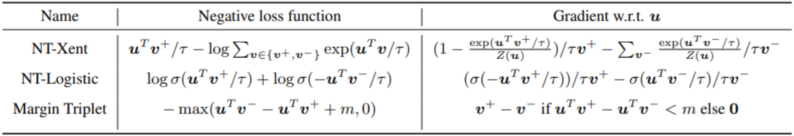
Algorithm



Algorithm 1 SimCLR's main learning algorithm.

```
input: batch size $N$, constant $\tau$, structure of $f$, $g$, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ do
    for all $k \in \{1, \ldots, N\}$ do
        draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
        # the first augmentation
        $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$          # representation
        $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$                  # projection
        # the second augmentation
        $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
        $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$              # representation
        $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$                      # projection
    end for
    for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ do
        $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$     # pairwise similarity
    end for
    define $\ell(i,j)$ as $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
    $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
    update networks $f$ and $g$ to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$
```

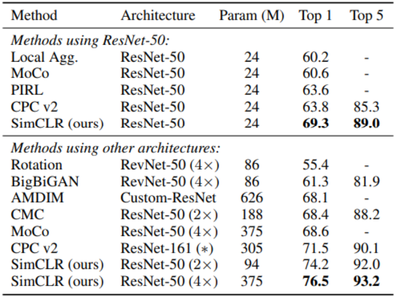
Loss candidates



Name	Negative loss function	Gradient w.r.t. $\boldsymbol{u}$
NT-Xent	$\boldsymbol{u}^T\boldsymbol{v}^+/\tau - \log\sum_{\boldsymbol{v}\in\{\boldsymbol{v}^+,\boldsymbol{v}^-\}}\exp(\boldsymbol{u}^T\boldsymbol{v}/\tau)$	$(1-\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^+/\tau)}{Z(\boldsymbol{u})})/\tau\boldsymbol{v}^+ - \sum_{\boldsymbol{v}^-}\frac{\exp(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)}{Z(\boldsymbol{u})}/\tau\boldsymbol{v}^-$
NT-Logistic	$\log\sigma(\boldsymbol{u}^T\boldsymbol{v}^+/\tau) + \log\sigma(-\boldsymbol{u}^T\boldsymbol{v}^-/\tau)$	$(\sigma(-\boldsymbol{u}^T\boldsymbol{v}^+/\tau))/\tau\boldsymbol{v}^+ - \sigma(\boldsymbol{u}^T\boldsymbol{v}^-/\tau)/\tau\boldsymbol{v}^-$
Margin Triplet	$-\max(\boldsymbol{u}^T\boldsymbol{v}^- - \boldsymbol{u}^T\boldsymbol{v}^+ + m, 0)$	$\boldsymbol{v}^+ - \boldsymbol{v}^-$ if $\boldsymbol{u}^T\boldsymbol{v}^+ - \boldsymbol{u}^T\boldsymbol{v}^- < m$ else $\mathbf{0}$

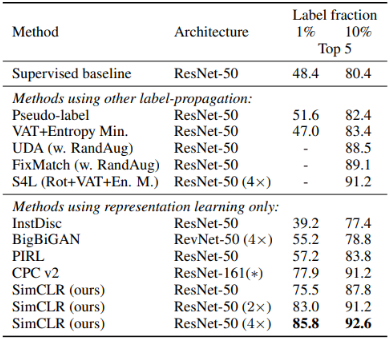
Table 2. Negative loss functions and their gradients. All input vectors, i.e. $\boldsymbol{u}, \boldsymbol{v}^+, \boldsymbol{v}^-$, are ℓ_2 normalized. NT-Xent is an abbreviation for "Normalized Temperature-scaled Cross Entropy". Different loss functions impose different weightings of positive and negative examples.

Experiments



Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.



Method	Architecture	Label fraction	
		1%	10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Paper 2 & 3

MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks

Yuyang Wang, Jianren Wang, Zhonglin Cao, Amir Barati Farimani

https://arxiv.org/abs/2102.10056

Graph Contrastive Learning with Augmentations (NeurIPS 2020)

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, Yang Shen

https://arxiv.org/pdf/2010.13902.pdf

Proposal

Proposal

1. A **self-supervised** learning framework for molecular representation learning.
2. Three molecular graph augmentation strategies to generate contrastive pairs:
   - Atom masking.
   - Bond deletion.
   - Subgraph removal.
3. Able to achive SOTA on several downstream molecular classification tasks.

Overview (1)

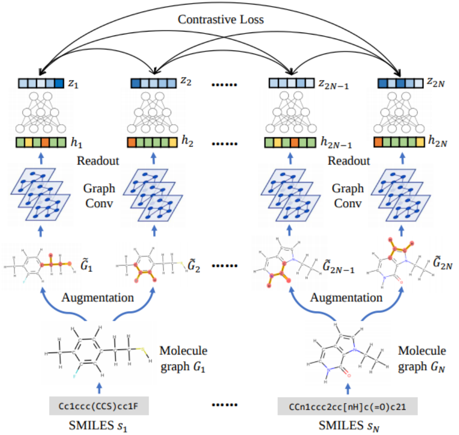


Figure 1: Molecular Contrastive Learning of Representations via Graph Neural Networks. A SMILES s_n from a mini-batch of N molecule data is converted to a molecule graph G_n. Two stochastic molecule graph data augmentation operators are applied to each graph, resulting two correlated masked graphs: $\tilde{G}_{2n-1}$ and $\tilde{G}_{2n}$. A base feature encoder built upon graph convolutions and the readout operation extracts the representation h_{2n-1}, h_{2n}. Contrastive loss is utilized to maximize agreement between the latent vectors z_{2n-1}, z_{2n} from the MLP projection head.

Overview (2)

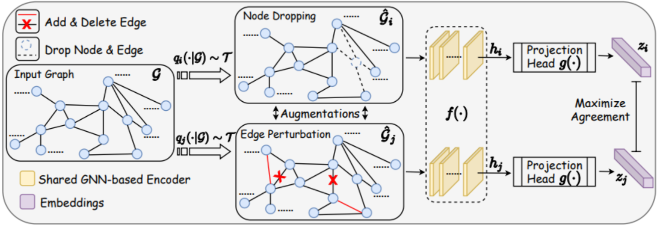


Figure 1: A framework of graph contrastive learning. Two graph augmentations $q_i(\cdot|\mathcal{G})$ and $q_j(\cdot|\mathcal{G})$ are sampled from an augmentation pool $\mathcal{T}$ and applied to input graph $\mathcal{G}$. A shared GNN-based encoder $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize the agreement between representations z_i and z_j via a contrastive loss.

Contrastive learning

- Contrastive learning aims at learning represetation through contrastive positive data pairs against negative ones.
- **SimCLR** demonstrates contrastive learning can greatly benefits from the composition of data augmentations and large batch sizes.
- Based on InfoNCE, **SimCLR** proposes the normalized temperature-scaled cross entropy (NT-Xent) loss:

$$\mathcal{L}_{i,j} = \log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} 1\{k \neq i\} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are latent vectors extracted from a positive data pair, N is the batch size, τ is the temperature parameter, and sim(.) measures the similarity between the two vectors (e.g. cosine):

$$\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{\boldsymbol{z}_i^T \boldsymbol{z}_j}{||\boldsymbol{z}_i||_2 ||\boldsymbol{z}_j||_2}$$

MolCLR Framework (1)

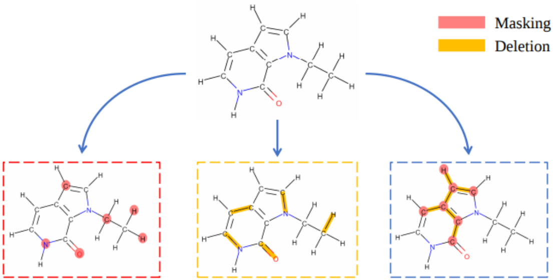
Molecular graph data augmentation strategies:

1. **Atom Masking:** Atoms in the graph are randomly masked with a given ratio (e.g. atom features $\boldsymbol{x}_v$ is replaced by a mask token $\boldsymbol{m}$).
2. **Bond Deletion:** Randomly removes edges completely out of the graph.
3. **Subgraph Removal:** Subgraph removal starts from a randomly picked origin atom. The removal process is implemented in DFS manner.

Algorithm:

- Given a mini-batch of size N, a molecular graph G_n is transformed into two different but correlated molecular graphs $\tilde{G}_i$ and $\tilde{G}_j$ where $i = 2n - 1$ and $j = 2n$.
- Molecular graphs augmented from the same molecule are denoted as positive pairs. From different molecules, negative pairs.

MolCLR Framework (2)



(a) Atom masking (b) Bond deletion (c) Subgraph removal

Figure 2: Three molecule graph augmentation strategies. (a) **Atom masking** randomly replaces the node feature $\boldsymbol{x}_v$ of an atom feature with a mask token $\boldsymbol{m}$. (b) **Bond deletion** randomly deletes the bond between two atoms, so that the they are not directly connected on the graph. (c) **Subgraph removal** randomly removes an induced subgraph [66] from the original molecule graph. Within the subgraph, all nodes are masked and all edges are deleted.

Experiments (1)

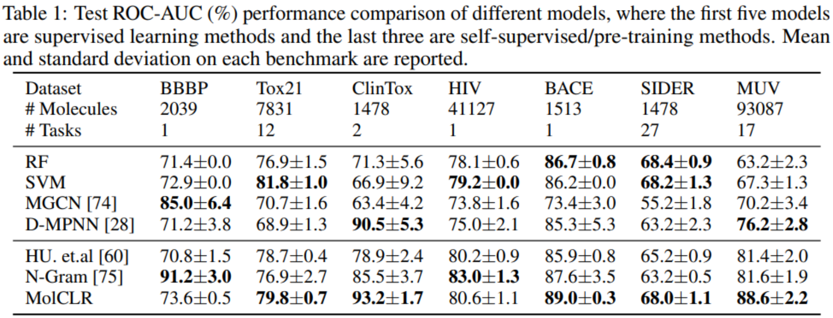


Table 1: Test ROC-AUC (%) performance comparison of different models, where the first five models are supervised learning methods and the last three are self-supervised/pre-training methods. Mean and standard deviation on each benchmark are reported.

Dataset # Molecules # Tasks	BBBP 2039 1	Tox21 7831 12	ClinTox 1478 2	HIV 41127 1	BACE 1513 1	SIDER 1478 27	MUV 93087 17
RF	71.4±0.0	76.9±1.5	71.3±5.6	78.1±0.6	**86.7±0.8**	**68.4±0.9**	63.2±2.3
SVM	72.9±0.0	**81.8±1.0**	66.9±9.2	**79.2±0.0**	86.2±0.0	**68.2±1.3**	67.3±1.3
MGCN [74]	**85.0±6.4**	70.7±1.6	63.4±4.2	73.8±1.6	73.4±3.0	55.2±1.8	70.2±3.4
D-MPNN [28]	71.2±3.8	68.9±1.3	**90.5±5.3**	75.0±2.1	85.3±5.3	63.2±2.3	**76.2±2.8**
HU. et.al [60]	70.8±1.5	78.7±0.4	78.9±2.4	80.2±0.9	85.9±0.8	65.2±0.9	81.4±2.0
N-Gram [75]	**91.2±3.0**	76.9±2.7	85.5±3.7	**83.0±1.3**	87.6±3.5	63.2±0.5	81.6±1.9
MolCLR	73.6±0.5	**79.8±0.7**	**93.2±1.7**	80.6±1.1	**89.0±0.3**	**68.0±1.1**	**88.6±2.2**

Experiments (2)

Table 2: Test ROC-AUC (%) performance comparison of different temperature parameter τ. Mean and standard deviation of all the seven benchmarks are reported.

Temperature (τ)	0.05	0.1	0.5
ROC-AUC (%)	76.8±1.2	80.2±1.3	78.4±1.7

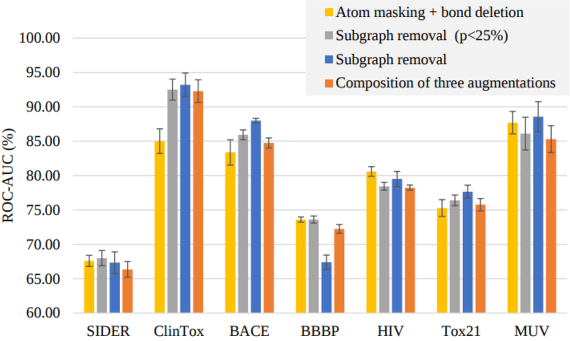


Figure 3: Test ROC-AUC (%) performance of pre-trained MolCLR model with different compositions of molecular graph augmentation strategies. Height of each bar represents the mean ROC-AUC on the benchmark, and length of each error bar represents the standard deviation.

Experiments (3)

Table 3: Test ROC-AUC (%) of GIN with/without molecule graph augmentations on all the seven supervised molecular classification benchmarks. GIN models are trained in the supervised learning manner without pre-training.

Dataset	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV
GIN w/o Aug	65.8±4.5	74.0±0.8	58.0±4.4	75.3±1.9	70.1±5.4	57.3±1.6	71.8±2.5
GIN w/ Aug	72.1±0.9	75.0±1.1	64.0±2.4	76.1±1.2	71.6±0.7	65.2±1.4	80.5±3.1

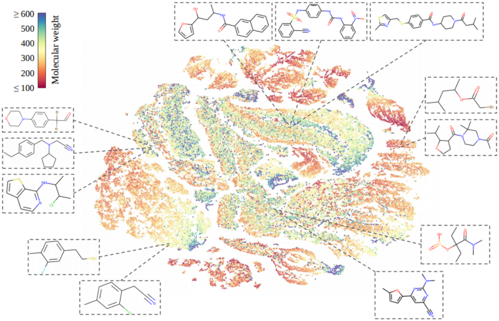
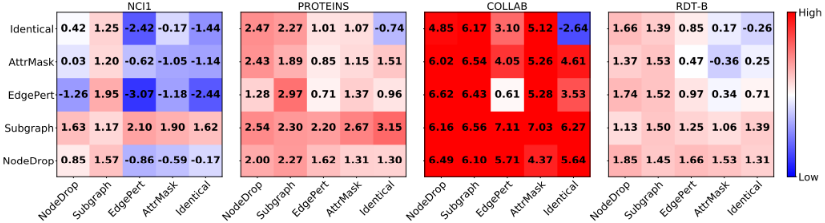


Figure 4: Two-dimensional t-SNE embedding of the molecular representations learned by our MolCLR pre-training. Representations are extracted from the validation set of the pre-training dataset, which contains 100k unique molecules. The color of each embedding point indicates its corresponding molecular weight.

Experiments (4)



NCI1

	NodeDrop	Subgraph	EdgePert	AttrMask	Identical
Identical	0.42	1.25	-2.42	-0.17	-1.44
AttrMask	0.03	1.20	-0.62	-1.05	-1.14
EdgePert	-1.26	1.95	-3.07	-1.18	-2.44
Subgraph	1.63	1.17	2.10	1.90	1.62
NodeDrop	0.85	1.57	-0.86	-0.59	-0.17

PROTEINS

	NodeDrop	Subgraph	EdgePert	AttrMask	Identical
Identical	2.47	2.27	1.01	1.07	-0.74
AttrMask	2.43	1.89	0.85	1.15	1.51
EdgePert	1.28	2.97	0.71	1.37	0.96
Subgraph	2.54	2.30	2.20	2.67	3.15
NodeDrop	2.00	2.27	1.62	1.31	1.30

COLLAB

	NodeDrop	Subgraph	EdgePert	AttrMask	Identical
Identical	4.85	6.17	3.10	5.12	-2.64
AttrMask	6.02	6.54	4.05	5.26	4.61
EdgePert	6.62	6.43	0.61	5.28	3.53
Subgraph	6.16	6.56	7.11	7.03	6.27
NodeDrop	6.49	6.10	5.71	4.37	5.64

RDT-B

	NodeDrop	Subgraph	EdgePert	AttrMask	Identical
Identical	1.66	1.39	0.85	0.17	-0.26
AttrMask	1.37	1.53	0.47	-0.36	0.25
EdgePert	1.74	1.52	0.97	0.34	0.71
Subgraph	1.13	1.50	1.25	1.06	1.39
NodeDrop	1.85	1.45	1.66	1.53	1.31

Figure 2: Semi-supervised learning accuracy gain (%) when contrasting different augmentation pairs, compared to training from scratch, under four datasets: NCI1, PROTEINS, COLLAB, and RDT-B. Pairing "Identical" stands for a no-augmentation baseline for contrastive learning, where the positive pair diminishes and the negative pair consists of two non-augmented graphs. Warmer colors indicate better performance gains. The baseline training-from-scratch accuracies are 60.72%, 70.40%, 57.46%, 86.63% for the four datasets respectively.

Paper 4

PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding

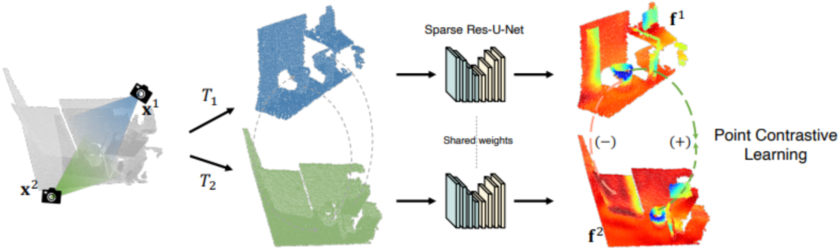
Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J. Guibas, Or Litany

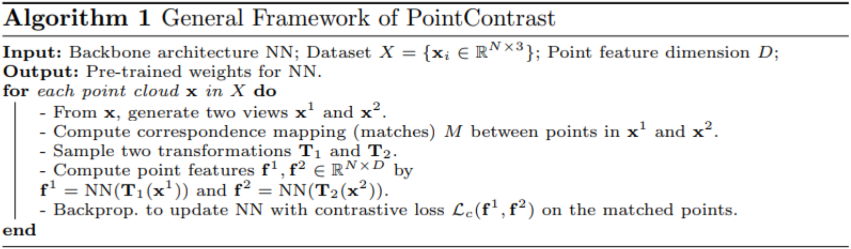
https://arxiv.org/abs/2007.10985

https://github.com/facebookresearch/PointContrast

Overview

Fig. 2: **PointContrast: Pretext task for 3D pre-training.**





Algorithm 1 General Framework of PointContrast

```
Input: Backbone architecture NN; Dataset $X = \{\mathbf{x}_i \in \mathbb{R}^{N\times 3}\}$; Point feature dimension $D$;
Output: Pre-trained weights for NN.
for each point cloud $\mathbf{x}$ in $X$ do
    - From $\mathbf{x}$, generate two views $\mathbf{x}^1$ and $\mathbf{x}^2$.
    - Compute correspondence mapping (matches) $M$ between points in $\mathbf{x}^1$ and $\mathbf{x}^2$.
    - Sample two transformations $\mathbf{T}_1$ and $\mathbf{T}_2$.
    - Compute point features $\mathbf{f}^1, \mathbf{f}^2 \in \mathbb{R}^{N\times D}$ by
    $\mathbf{f}^1 = \mathrm{NN}(\mathbf{T}_1(\mathbf{x}^1))$ and $\mathbf{f}^2 = \mathrm{NN}(\mathbf{T}_2(\mathbf{x}^2))$.
    - Backprop. to update NN with contrastive loss $\mathcal{L}_c(\mathbf{f}^1, \mathbf{f}^2)$ on the matched points.
end
```

PointInfoNCE Loss

$$\mathcal{L}_c = - \sum_{(i,j)\in\mathcal{P}} \log \frac{\exp(\boldsymbol{f} \cdot \boldsymbol{f}_j/\tau)}{\sum_{(\cdot,k)\in\mathcal{P}} \exp(\boldsymbol{f}_i \cdot \boldsymbol{f}_k/\tau)}$$

where $\mathcal{P}$ is the set of all the positive matches from two views:

- For a matched pair $(i,j) \in \mathcal{P}$, point feature $\boldsymbol{f}_i^1$ will serve as the query, and $\boldsymbol{f}_j^2$ will serve as the positive key.
- Point feature $\boldsymbol{f}_k^2$ where $\exists(\cdot,k) \in \mathcal{P}$ and $k \neq j$ as the set of negative keys.
- The number of points is 100K, so sample 4,096 pairs of matching only.