Self-Supervised Relational Reasoning for Representation Learning

M. Patacchiola & A. Storkey, Schoolf of informatics, University of Edinburgh

What is self-supervised learning?

Goal: train a backbone neural network from unlabeled data and transfer acquired knowledge to other tasks.

Self-supervised learning:

1. Define a proxy task over the unlabeled dataset (e.g. classification on surrogate classes).
2. Train a network (backbone) to solve the proxy task and learn useful representations on the way.
3. Transfer the knowledge to downstream-tasks (e.g. classification, segmentation, image retrieval).

Previous work: predict image rotations (RotationNet, Gidaris et al. 2018), contrastive losses (SimCLR, Chen et al. 2020), maximize mutual information (Deep InfoMax, Hjelm et al. 2019), supervised learning using pseudolabels (DeepCluster, Caron et al. 2018).

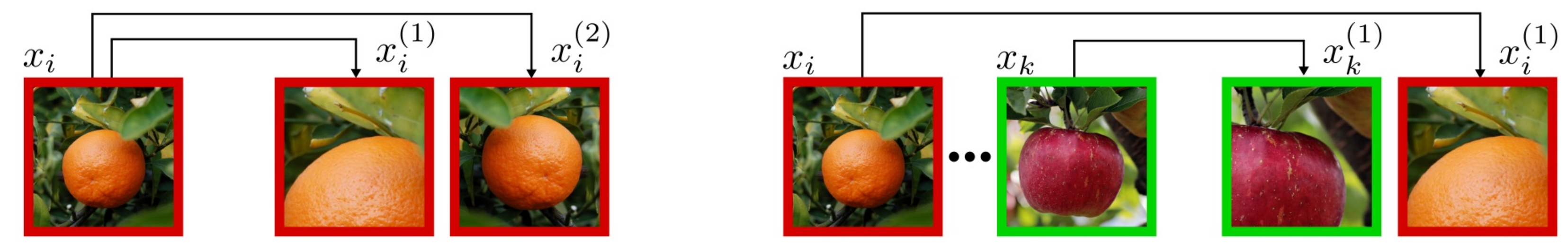
Description of the method

Algorithm 1 Self-supervised relational learning: training function and shuffling without collisions.

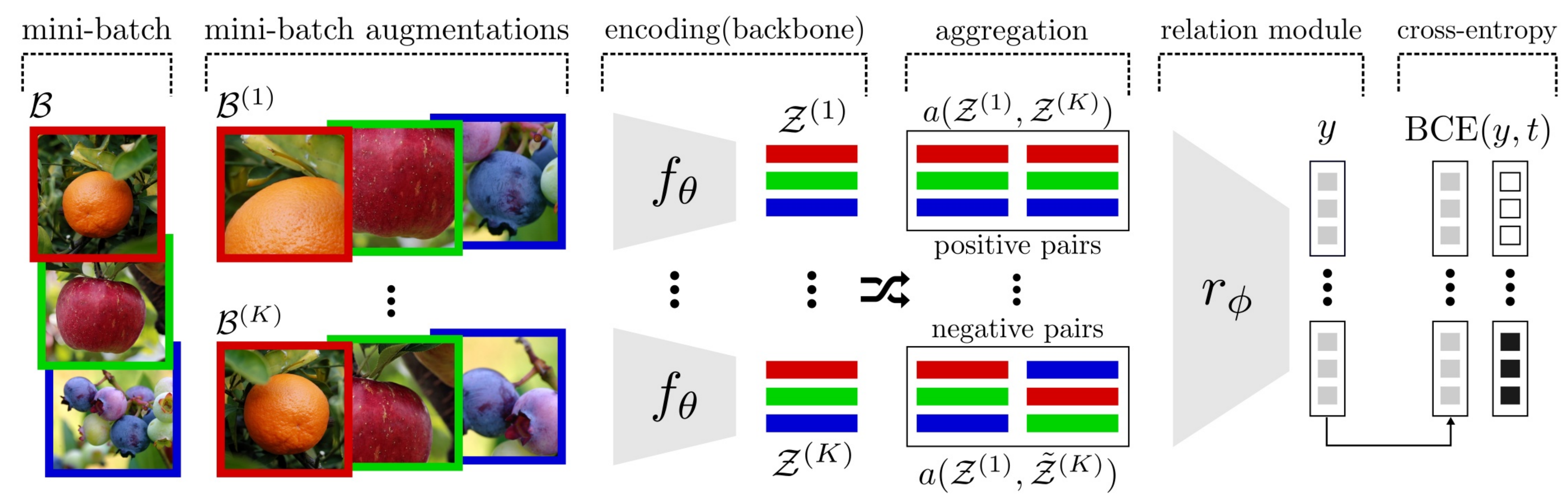
Require: $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^{N}$ unlabeled training set; $\mathcal{A}(\cdot)$ augmentation distribution; $\boldsymbol{\theta}$ parameters of f_θ (neural network backbone); $\boldsymbol{\phi}$ parameters of r_ϕ (relation module); aggregation function $a(\cdot,\cdot)$; α and β learning rate hyperparameters; K number of augmentations; M mini-batch size;

```
function TRAIN(D, α, β, M, K, θ, φ)
    while not done do
        B = {x_m}_{m=1}^{M} ~ D                          ▷ Sampling a mini-batch
        for k = 1 to K do
            B^(k) ~ A(B)                                 ▷ Sampling K mini-batch augmentations
            Z^(k) = f_θ(B^(k))                           ▷ Forward pass in the backbone
        end for
        P = {}                                           ▷ Empty set to store aggregated pairs and targets
        for i = 1 to K − 1 do
            for j = i + 1 to K do
                P ← (a(Z^(i), Z^(j)), t = 1)             ▷ Aggregating and appending positive pairs
                Z̃^(j) = SHUFFLE(Z^(j))                   ▷ Shuffling without collisions
                P ← (a(Z^(i), Z̃^(j)), t = 0)             ▷ Aggregating and appending negative pairs
            end for
        end for
        y = r_φ(P)                                       ▷ Forward pass in the relation module
        L = BCE(y, t)                                    ▷ Estimating the Binary Cross-Entropy loss
        θ ← θ − α∇_θ L                                   ▷ Updating backbone
        φ ← φ − β∇_φ L                                   ▷ Updating relation module
    end while
    return θ, φ                                          ▷ Returning the learned weights
end function

function SHUFFLE(Z)
    Z̃ = Z                                                ▷ Copying the input set
    for m = 1 to M do
        m̃ ~ {1, ..., M} \ {m}                            ▷ Sampling an index m̃ ≠ m
        Z̃_m ← Z_m̃                                        ▷ Assigning a random representation with index m̃
    end for
    return Z̃                                             ▷ Returning the shuffled set
end function
```



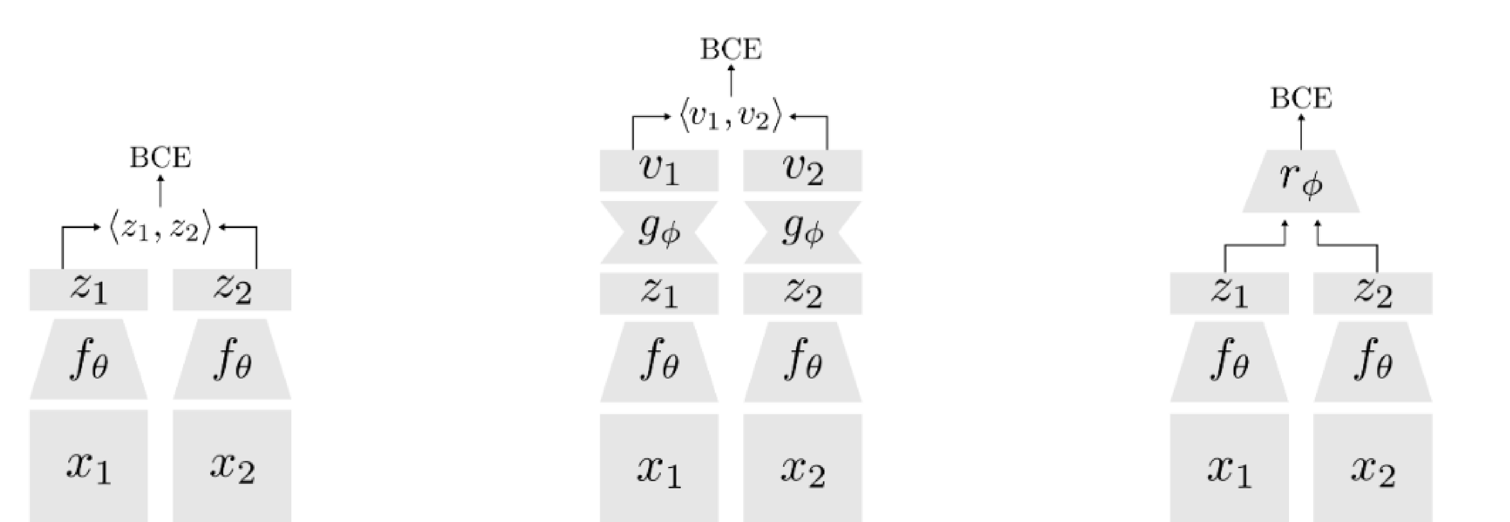
Example of augmentation and aggregation of positive (left) and negative (right) image pairs. Note that pairs are aggregated in representation space in the full pipeline.



1) The mini-batch is augmented K times; **2)** All instances are passed through the backbone with a forward pass; **3)** Representations are aggregated generating positive and negative pairs; **4)** Pairs are passed through the relation head; **5)** The prediction of the relation head is compared with the target pseudo-label (1=positives, 0=negatives) and assigned to a Binary Cross-Entropy loss (BCE) for the optimization.

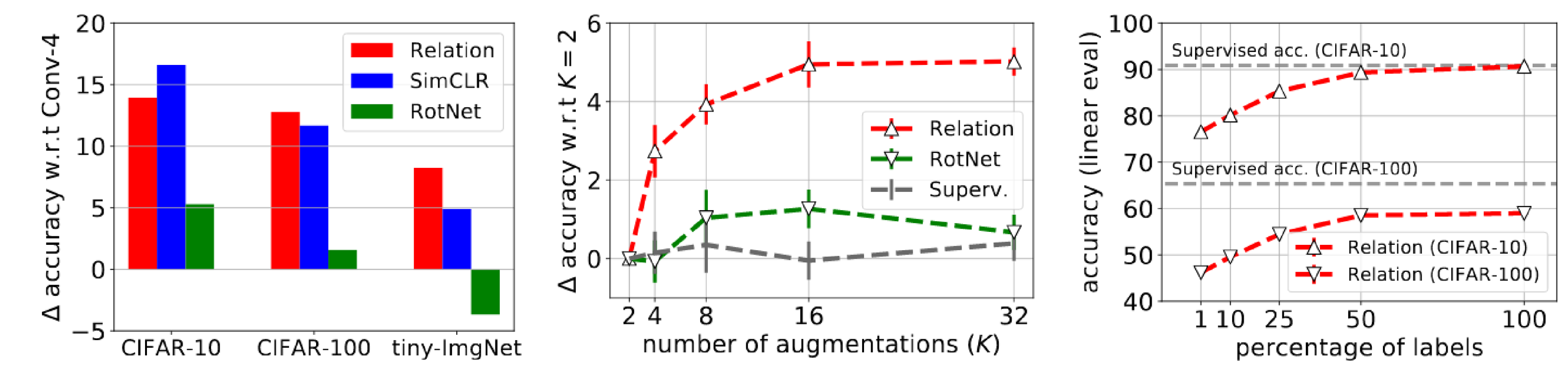
Experiments: overview

Method	Linear Evaluation		Domain Transfer		Grain	Finetune
	CIFAR-100	tiny-ImgNet	10→100	100→10	CIFAR-100-20	STL-10
Supervised (upper bound)	65.32±0.22	50.09±0.32	33.98±0.71	71.01±0.44	76.35±0.57	69.82±3.36
Random Weights (lower bound)	7.65±0.44	3.24±0.43	7.65±0.44	27.47±0.83	16.56±0.48	n/a
DeepCluster	20.44±0.80	11.64±0.21	18.37±0.41	43.39±1.84	29.49±1.36	73.37±0.55
RotationNet	29.02±0.18	14.73±0.48	27.02±0.20	52.22±0.70	40.45±0.39	83.29±0.44
Deep InfoMax	24.07±0.05	17.51±0.15	23.73±0.04	45.05±0.24	33.92±0.34	76.03±0.37
SimCLR	42.13±0.35	25.79±0.35	36.20±0.16	65.59±0.76	51.88±0.48	89.31±0.14
Relational Reasoning (ours)	**46.17**±0.17	**30.54**±0.42	**41.50**±0.35	**67.81**±0.42	**52.44**±0.47	**89.67**±0.33



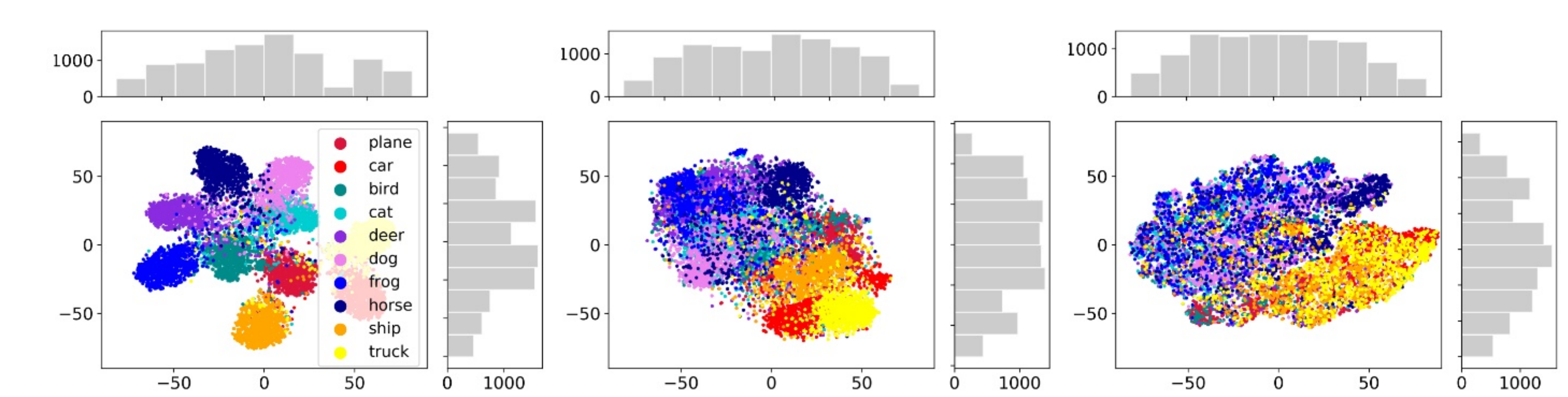
(a) dot product (b) Encoder + dot product (c) Relation module (ours)

Head type	Linear Evaluation		Domain Transfer		Grain
	CIFAR-10	CIFAR-100	10→100	100→10	CIFAR-100-20
(a) dot product	72.74±0.22	28.77±0.44	18.19±0.10	51.9±0.50	45.05±1.07
(b) Encoder + dot product	59.44±0.59	29.91±1.28	28.29±0.90	53.65±0.85	36.94±1.30
(c) *Relation module* (ours)	**74.99**±0.07	**46.17**±0.17	**41.50**±0.35	**67.81**±0.42	**52.44**±0.47



Left: Conv4 VS ResNet32 performance; **Center:** accuracy VS tot augmentations; **Right:** accuracy VS availabe labels

Image retrieval task, query (red frame) and top-10 closests images. **Left:** our method, **Right:** RotationNet



(a) Supervised (b) Relational Reasoning (ours) (c) RotationNet

Projection on the Cartesian plane of representations using t-SNE for pre-trained methods on CIFAR-10 test points

References

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. InEuropean Conference on Computer Vision.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations.arXiv preprint arXiv:2002.05709.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting imagerotations. In International Conference on Learning Representations.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y.(2019). Learning deep representations by mutual information estimation and maximization. In InternationalConference on Learning Representations.

Acknowledgment

This work was supported by a Huawei DDMPLab Innovation Research Grant.