Information based Deep Clustering: An experimental study

Jizong Peng  Christian Desrosiers  Marco Pedersoli
ETS Montreal
jizong.peng.1@etsmtl.net

Abstract

Recently, two methods have shown outstanding performance for clustering images and jointly learning the feature representation. The first, called Information Maximizing Self-Augmented Training (IMSAT), maximizes the mutual information between input and clusters while using a regularization term based on virtual adversarial examples. The second, named Invariant Information Clustering (IIC), maximizes the mutual information between the clustering of a sample and its geometrically transformed version. These methods use mutual information in distinct ways and leverage different kinds of transformations. This work proposes a comprehensive analysis of transformation and losses for deep clustering, where we compare numerous combinations of these two components and evaluate how they interact with one another. Results suggest that mutual information between a sample and its transformed representation leads to state-of-the-art performance for deep clustering, especially when used jointly with geometrical and adversarial transformations.

1. Introduction

In deep learning, supervised methods have shown excellent performance, sometimes even surpassing the human level [22, 11]. However, these methods require large datasets with fully annotated data, which cannot be affordable in many cases [42]. Instead, unsupervised methods can learn from data without annotations, which is very appealing given the large amount of data that can easily be collected from social media and other sources [38]. In this paper, we are interested in the unsupervised learning problem of deep clustering, which consists in learning to group data into clusters, while at the same time finding the representation that best explains the data. Jointly learning to group data (clustering) and represent data (representation learning) is an ill-posed problem which can lead to poor or degenerate solutions [46, 45, 5]. A principled way to avoid most of these problems is mutual information [34]. Mutual information is a powerful approach for clustering because it does not make assumptions about the data distribution and reduces the problems of mode collapse, where most of the data is grouped in a single large cluster [5].

In recent publications, two papers obtained outstanding results for deep clustering by using mutual information in different ways. The first one, IMSAT [44] maximizes the mutual information between input data and the cluster assignment, and regularizes it with virtual adversarial examples [29] by imposing that the original sample and the adversarial sample should have similar cluster assignment probability distribution (by minimizing their KL divergence). The second one, IIC [14], maximizes the mutual information of the cluster assignment between a sample and its transformed version. Both algorithms are based on a mutual information loss (applied either to input output for IMSAT or output and transformed output for IIC) which relies on pairwise associations of an image and its transformed version (either by a geometrical transformation or an adversarial one). In this paper, we aim to analyze and better understand these algorithms by decomposing them in two basic building blocks: the information-based loss and the used transformation. As
illustrated in Fig. 1, these two blocks can be combined in multiple and non-trivial ways. By following this strategy, we build a family of clustering approaches in which IMSAT and IIC are special cases. We evaluate the different instances of this family of methods on three datasets with different difficulty, and perform an in-depth analysis of obtained results.

Our empirical analysis shows that: i) maximizing the mutual information between an image clustering distribution and its transformed version seems to be more robust than other approaches when dealing with challenging datasets; ii) the combined use of different transformations seems beneficial (e.g., geometrical and adversarial) as they capture complementary characteristics of images; iii) the optimal way of including a transformation in the loss differs across transformations. For instance, we found that combining geometrical transformations in the mutual information loss with adversarial transformations used as regularization leads to improved accuracy.

Finally, we show in the experimental results that deep clustering is a convenient way to perform unsupervised learning since the final task is very related to classification. Moreover, when the representation is used for supervised tasks, it largely outperforms other approaches.

In the remainder of this paper, we first introduce related work in section 2, with special attention to the two mentioned methods. Then, in section 3, we propose the experimental protocol by defining the different components of our experiments. Finally, we report results in section 4 and draw conclusions in section 5.

2. Related work

Mutual information Mutual information $I(X;Y)$ is a information-theoretic criterion to measure the dependency between two random variables $X, Y$ [4]. It is defined as the KL divergence between the joint distribution $p(X, Y)$ of two variables and the product of their marginal: $I(X;Y) = D_{KL}(p(X,Y), p(X)p(Y))$.

The criterion of maximizing mutual information for clustering is first introduced in [4], as the firm but fair criterion. In this case the mutual information between input data (i.e., image or representation) and output categorical distribution is maximized, believing that the class distribution can be deduced given the input. This principle is extended in [20], in which mutual information is maximized with additionally an explicit regularization, such as $L_2$ loss. This helps to avoid too complex decision boundaries.

In [41, 37, 40] mutual information is used to align inputs with different modalities [41, 37, 40], because with different modalities normal distances are meaningless. Finally, mutual information is also used as regularization in a semi-supervised setting [28]. Recently, DeepInfoMax [13] simultaneously estimates and maximizes mutual information between images and learns high-level representations. However, estimating the mutual information of images is hard and requires complex techniques [3].

Finally, two recent techniques for deep clustering based on mutual information are IMSAT [44] and IIC [15]. As these are the starting point of this study, they will be analyzed in more detail in section 3.

Self-supervised approaches Self-supervised learning has recently emerged as a way to learn a representative knowledge based on non-annotated data. The main principle is to transfer the unsupervised task to a supervised one by defining some pseudo labels that are automatically generated by a pretext task without involving any human annotations [16]. The network trained with the pretext task is then used as the initialization of some downstream tasks, such as image classification, semantic segmentation and object detection. It has been shown that a good pretext task can be beneficial and help to improve the performance of the downstream task [35, 8, 32, 6, 1]. Without the access to label information, the pretext task is usually defined based on the data structure and is somewhat related but different from the downstream tasks. Various pretexts have been proposed and investigated. Similarity-based methods [32, 6] design a pretext task that let the network learn the semantic similarities between image patches. Likewise, [31, 1, 43] trained a network to recognize the spatial relationship between image patches. See [16] for a complete survey on such methods.

If the task we want to learn with non-annotated data is classification, in this work we argue that the best pretext task is clustering. With clustering, the pretext task is very close to the downstream task that is classification. In fact, clustering aims to group the data in a meaningful way and therefore split the data into categories. If these categories are not only visually similar but also semantically, classification and clustering become the same task. In other words, with clustering, there is not need for a downstream task. By assigning the most likely category to each cluster, our clustering method becomes a classifier. In the experimental evaluation, we will compare the performance of our information based clustering with state-of-the-art self-supervised learning approaches. For doing that, we use two simple assumptions: i) the sample distribution per class is known (normally uniform) and ii) the exact number of classes, which corresponds to the number of clusters is also known.

Clustering approaches Clustering has been long time studied before the deep learning era. K-means [18] and GMM algorithms [2] were popular choices given representative features. Recently, much progress has been made by jointly training a network that perform feature extraction together with a clustering loss [45, 25, 10, 6]. Deep Embedded Clustering (DEC) [45] is a representative method that uses an auto-encoder as the network architecture and a cluster-
assignment hardening loss for regularization. Li et al. [25] proposed a similar network architecture but with a boosted discrimination module to gradually enforce cluster purity. DEPICT [10] improved the clustering algorithm’s scalability by explicitly leveraging class distributions as prior information. DeepCluster [6] is an end-to-end algorithm that jointly trains a Convnet with K-means and groups high-level features to $N$ pseudo labels. Those pseudo labels are in turn used to retrain the network after each iteration. In this work, we focus on two state of the art information-based clustering approaches [44, 15] and analyze their different components and how they can be combined in a meaningful way.

3. Information-based clustering

3.1. Overview

As shown in Fig. 1, we consider information-based clustering as a family of methods having two main components: the mutual information loss and the used transformations. The maximization of the mutual information aims to produce meaningful groups of data, i.e. clusters with a similar representation and with a even number of samples. On the other hand, transformations are used to make the learned representation locally smooth and ease the optimization in a similar way as in data augmentation. For these two components, we consider and evaluate different possible choices and their combination.

3.2. Mutual Information losses

$I(Y, \tilde{Y})$: This formulation introduced in IIC [15] maximizes the mutual information between the clustering assignment variable $Y$ and the clustering assignment of a transformed sample $\tilde{Y}$. Mutual information is defined as the KL divergence between the joint probability of two variables and the product of their marginals [4]:

$$I(X_1, X_2) = D_{KL}(p(X_1, X_2), p(X_1)p(X_2)).$$  \hspace{1cm} (1)

It represents a measure of the information between the two variables. If the variables are independent, the mutual information is zero because the joint will be equal to the product of the marginals. Thus, we need to estimate the joint probability of the clustering assignment and its transformed version $p(Y, \tilde{Y})$, as well as the marginals $p(Y)$ and $p(\tilde{Y})$. The marginals are defined as

$$p(Y) = \mathbb{E}_{X}[p(Y|X)], \hspace{1cm} p(\tilde{Y}) = \mathbb{E}_{X}[p(Y|T(X))].$$  \hspace{1cm} (2)

and can be empirically estimated by averaging output mini-batches. For the joint, we compute the dot product between $Y|X$ and $Y|T(X)$ for each sample and marginalize over $X$:

$$p(Y, \tilde{Y}) = \mathbb{E}_{X}[p(Y|X) \cdot p(Y|T(X))^\top].$$  \hspace{1cm} (3)

For each sample, the joint probability of $c_1 \in Y|X$ and $c_2 \in Y|T(X)$ is $c_1 c_2$. For a single sample, by construction the joint and the marginals will be equivalent. However, when marginalizing the joint over samples $X$, the final $p(Y, \tilde{Y})$ will be different than $p(Y)p(\tilde{Y})$.

This formulation maximizes the predictability of a variable given the other. It is different than enforcing KL divergence between two distributions because: i) it does not enforce the two distributions to be the same, but only to contain the same information. For instance, one distribution can be transformed by an invertible operation without altering the mutual information. ii) it penalizes distributions that do not have uniform marginal, i.e. all the cluster should contain an even number of samples.

$I(X, Y)$: We consider the MI formulation used as loss in IMSAT [44]. It connects the input image distribution $X$ with the output cluster assignment of the used neural network $Y$. As the input of the neural network is a continuous vector, estimating its probability distribution is hard and we cannot use directly equ. (1). Instead, in IMSAT the mutual information between input and output is computed as:

$$I(X, Y) = H(Y) - H(Y|X)$$

$$= \mathbb{E}_{Y} [\log \mathbb{E}_{X}[p(Y|X)]] - \mathbb{E}_{X,Y} [\log p(Y|X)].$$  \hspace{1cm} (4)

In this formulation, the mutual information is easy to compute because it is the difference between the entropy of the output $H(Y)$ and the conditional entropy $H(Y|X)$. Both quantities can be approximated in a mini-batch stochastic gradient descent setting. $H(Y)$ is approximated as the entropy of the average probability distribution $p(Y|X)$ over the given samples, while $H(Y|X)$ is approximated as average of the conditional entropy of each sample. As entropy is a non-linear operation, the two quantities are different. $H(Y)$ is maximized when the probability of each cluster is the same, i.e. the output has the same probability distribution for each cluster. On the other hand, $H(Y|X)$ is minimized when, for each sample $X$, $p(Y|X)$ has most of the probability distribution assigned to a single cluster, i.e. the model is certain about a given choice. The two combined means that the clustering has always to choose just one cluster and globally, each cluster should contain the same number of samples. In case the class distribution is not uniform, another distribution $C$ can be enforced by $D_{KL}(p(X), p(C))$. In our experiments, we limit ourselves to uniform class distribution. Notice that, in IMSAT, the authors add an hyper-parameter MI formulation $\lambda H(Y) - H(Y|X)$. This parameter is the minimum value that ensures that the data is evenly distributed on all clusters. We follow the same approach as in the original paper.

Regularization: In the context of this work, we call regularization an additional loss that penalizes when the output of the model for the original image and the transformed
image are different. While for $\mathcal{I}(X, Y)$ this regularization is fundamental for good results, for $\mathcal{I}(Y, \tilde{Y})$ it is an optional step to strongly enforce a transformation. In this work, as we have access to the clustering probability distribution $p(Y|X)$ of a sample $X$, we use the KL divergence as penalty term $D_{KL}(p(Y|X), p(Y|\tilde{T}(X)))$. Although in unsupervised settings there is not real difference between loss and regularization because in both cases the training is performed without annotations, here we consider this KL divergence as a regularization because it cannot be used alone for clustering. It will need at least another term such as $H(Y)$ to enforce an even distribution of samples in the clusters.

3.3. Transformations

Transformations seem to be a key component of MI-based clustering. To be useful, any transformation needs to change the appearance of the image (in terms of pixels) while maintaining its semantic content, i.e. the class of the image. We can think of transformations as a pseudo ground-truth that helps to train the model. In this work, we consider three kind of transformations: Geometrical, Adversarial and Mixup.

Geometrical: Geometrical transformations are the image transformations that are normally used for data augmentation. As in [14], we use random crop, resize at multiple scales, horizontal flip, and color jitter with the same range of parameters that is normally used in data augmentation. Note that some transformations can actually change the category of a class. For instance, on MNIST [24] a dataset composed of numbers, a crop of a 6 can zoom in the lower circle and look very similar to a 0. We will talk more about this problem in the experimental results. In IIC, geometrical transformations are used directly in the mutual information loss, however in our experiments we also tested their usefulness for regularization.

Adversarial: Adversarial samples [47] are samples that are slightly modified by an adversarial noise which is usually unnoticeable by the human eye, but can induce a neural network to misclassify an example. Recently, methods based on adversarial examples have attracted a lot of attention because they can easily fool machine learning algorithms and thus represent a threat to any system using machine learning [39, 7]. It has been shown [26] that adding those samples during training can help to improve the robustness of the classifier. In this study, we use Virtual Adversarial Training (VAT) [29], an extension of adversarial attack that can also be used for non-labelled samples and has shown promising results for fully-supervised, semi-supervised [29] and unsupervised learning [44]. The adversarial noise $r$ can be found as the value within a certain neighbourhood $\epsilon$ that maximizes the distance $D$ between the probability distribution of the original sample $x$ and the transformed sample $x + r$:

$$r_{\text{adv}} = \arg\max_{\|r\|_2 \leq \epsilon} D[p(y|x, \theta), p(y|x + r)], \quad (5)$$

where $D[p_1(\cdot), p_2(\cdot)]$ is a divergence function, usually defined as KL-divergence. In practice, eq. (5) can be optimized in order to find $r$ with a few iterations of the power method [17]. Note that we could also experiment with adversarial geometrical transformations as in [36], but we leave this direction as future work.

Mixup: This is a simple data augmentation technique that has proven successful for supervised learning [48]. It consists on creating a new sample $x_{\text{new}}$ and label $y_{\text{new}}$ by linearly combining two training samples $x_1$ and $x_2$ (e.g. images) and labels $y_1$ and $y_2$ (e.g. class probabilities):

$$x_{\text{new}} = \alpha x_1 + (1 - \alpha) x_2$$

$$y_{\text{new}} = \alpha y_1 + (1 - \alpha) y_2. \quad (6)$$

$\alpha$ is the mixing coefficient and is normally sampled from a $\beta$ distribution. Although very simple and effective, Mixup has received multiple criticisms because it is clear that the generated images do not represent real samples. However, the $\beta$ distribution has most of its mass near 0 and 1, which means that in most of the cases the mixed samples look very similar to one of the samples, but with a structured noise coming from the other image. This transformation differs from the previous one because it requires two input samples to generate a new one. Thus, to use it in our family of algorithms, we had to adapt it. For $\mathcal{I}(Y, \tilde{Y})$, as before we consider $Y = \mathbb{E}_X[p(Y|X)]$ the expected output of real samples $X$, while $\tilde{Y} = \mathbb{E}_{X, X_2, \alpha}[\alpha p(Y|X)(1 - \alpha) p(Y|X_2)]$ is now the output associated to mixup samples generated using the same real samples $X$ in combination with other samples $X_2$ that is randomly selected. Mixup can also be used as direct regularization (see next section). In this case, the first output distribution is associated to real samples while the second is associated to mixup samples built as above.

4. Experiments

Our main experiment evaluates on the three datasets (presented below) the two identified components for information based clustering: information based losses and image transformations. To do so, we summarize all our experiments in two main tables in which only one component (either the losses or the transformations) is considered and the other is considered as an hyper-parameter that we want to optimize. For completeness, we have reported all results with all combinations of the different components in the supplementary material. In the following tables, for each result, we added a code that indicates how to find that result
in the complete set of experiments. In the second part of our experiments, we present other interesting findings that were not visible from the two tables. We first compare the performance of different combinations of mutual information losses and transformations, showing that only some of them make sense and can produce meaningful results. Then, we show that the proposed clustering can be used to initialize the parameters of a network. We compare our best model with DeepInfoMax in the task of linear supervised classification using the representation learned by the respective methods. Finally, we visualize the clusters of our models on three datasets.

4.1. Datasets

We evaluate the different methods on 3 datasets:

- MNIST dataset [24] of hand-written digit classification consists of 60,000 training images and 10,000 validation images. 10 classes are evenly distributed in both train and test sets. Following common practice, we mix the training and test set to form a large training set. During training, we do not show any ground truth information, while for testing, we use image annotations to find a mapping between true class label and cluster assignment, thus assessing the clustering performance by the classification accuracy.

- CIFAR10 [21] is a popular dataset consisting of 60,000 32×32 color images in 10 classes, with 6,000 images per class. Similar to MNIST dataset, we mix the 50,000 training images with 10,000 test images to build a larger dataset for clustering.

- SVHN [30] is a real-world image dataset for digit recognition, consisting of 73,257 digits for training, 26,032 digits for testing. Images come from natural scene images. We adopt the previously described strategy to use this dataset too.

4.2. Evaluation Metric

Our method groups samples into clusters. If the grouping is meaningful, it should be related to the dataset classes. Thus, in most of our experiments, we use classification accuracy as measure of the clustering quality. This makes sense because the final aim of this approach is exactly to produce a classifier without using training labels. This accuracy is based on the best possible one-to-one mapping (using the Hungarian method [23]) between clustering assignment and ground truth label (assuming they share the same number of classes). We run the experiments 3 times with different initialization and report mean and standard deviation values.

4.3. Implementation Details

In order to provide a fair comparison, we use the same network for a given dataset across methods. For both MNIST and SVHN dataset, we borrow the setting of IMSAT [12] based network. For CIFAR-10, we use a ResNet-34 [14] based network. It is worth mentioning that, in original IMSAT paper [44], the used network was just a 2 fully connected layers with pre-trained features on CIFAR-10 or GIST features [33] on SVHN. Instead, in this work, we want to compare all results on the same convolutional architecture and without pre-trained models or any hand-crafted features.

To boost performance, as in [14] we use two additional procedures. The first one, over-clustering, consist in using more clusters than the number of classes in the training data. This can help to find sub-classes and therefore reduce the intra-class variability on each cluster. The second consists in splitting the last layer of the network in multiple final layers (there called heads) and therefore multiple clusters. This can increase diversity and acts as a simplified form of ensembling. Combining these two techniques can highly boost the final performance of the clustering approach. However, they also increase the computational cost of the model. Thus, for the evaluation of all configurations in a same setting, we use a basic model without additional over-clustering or multiple final layers. However, for our best configuration, we retrained it with 5 final layers with 10 clusters (as the number of classes) and other 5 final layers with 50 clusters for MNIST and SVHN or 70 clusters for CIFAR10.

4.4. Mutual Information losses

In table 1, we consider the two ways of using mutual information for clustering as explained in section 3.2. As we want to factorize our transformations on this analysis, reported results are obtained with the best performing transformations. For more details, see table 1 in the supplementary material. On MNIST, most of the methods perform quite well. Only \(I(X, Y)\) does not obtain good results and, in fact, it obtains the lowest accuracy on all datasets (row 1). In contrast, if we induce similarity between the clustering probabilities of a sample and its transformed version with KL divergence (row 2, this configuration corresponds to IMSAT when the transformation is Adversarial) the obtained results are much improved. This is because we can consider inducing similarity on the cluster probabilities as a form of self-supervision. Thus, not leveraging this source of information drastically reduces results quality. When using \(I(Y, \tilde{Y})\) instead, results are already good without KL divergence (row 3, this configuration corresponds to IIC when the transformation is geometrical). This is because, in this formulation, the mutual information directly considers the
transformations. However, adding explicitly a KL term further improves results (row 4). This shows that enforcing similarity between transformed samples with mutual information and with KL divergence are complementary strategies, and therefore using them jointly helps to further improve performance. Finally, combining the two different information losses together with the KL divergence (row 5) gives the best results on two of the three datasets.

Table 2: Transformations. The reported results show the performance of each transformation associate with the best information based loss. The complete table with all experiments can be found in the supplementary material. The letters beside every accuracy value refer to the corresponding row in the supplementary material table.

4.5. Transformations

In table 2, we compare the three different kinds of transformations that are described in section 3.3. From the table, we can see that the best single transformation seems to be geometric. Compared to Adversarial, the gap is small on MNIST, but it becomes larger on the other datasets. For Mixup, performances are much lower than geometric, but slightly better than adversarial on CIFAR10. We also tested possible combinations of transformations. In this case, the most promising seems to be Geometric + Adversarial, although Geometric + Mixup also performs well. Finally, using the three combinations jointly does not seem to help. Thus, our best configuration is Geometric + Adversarial transformations.

4.6. Matching losses and transformations

So far, we have analyzed different ways of clustering data from the point of view of information loss and transformations. However, this analysis not show how information loss and transformations are combined in our experiments. Different transformations can be matched with the information loss in distinct, and only some of these configurations make sense and provide good results. In table 3, we consider this problem for Geometrical and Adversarial transformations.

Geometrical For Geometrical transformations, \( I(Y, Y_{\text{Geo}}) \) works very well (row 3). However, when we try to use the same transformations for \( I(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|T(X))) \) (row 2), results are poor, sometimes even inferior to the mutual information loss without any regularization \( I(X, Y) \) (row 1). We believe this is due to the different ways this transformations are used in the two cases. In the case of \( I(Y, Y_{\text{Geo}}) \), we impose that the network output for an example and its transformed version should have maximum mutual information. In contrast, in \( I(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|\text{Geo}(X))) \), we explicitly want the two output distribution to be similar. Thus, the latter is a much stronger constraint than the former. However, we analyzed the used geometrical transformation and we discover that this transformation sometimes does not respect the basic rule of maintaining the same semantic meaning of the image, i.e. the same class. For instance in MNIST, a 6 with a crop can easily become a 0. We hypothesize that these ambiguities can confuse the training, when enforcing geometrical consistency with KL divergence. Instead, if we only impose to maximize the information between the original image and the transformation, clustering works well. To verify this hypothesis, we ran a second experiment with weaker geometrical transformations (just an image crop in which the new image is a few pixel smaller than the original), such that we ensure that an image and its transformed version maintain the same class on all datasets. This new transformation, even if very weak, helps \( I(X, Y) \) with KL divergence to obtain better
results (row 4), while using such weak transformation in \( \mathcal{I}(Y, Y_{\text{Geo}}) \) does not help much (row 5).

**Adversarial** Adversarial transformations are different from the other kinds of transformations because they depend on the actual loss that the network is optimizing. For instance, in \( \mathcal{I}(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|\text{Geo}(X))) \), the adversarial transformation works well and helps to improve the performance of the method on all datasets (row 6). However, if we use the same adversarial transformations with \( \mathcal{I}(Y, Y_{\text{VAT}}) \), results are very poor. We assume that this is due to a mismatch of losses. In fact, adversarial samples are generated to be adversarial to the KL divergence between the original sample and the noise sample (see equ. (5)). However, with \( \mathcal{I}(Y, Y_{\text{VAT}}) \), we optimize the mutual information between input and adversarial transformation. So, the adversarial transformation is actually adversarial for KL divergence, but not for mutual information and therefore it will not really help the clustering. Instead, if we use an adversarial sample generated against the mutual information loss, results improve. We still generate this new kind of adversarial samples with equ. (5), but using mutual information \( \mathcal{I}(Y, \tilde{Y}) \) between the original sample and the transformed sample as distance. We call this new transformation IVAT, as information based VAT. Results of this experiment are shown in the last row of the table. The obtained accuracy is much better on MNIST, while slightly lower than the normal VAT on CIFAT10 and SVHN.

| Training | DA | Accuracy |
|----------|----|----------|
| Scratch  | no | 72.9%    |
| Pretrained | no | 80.6%    |
| Scratch  | yes| 81.6%    |
| Pretrained | yes| 82.8%    |

Table 4: **Clustering for pre-training.** We evaluate how our best model can be used as initialization for supervised learning.

| Transform | Loss | MNIST | CIFAR10 | SVHN |
|-----------|------|-------|---------|------|
| None      | \( \mathcal{I}(X, Y) \) | 42.6 ± 3.9 | 16.0 ± 1.0 | 15.5 ± 1.5 |
| Geo       | \( \mathcal{I}(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|\text{Geo}(X))) \) | 43.8 ± 47.8 | 14.5 ± 0.7 | 17.4 ± 1.5 |
|           | \( \mathcal{I}(Y, Y_{\text{Geo}}) \) | 97.8 ± 0.0 | 31.9 ± 1.1 | 28.0 ± 4.0 |
| Weak Geo  | \( \mathcal{I}(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|\text{Geo}(X))) \) | 56.7 ± 0.5 | 22.2 ± 0.5 | 32.3 ± 18.4 |
|           | \( \mathcal{I}(Y, Y_{\text{Geo}}) \) | 25.1 ± 2.5 | 18.5 ± 0.6 | 19.3 ± 5.0 |
| VAT       | \( \mathcal{I}(X, Y) + D_{\text{KL}}(p(Y|X), p(Y|\text{VAT}(X))) \) | 97.7 ± 0.3 | 18.4 ± 0.4 | 17.1 ± 2.0 |
|           | \( \mathcal{I}(Y, Y_{\text{VAT}}) \) | 11.3 ± 0.0 | 17.4 ± 0.4 | 17.0 ± 1.9 |
|           | \( \mathcal{I}(Y, Y_{\text{IVAT}}) \) | 65.6 ± 7.4 | 15.8 ± 2.6 | 16.9 ± 1.9 |

Table 3: **Matching geometrical transformations and losses.** We show that each transformation should be associated to the correct loss in order to obtain good results.

Table 5: **Supervised Linear Classifier.** To compare our model with self-supervised approaches, we extract a representation from the fully connected layer (FC), the penultimate convolutional layer (Conv) and the output (Y) of our clustering network and use it for supervised training on CIFAR10.

### 4.7. Clustering for pre-training

In table 4, we consider the effect of using the network trained for clustering as pre-training for a fully-supervised training on the same dataset (CIFAR10). When we train without any data augmentation, the initialization based on clustering gives an important boost in performance, going from an accuracy of 72.9% for a network trained from scratch to 80.6% for a network whose weights are initialized with our best model. When the network is trained with data augmentation, the gap between the two initialization is reduced to 1.2%. This is probably due to the fact that most of the useful information brought with the clustering pretrained model is about image transformations. Thus, when adding data augmentation that information is included directly in the training.

### 4.8. Train a Linear Classifier

As we perform clustering, the final results are already groups of samples that are visually similar. If the clustering works well, these groups should represent classes. Thus, in our experiments, it is not necessary to perform any additional supervised learning for evaluation. In contrast, methods based on self-supervision will normally need a fi-
We visually compare for the three datasets the second and third rows of Table 3 which corresponds to our re-implementation of IIC and IMSAT but with geometrical transformations. Each row represents a class in the dataset. Images have been randomly selected.

Figure 2: **IIC and IMSAT with geometrical transformations** We visually compare for the three datasets the second and third rows of Table 3, which corresponds to our re-implementation of IIC and IMSAT but with geometrical transformations. Each row represents a class in the dataset. Images have been randomly selected.

4.9. Visualization of the clusters

In Fig. 2, we visually compare the clustering performance of the two different mutual information losses with geometrical transformations (as in rows 2 and 3 of Table 3). Each row represents a cluster found in an unsupervised way. If the samples in each row belong to the same class, the clustering has managed to find a meaningful grouping strategy and the associated classification accuracy will be high. From visual inspection and similarly to the accuracies in Table 3, we observe that the clustering based on $I(Y, \hat{Y})$ with geometrical transformations performs better than $I(X, Y)$ on the same transformations.

5. Conclusions

In this paper, we have presented an in-depth analysis of two very popular information-based clustering approaches. We first consider the two approaches as a combination of a MI-based loss and a class of transformations. This has lead us to a more general formulation of a information-based clustering in which we can combine different losses formulations and different transformations. From our empirical evaluation of these different configurations, we conclude that different transformations require different losses for optimal performance. Our best configuration is then a clustering that maximizes the mutual information between input and output as well as the mutual information between a samples and its geometrical transformation, and which enforces with KL divergence the similarity between a sample and its adversarial transformation. This configuration seems to be better than all previous work on clustering as well as other self-supervised approaches.
References

[1] U. Ahsan, R. Madhok, and I. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 179–189. IEEE, 2019.

[2] J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

[3] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[4] J. S. Bridle, A. J. R. Heading, and D. J. C. MacKay. Unsupervised classifiers, mutual information and ‘phantom targets’. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 1096–1101. Morgan-Kaufmann, 1992.

[5] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. *CoRR*, abs/1807.05520, 2018.

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[7] E. Chou, F. Tramèr, G. Pellegrino, and D. Boneh. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018.

[8] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[9] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[10] K. Ghaseli Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 1026–1034, Washington, DC, USA, 2015. IEEE Computer Society.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[14] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *CoRR*, abs/1807.06653, 2018.

[15] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018.

[16] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv preprint arXiv:1902.06162*, 2019.

[17] M. Journé, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11(Feb):517–553, 2010.

[18] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1(7):881–892, 2002.

[19] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[20] A. Krause, P. Perona, and R. G. Gomes. Discriminative clustering by regularized information maximization. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 775–783. Curran Associates, Inc., 2010.

[21] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

[23] H. W. Kuhn and B. Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, pages 83–97, 1955.

[24] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] F. Li, H. Qiao, and B. Zhang. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, 83:161–173, 2018.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[27] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[28] V. Manohar, D. Povey, and S. Khudanpur. Semi-supervised maximum mutual information training of deep neural network acoustic models. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[29] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, Aug 2019.
[30] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011. 5

[31] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In European Conference on Computer Vision, pages 69–84. Springer, 2016. 2

[32] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9359–9367, 2018. 2

[33] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision, 42(3):145–175, 2001. 5

[34] L. Paninski. Estimation of entropy and mutual information. Neural Comput., 15(6):1191–1253, June 2003. 1

[35] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 16–17, 2017. 2

[36] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2226–2234, 2018. 4

[37] J. P. Pluim, J. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. IEEE transactions on medical imaging, 22(8):986–1004, 2003. 2

[38] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. IEEE Trans. Pattern Anal. Mach. Intell., 33(4):754–766, Apr. 2011. 1

[39] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 2019. 4

[40] P. Thévenaz and M. Unser. Optimization of mutual information for multiresolution image registration. IEEE transactions on image processing, 9(ARTICLE):2083–2099, 2000. 2

[41] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. International journal of computer vision, 24(2):137–154, 1997. 2

[42] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. Int. J. Comput. Vision, 101(1):184–204, Jan. 2013. 1

[43] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1910–1919, 2019. 2

[44] S. T. E. M. M. S. Weihua Hu, Takeru Miyato. Learning discrete representations via information maximizing self-augmented training. ICML, 2017. 1, 2, 3, 4, 5

[45] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML’16, pages 478–487. JMLR.org, 2016. 1, 2

[46] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. CoRR, abs/1604.03628, 2016. 1

[47] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. IEEE transactions on neural networks and learning systems, 2019. 4

[48] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017. 4