Selective Pseudo-Label Clustering

Louis Mahon and Thomas Lukasiewicz

Department of Computer Science
University of Oxford, UK

Abstract. Deep neural networks (DNNs) offer a means of addressing the challenging task of clustering high-dimensional data. DNNs can extract useful features, and so produce a lower dimensional representation, which is more amenable to clustering techniques. As clustering is typically performed in a purely unsupervised setting, where no training labels are available, the question then arises as to how the DNN feature extractor can be trained. The most accurate existing approaches combine the training of the DNN with the clustering objective, so that information from the clustering process can be used to update the DNN to produce better features for clustering. One problem with this approach is that these "pseudo-labels" produced by the clustering algorithm are noisy, and any errors that they contain will hurt the training of the DNN. In this paper, we propose selective pseudo-label clustering, which uses only the most confident pseudo-labels for training the DNN. We formally prove the performance gains under certain conditions. Applied to the task of image clustering, the new approach achieves a state-of-the-art performance on three popular image datasets.

1 Introduction

Clustering is the task of partitioning a dataset into clusters such that data points within the same cluster are similar to each other, and data points from different clusters are different to each other. It is applicable to any set of data for which there is a notion of similarity between data points. It requires no prior knowledge, neither the explicit labels of supervised learning nor the knowledge of expected symmetries and invariances leveraged in self-supervised learning.

The result of a successful clustering is a means of describing data in terms of the cluster that they belong to. This is a ubiquitous feature of human cognition. For example, we hear a sound and think of it as an utterance of the word "water", or we see a video of a biomechanical motion and think of it as a jump. This can be further refined among experts, so that a musician could describe a musical phrase as an English cadence in A major, or a dancer could describe a snippet of ballet as a right-leg fouette into arabesque. When clustering high-dimensional data, the curse of dimensionality [2] means that many classic algorithms, such as k-means [29] or expectation maximization [10], perform poorly. The Euclidean distance, which is the basis for the notion of similarity in the Euclidean space, becomes weaker in higher dimensions [51]. Several solutions to this problem have been proposed. In this paper, we consider those termed deep clustering.

Deep clustering is a set of techniques that use a DNN to encode the high-dimensional data into a lower-dimensional feature space, and then perform clustering in this feature space. A major challenge is the training of the encoder. Much of the success of DNNs as image feature extractors (including [24,46]) has been in supervised settings, but if we already had labels for our data, then there would be no need to cluster in the first place. There are two common approaches to training the encoder. The first is to use the reconstruction loss from a corresponding decoder, i.e., to train it as an autoencoder [47]. The second is to design a clustering loss, so that the encoding and the clustering are optimized jointly. Both are discussed further in Section 2.

Our model, *selective pseudo-label clustering* (*SPC*), combines reconstruction and clustering loss. It uses an ensemble to select different loss functions for different data points, depending on how confident we are in their predicted clusters.

Ensemble learning is a function approximation where multiple approximating models are trained, and then the results are combined. Some variance across the ensemble is required. If all individual approximators were identical, there would be no gain in combining them. For ensembles composed of DNNs, variance is ensured by the random initializations of the weights and stochasticity of the training dynamics. In the simplest case, the output of the ensemble is the average of each individual output (mean for regression and mode for classification) [36].

When applying an ensemble to clustering problems (referred to as consensus clustering; see [3] for a comprehensive discussion), the sets of cluster labels must be aligned across the ensemble. This can be performed efficiently using the Hungarian algorithm. SPC considers a clustered data point to be confident if it received the same cluster label (after alignment) in each member of the ensemble. The intuition is that, due to random initializations and stochasticity of training, there is some non-zero degree of independence between the different sets of cluster labels, so the probability that all cluster labels are incorrect for a particular point is less than the probability that a single cluster label is incorrect.

Our main contributions are briefly summarized as follows.

- We describe a generally applicable deep clustering method (SPC), which treats cluster assignments as pseudo-labels, and introduces a novel technique to increase the accuracy of the pseudo-labels used for training. This produces a better feature extractor, and hence a more accurate clustering.
- We formally prove the advantages of SPC, given some simplifying assumptions. Specifically, we prove that our method does indeed increase the accuracy of the targets used for pseudo-label training, and this increase in accuracy does indeed lead to a better clustering performance.
- We implement SPC for image clustering, with a state-of-the-art performance on three popular image clustering datasets, and we present ablation studies on its main components.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. Sections 3 and 4 give a detailed description of SPC and a proof of correctness, respectively. Section 5 presents and discusses our experimental results,

including a comparison to existing image clustering models and ablation studies on main components of SPC. Finally, Section 6 summarizes our results and gives an outlook on future work. Full proofs and further details are in the appendix.

2 Related Work

One of the first deep image clustering models was [19]. It trains an autoencoder (AE) on reconstruction loss (rloss), and then clusters in the latent space, using loss terms to make the latent space more amenable to clustering.

In [44], the training of the encoder is integrated with the clustering. A second loss function is defined as the distance of each encoding to its assigned centroid. It then alternates between updating the encoder and clustering by k-means. A different differentiable loss is proposed in [43], based on a soft cluster assignment using Student's t-distribution. The method pretrains an AE on rloss, then, like [44], alternates between assigning clusters and training the encoder on cluster loss. Two slight modifications were made in later works: use of rloss after pretraining in [16] and regularization to encourage equally-sized clusters in [14].

This alternating optimization is replaced in [13] by a clustering loss that allows cluster centroids to be optimized directly by gradient descent.

Pseudo-label training is introduced by [6]. Cluster assignments are interpreted as pseudo-labels, which are then used to train a multilayer perceptron on top of the encoding DNN, training alternates between clustering encodings, and treating these clusters as labels to train the encoder.

Generative adversarial networks [15] (GANs) have produced impressive results in image synthesis [22,12,5]. At the time of writing, the most accurate GAN-based image clustering models [35,11] design a generator to sample from a latent space that is the concatenation of a multivariate normal vector and a categorical one-hot encoding vector, then recover latent vectors for the input images as in [9,28], and cluster the latent vectors. A similar idea is employed in [21], though not in an adversarial setting. For more details on GAN-based clustering, see [50,49,11,27,40] and the references therein.

Adversarial training is used for regularization in [33]. In [34], the method is developed. Conflicted data points are identified as those whose maximum probability across all clusters is less than some threshold, or whose max and next-to-max are within some threshold of each other. Pseudo-label training is then performed on the unconflicted points only. A similar threshold-based filtering method is employed by [7].

A final model to consider is [30], which uses a second round (i.e., after the DNN) of dimensionality reduction via UMAP [32], before clustering.

3 Method

Pseudo-label training is an effective deep clustering method, but training on only partially accurate pseudo-labels can hurt the encoder's ability to extract

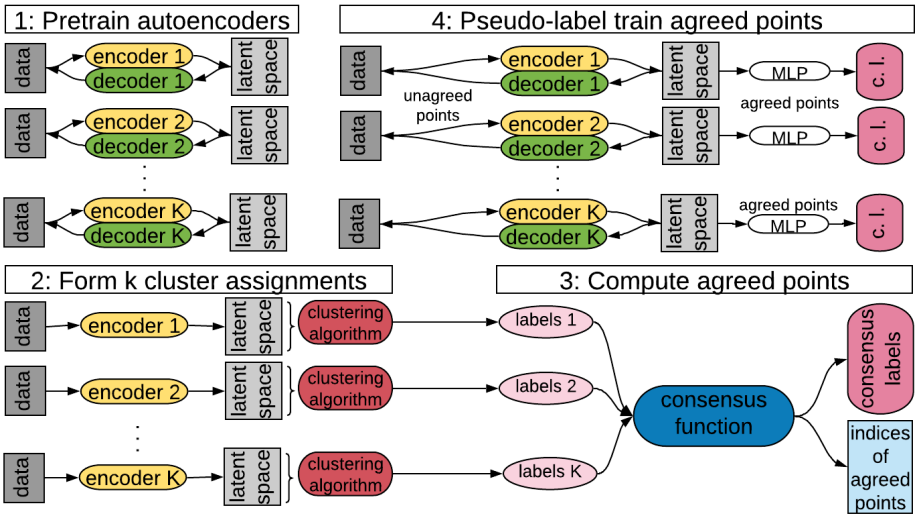

Fig. 1: The complete SPC method. (1) Pretrain autoencoders. (2) Perform multiple clusterings independently. (3) Identify agreed points as those that receive the same label in all ensemble members. (4) Perform pseudo-label training on agreed points and autoencoder training on unagreed points. Steps (2)–(4) are looped until the number of agreed points stops increasing.

relevant features. Selective pseudo-label clustering (SPC) addresses this problem by selecting only the most confident pseudo-labels for training, using the four steps shown in Fig. 1.

1. Train K autoencoders in parallel.
2. Cluster in the latent space of each, to obtain K sets of pseudo-labels.
3. Select for pseudo-label training, those points are those that received the same label in all K sets of pseudo-labels, after the labellings have been aligned using the Hungarian algorithm.
4. Train on the selected pseudo-labels. Go back to (2).

Training ends when the number of agreed points stops increasing. Then, each data point is assigned its most common cluster label across the (aligned) ensemble.

3.1 Formal Description

Given a dataset $\mathcal{X} \subseteq \mathbb{R}^n$ of size N with C true clusters, let $(f_j)_{1\leq j\leq K}, f_j : \mathbb{R}^n \to \mathbb{R}^m$, and $(g_j)_{1\leq j\leq K}, g_j : \mathbb{R}^m \to \mathbb{R}^n$ be the K encoders and decoders, respectively. Let $\psi : \mathbb{R}^{N\times m} \to \{0,\dots,C-1\}^N$ be the clustering function, which takes the N encoded data points as input, and returns a cluster label for each. We refer to the output of ψ as a labelling. Let $\Gamma : \{0,\dots,C-1\}^{K\times N} \to \{0,\dots,C-1\}^N \times \{0,1\}^N$ be the consensus function, which aggregates K different labellings of $\mathcal{X}$ into a single labelling, and also returns a Boolean vector indicating agreement. Then,

$$(c_1,\dots,c_N),(a_1,\dots,a_N) = \Gamma(\psi(f_1(\mathcal{X})) \circ \cdots \circ \psi(f_K(\mathcal{X}))), \tag{1}$$

Algorithm 1 Training algorithm for SPC

for $j = 1, \ldots, K$ **do**
  Update parameters of f_j and g_j using autoencoder reconstruction
end for
while number of agreed points increases **do**
  compute $(c_1, \ldots, c_N), (a_1, \ldots, a_N)$ as in (1)
  for $j = 1, \ldots, K$ **do**
    Update parameters of f_j and h_j to minimize (2)
  end for
end while

where $(c_1, \ldots, c_N)$ are the consensus labels, and $a_i = 1$ if the i-th data point received the same cluster label (after alignment) in all labellings, and 0 otherwise. The consensus function is the ensemble mode average, c_i is the cluster label that was most commonly assigned to the i-th data point.

Define K pseudo-classifiers $(h_j)_{1 \leq j \leq K}, h_j : \mathbb{R}^m \to \mathbb{R}^C$, and let

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \begin{cases} CE(h_j(f_j(x_i)), c_i) & a_i = 1 \\ ||g_j(f_j(x_i)) - x_i|| & \text{otherwise,} \end{cases} \tag{2}$$

where CE denotes categorical cross-entropy:

$$CE : \mathbb{R}^C \times \{0, \ldots, C-1\} \to \mathbb{R}$$
$$CE(x, n) = -\log(x[n])\,.$$

First, we pretrain the autoencoders, then compute $(c_1, \ldots, c_N), (a_1, \ldots, a_N)$ and minimize $\mathcal{L}$, recompute, and iterate until the number of agreed points stops increasing. The method is summarized in Algorithm 1.

Figure 2 shows the training dynamics. Agreed points are those that receive the same cluster label in all members of the ensemble. As expected, the agreed points' accuracy is higher than the accuracy on all points. Initially, the agreed points will not include those that are difficult to cluster correctly, such as an MNIST digit 3 that looks like a digit 5. Some ensemble members will cluster it as a 3 and others as a 5. The training process aims to make these difficult points into agreed points, thus increasing the fraction of agreed points, without decreasing the agreed points' accuracy. Figure 2 shows that this aim is achieved. As more points become agreed (black dotted line), the total accuracy approaches the agreed accuracy. The agreed accuracy remains high, decreasing only very slightly (blue line). The result is that the total accuracy increases (orange line). We end training when the number of agreed points plateaus.

3.2 Implementation Details

Encoders are stacks of convolutional and batch norm layers; decoders of transpose convolutional layers. Decoders have a *tanh* activation on their output layer, all

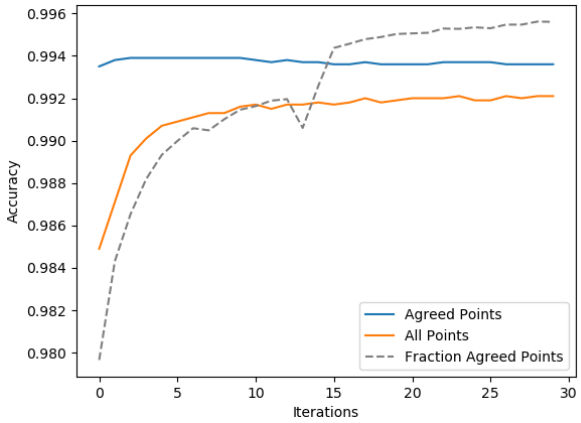

Fig. 2: Iterations of (2)–(4) in Figure 1 on MNIST.

other layers use leaky ReLU. The MLP pseudo-classifier has a hidden layer of size 25. The latent space of the autoencoders has the size 50 for MNIST and FashionMNIST, and 20 for smaller USPS. We inject noise from a multivariate normal into the latent space as a simple form of regularization. As suggested in [48], the reconstruction loss is ℓ_1. The architectures are the same across the ensemble, diversity comes from random initialization and training dynamics.

The clustering function (ψ above) is a composition of UMAP [32] and either HDBSCAN [31] or a Gaussian mixture model (GMM). As in previous works, we set the number of clusters to the ground truth. UMAP uses the parameters suggested in the clustering documentation clustering, $n\_neighbours$ is 30 for MNIST and scaled in proportion to the dataset size for the others. HDBSCAN uses all default parameters. We cut the linkage tree at a level that gives the correct number of clusters. On the rare occasions when no such cut can be found, the clustering is excluded from the ensemble. The GMM uses all default parameters.

Consensus labels are taken as the most common across the ensemble, after alignment with the Hungarian algorithm (called the "direct" method in [3]).

4 Proof of Correctness

Proving correctness requires proving that the expected accuracy of the agreed pseudo-labels is higher than that of all pseudo-labels, and that training with more accurate pseudo-labels makes the latent vectors easier to cluster correctly.

4.1 Agreed Pseudo-Labels are More Accurate

Given that each member of the ensemble is initialized independently at random, and undergoes different stochastic training dynamics, we can assume that each cluster assignment contains some unique information. Formally, there is strictly positive conditional mutual information between any one assignment in the

ensemble and the true cluster labels, conditioned on all the other assignments in the ensemble. From this assumption, the reasoning proceeds as follows.

Choose an arbitrary data point x_0 and cluster c_0. Let X be a random variable (r.v.), indicating the true cluster of x_0, given that n members of the ensemble have assigned it to c_0, and other assignments are unknown, $n \geq 0$. Thus, the event $X = c_0$ is the event that x_0 is correctly clustered. Let Y be a Boolean r.v. indicating that the $(n+1)$-th member of the ensemble also assigns it to c_0. Assume that, if n ensemble members have assigned x_0 to c_0, and other assignments are unknown, then x_0 belongs to c_0 with probability at least $1/C$ and belongs to all other clusters with equal probability, i.e.,

$$p(X = c_0) = t$$
$$\forall c \neq c_0, p(X = c) = (1-t)/(C-1)\,,$$

for some $1/C \leq t \leq 1$. It follows that the entropy $H(X)$ is a strictly decreasing function of t (see appendix for proof). Thus, the above assumption on conditional mutual information, written $I(X;Y) > 0$, is equivalent to $p(X = c_0|Y) > p(X = c_0)$. This establishes that the accuracy of the agreed labels is an increasing function of ensemble size. Standard pseudo-label training uses $n = 1$, whereas SPC uses $n > 1$ and so results in more accurate pseudo-labels for training.

4.2 Increased Pseudo-Label Accuracy Improves Clustering

Problem Formulation. Let $\mathcal{D}$ be a dataset of i.i.d. points from a distribution over $\mathcal{S} \in \mathbb{R}^n$, where $\mathcal{S}$ contains C true clusters $c_1, \ldots, c_C$. Let T be the r.v.defined by the identity function on S and $f : \mathcal{S} \to \mathbb{R}^m$, an encoding function parametrized by θ, whose output is an r.v. X. The task is to recover the true clusters conditional on X, and we are interested in choosing θ such that this task is as easy as possible. Pseudo-label training applies a second function $h : \mathbb{R}^m \to \{0, \ldots, C-1\}$ and trains the composition $h \circ f : \mathbb{R}^n \to \{0, \ldots, C-1\}$ using gradient descent (g.d.), with cluster assignments as pseudo-labels. The claim is that an increased pseudo-label accuracy facilitates a better choice of θ.

To formalize "easy", recall the definition of clustering as a partition that minimizes intra-cluster variance and maximizes inter-cluster variance. We want the same property to hold of the r.v. X. Let $y : \mathcal{D} \to \{0, \ldots, C-1\}$ be the true cluster assignment function and Y the corresponding random variable, then ease of recovering the true clusters is captured by a high value of d, where

$$d = \mathrm{Var}(\mathbb{E}[X|Y]) - \mathbb{E}[\mathrm{Var}(X|Y)]\,.$$

High d means that a large fraction of the variance of X is accounted for by cluster assignment, as, by Eve's law, we can decompose:

$$\mathrm{Var}\,(X) = \mathbb{E}[\mathrm{Var}(X|Y)] + \mathrm{Var}(\mathbb{E}[X|Y])\,, \tag{3}$$

In the following, we assume that f and g are linear, $C = 2$, $h \circ f(\mathcal{D}) \subseteq (0,1)$, and $\mathbb{E}[T] = \vec{0}$. The proof proceeds by expressing the value of d in terms of

expected distances between encoded points after a training step with correct labels and with incorrect labels, and hence proving that the value is greater in the former case. We show that the expectation is greater in each coordinate, from which the claim follows by linearity (see appendix for details).

Lemma 1. *Let $x, x' \in \mathcal{D}$ be two data points, and consider the expected squared distance between their encodings under f. Let u_{same} and u_{diff} denote the value of this difference after a g.d. update in which both labels are the same and after a step in which both labels are different, respectively. Then, $u_{same} < u_{diff}$.*

If $w \in \mathbb{R}^m$ and $w' \in \mathbb{R}$ are, respectively, the vector of weights mapping the input to the i-th coordinate of the latent space, and the scalar mapping the i-th coordinate of the latent space to the output, then the expected squared distance in the i-th coordinate of the latent vectors before the g.d. update is

$$\underset{x,x' \sim T}{E}[(w^T x - w^T x')^2] = \underset{x,x' \sim T}{E}[(w^T (x - x'))^2] \,.$$

When the two labels are the same, assume w.l.o.g. that $y = y' = 0$. Then, with step size η , the update for w and following expected squared difference u_{same} is

$$\begin{aligned} w &\leftarrow w - \eta(w'(x + x')) \\ u_{same} &= \underset{x,x' \sim T}{E}[((w - \eta w'(x + x'))^T (x - x'))^2] \\ &= \underset{x,x' \sim T}{E}[(w^T (x - x') - \eta w'(||x||^2 - ||x'||^2))^2] \,. \end{aligned}$$

When the two labels are different, assume w.l.o.g. that $y = 0$, $y' = 1$, giving

$$\begin{aligned} w &\leftarrow w - \eta(w'(x - x')) \\ u_{diff} &= \underset{x,x' \sim T}{E}[((w - \eta(w'(x - x'))^T)(x - x')])^2] = \\ &\underset{x,x' \sim T}{E}[(w^T (x - x') - \eta w' ||x - x'||^2)^2] \,. \end{aligned}$$

It can then be shown (see appendix) that $u_{same} < u_{diff}$.

Lemma 2. *Let z be a third data point, $z \in \mathcal{D}, z \neq x, x'$, and consider the expected squared distance of the encodings, under f, of x and z. Let v_{same} and v_{diff} denote, respectively, the value of this difference after a g.d. update with two of the same labels, and with two different labels. Then, $v_{same} = v_{diff}$.*

Lemma 3. *Let s and r denote, respectively, the expected squared distance between the encodings, under f, of two points in the same cluster and between two points in different clusters. Then, there exist $\lambda_1, \lambda_2 > 0$ whose values do not depend on the parameters of f, such that $d = \lambda_1 r - \lambda_2 s$.*

For simplicity, assume that the clusters are equally sized. The argument can easily be generalized to clusters of arbitrary sizes. We then obtain

$$d = \frac{C - 1}{2C} r - \frac{2C - 1}{2C} s \,,$$

where C is the number of clusters (see appendix for proof).

Definition 1. Let $\tilde{y} : \mathcal{D} \to \{0, \ldots, C-1\}$ be the pseudo-label assignment function. For $d_i, d_j \in \mathcal{D}$, the pseudo-labels are *pairwise correct* iff $y(x_i) = y(x_j)$ and $\tilde{y}(x_i) = \tilde{y}(x_j)$, or $y(x_i) \neq y(x_j)$ and $\tilde{y}(x_i) \neq \tilde{y}(x_j)$.

Theorem 1. *Let d_T and d_F denote, respectively, the value of d after a g.d. step from two pairwise correct labels and from two pairwise incorrect labels, and let $x, x' \in \mathcal{D}$ as before. Then, $d_T > d_F$.*

Proof. Let r_T, s_T, and r_F, s_F be, respectively, the values of r and s after a g.d. step from two pairwise correct labels and from two pairwise incorrect labels. Consider two cases. If $y(x) = y(x')$, then $r_T = r_F$, by Lemma 2, and $s_T < s_F$, by Lemmas 1 and 2, so by Lemma 3, $d_T > d_F$. If $y(x) \neq y(x')$, then $s_T = s_F$, by Lemma 2, and $r_T > r_F$, by Lemmas 1 and 2, so again $d_T > d_F$, by Lemma 3.

The fraction of pairwise correct pairs is one measure of accuracy (Rand Index). Thus, training with more accurate pseudo-labels facilitates better clustering.

5 Experimental Results

Following previous works, we measure accuracy and normalized mutual information (NMI). Accuracy is computed by aligning the predicted cluster labels with the ground-truth labels using the Hungarian algorithm [25] and then calculating as in the supervised case. NMI, as in [41], is defined as $2I(\tilde{Y}; Y)/(H(\tilde{Y}) + H(Y))$, where $\tilde{Y}$, Y, $I(\cdot, \cdot)$, and $H(\cdot)$ are, respectively, the cluster labels, ground truth labels, mutual information, and Shannon entropy. We report on two handwritten digits datasets, MNIST (size 70000) [26] and USPS (size 9298) [20], and Fashion-MNIST (size 70000) [42] of clothing items. Table 1 shows the central tendency for five runs and the best single run.

We show results for two different clustering algorithms: Gaussian mixture model and the more advanced HDBSCAN [31]. Both perform similarly, showing robustness to clustering algorithm choice. SPC-GMM performs slightly worse on USPS and FashionMNIST (though within margin of error), suggesting that HDBSCAN may cope better with the more complex images in FashionMNIST and the smaller dataset in USPS. In Table 1, 'SPC' uses HDBSCAN.

SPC (using either clustering algorithm) outperforms all existing approaches for both metrics on MNIST and USPS, and for NMI on FashionMNIST. The disparity between the two metrics, and between HDBSCAN and GMM, on FashionMNIST is due to the variance in cluster size. Many of the errors are lumped into one large cluster, and this hurts accuracy more than NMI, because being in this large cluster still conveys some information about what the ground truth cluster label is (see appendix for full details).

The most accurate existing methods use data augmentation. This is to be expected, given the well-established success of data augmentation in supervised learning [18]. More specifically, [17] have shown empirically that adding data augmentation to deep image clustering models improves performance in virtually all cases. Here, its effect is especially evident on the smaller dataset, USPS. For

	MNIST		USPS		FashionMNIST	
	ACC	NMI	ACC	NMI	ACC	NMI
SPC-best	**99.21**	**97.49**	**98.44**	**95.44**	*67.94*	**73.48**
SPC	**99.03 (.1)**	**97.04 (.25)**	**98.40 (.94)**	**95.42 (.15)**	65.58 (2.09)	**72.09 (1.28)**
SPC-GMM	**99.05 (.2)**	**97.10 (.47)**	**98.18 (.14)**	**94.93 (.32)**	65.03 (1.54)	**69.51 (1.21)**
DynAE [34]†	*98.7*	*96.4*	*98.1*	94.8	59.1	64.2
ADC [33]†	98.6	96.1	*98.1*	94.8	58.6	66.2
DDC [39]†	98.5	96.1	97.0	*95.3*	57.0	63.2
n2d [30]	97.9	94.2	95.8	90.0	67.2	*68.4*
DLS [11]	97.5	93.6	-	-	**69.3**	66.9
JULE [45]	96.4	91.3	95.0	91.3	56.3*	60.8*
DEPICT [14]	96.5	91.7	96.4	92.7	39.2*	39.2*
DMSC [1]†	95.15	92.09	95.15	92.09	-	-
ClusterGAN [35]	95	89	-	-	63	64
VADE [21]	94.5	87.6*	56.6*	51.2*	57.8*	63.0*
IDEC [16]	88.06	86.72	76.05	78.46	52.9*	55.7*
CKM [13]	85.4	81.4	72.1	70.7	-	-
DEC [43]	84.3	83.4*	76.2*	76.7*	51.8*	54.6*
DCN [44]	83	81	68.8*	68.3*	50.1*	55.8*

† =uses data augmentation *=results taken from [34]

Table 1: Accuracy and NMI of SPC compared to other top-performing image clustering models. The best results are in bold, and the second-best are emphasized. We report the mean and standard deviation (in parentheses) for five runs.

example, on MNIST, n2d [30] (which does not use data augmentation) is only 0.6 and 1.9 behind DDC [39], which does on ACC and NMI, respectively, but is 1.2 and 5.3 behind on USPS. SPC could easily be extended to include data augmentation, and even without using it, outperforms models that do.

5.1 Ablation Studies

Table 2 shows the effect of removing each component of our model. All settings use HDBSCAN. Particularly relevant are rows 2 and 3. As described in Section 3, we produce multiple labellings of the dataset and select for pseudo-label training only those data points that received the same label in all labellings. We perform two different ablations on this method: A1 and A2. Both use all data points for training, but A1 trains each ensemble on all data points using the labels computed in that ensemble member, and A2 uses the consensus labels. At inference, both use consensus labels. The significant drop in accuracy in both settings demonstrates that the strong performance of SPC is not just due to the application of an ensemble to existing methods, but rather to the novel method of label selection.

It is interesting to observe that A1 performs worse than A2 on MNIST and USPS. Combining approximations in an ensemble has long been observed to give higher expected accuracy ([8,37,38,4]), so the training targets would be more accurate in A1 than in A2. We hypothesize that the reason that this fails to translate to improved clustering is a reduction in ensemble variance. On MNIST

	MNIST		USPS		FashionMNIST	
	ACC	NMI	ACC	NMI	ACC	NMI
SPC	**99.03 (.1)**	**97.04 (.25)**	**98.40 (.94)**	**95.42 (.15)**	**65.58 (2.09)**	**72.09 (1.28)**
A1	98.01 (.04)	94.46 (.11)	97.03 (.65)	92.43 (1.29)	63.12 (.16)	70.59 (.01)
A2	98.18 (.05)	94.86 (.09)	97.31 (.89)	92.99 (1.84)	60.60 (4.45)	68.77 (.48)
A3	98.02 (.19)	94.45 (.43)	95.85 (.80)	89.77 (1.65)	59.23 (3.58)	67.09 (3.77)
A4	97.88 (.72)	94.8 (.85)	87.49 (7.93)	82.68 (2.6)	61.2 (4.28)	67.28 (1.72)
A5	96.17 (.26)	91.07 (.23)	87.00 (8.88)	80.79 (7.43)	55.29 (3.54)	66.07 (1.04)
A6	70.24	77.42	70.46	71.11	42.08	49.22

Table 2: Ablation results, central tendency for three runs. A1=w/o label filtering; A2=w/o label sharing; A3=w/o ensemble; A4=pseudo-label training only; A5=UMAP+AE; A6=UMAP. Both A1 and A2 train on all data points. The former trains each member of the ensemble on their own labels, and the latter uses the consensus labels. A3 sets $K = 1$, in the notation of Section 3.1.

	MNIST		USPS		FashionMNIST	
	ACC	NMI	ACC	NMI	ACC	NMI
25	98.48	95.60	97.70	93.82	67.67	73.25
20	98.49	95.64	97.87	94.21	67.52	73.13
15	99.03 (.10)	97.04 (.25)	98.40 (.94)	95.42 (.15)	65.58 (2.09)	72.09 (1.28)
12	98.82	96.54	98.20	95.02	67.77	73.13
10	98.78	96.42	98.39	95.47	62.93	69.89
8	98.75	96.32	98.41	95.44	67.45	71.99
6	98.61	95.90	98.40	95.39	63.84	70.62
5	98.56	95.82	98.30	95.19	67.91	73.46
4	98.47	95.60	98.27	95.18	67.90	73.38
3	98.44	95.50	98.15	94.84	63.36	70.88
2	98.27	95.07	97.98	94.40	62.9	70.41
1	98.02 (.19)	94.45 (.43)	95.85 (.80)	89.77 (1.65)	59.23 (3.58)	67.09 (3.77)

Table 3: Ablation studies on the size of the ensemble.

and USPS, high accuracy across the ensemble means high agreement. Giving the same training signal for every data point reduces variance further. Especially, compared with A2, the reduction is greatest on incorrectly clustered data points, because most incorrectly clustered data points are non-agreed points, and as argued in [23], high ensemble variance in the errors is important for performance.

A4 clusters in the latent space of one untrained encoder and then pseudo-label trains (essentially the method in [6]). It performs significantly worse than SPC, showing the value of the decoder, and of SPC's label selection technique.

A3 omits the ensemble entirely. Comparing with A2 again shows that the ensemble itself only produces a small improvement. Alongside SPC's label selection method, the improvement is much greater.

5.2 Ensemble Size

The number of autoencoders in the ensemble, K in the terminology of Section 3.1, is a hyperparameter. We add the concatenation of all latent spaces as an additional element. Table 3 shows the performance for smaller ensemble sizes. In MNIST and USPS, where the variance is reasonably small, there is a discernible trend of the performance increasing with K, then plateauing and starting to decrease. For FashionMNIST, where the variance is higher, the pattern is less clear. For all three datasets, however, we can see a significant difference between an ensemble of size two and an ensemble of size one (i.e., no ensemble). We hypothesize that the decrease for $K = 20, 25$ is due to a decrease in the number of agreed points, and so fewer pseudo-labels to train the encoders.

6 Conclusion

This paper has presented a deep clustering model, called selective pseudo-label clustering (SPC). SPC employs pseudo-label training, which alternates between clustering features extracted by a DNN, and treating these clusters as labels to train the DNN. We have improved this framework with a novel technique for preventing the DNN from learning noise. The method is formally sound and achieves a state-of-the-art performance on three popular image clustering datasets. Ablation studies have demonstrated that the high accuracy is not merely the result of applying an ensemble to existing techniques, but rather is due to SPC's novel filtering method. Future work includes the application to other clustering domains, different from images, and an investigation of how SPC combines with existing deep clustering techniques.

Acknowledgments. This work was supported by the Alan Turing Institute under the UK EPSRC grant EP/N510129/1 and by the AXA Research Fund. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1) and GPU computing support by Scan Computers International Ltd.

References

1. Abavisani, M., Patel, V.M.: Deep multimodal subspace clustering networks. IEEE Journal of Selected Topics in Signal Processing **12**(6), 1601–1614 (2018)
2. Bellman, R.: Dynamic programming. Science **153**(3731), 34–37 (1966)
3. Boongoen, T., Iam-On, N.: Cluster ensembles: A survey of approaches with recent extensions and applications. Computer Science Review **28**, 1–25 (2018)
4. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. arXiv:1809.11096 (2018)
6. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proc. ECCV. pp. 132–149 (2018)
7. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. In: Proc. ICCV. pp. 5879–5887 (2017)

8. Clemen, R.T.: Combining forecasts: A review and annotated bibliography. International Journal of Forecasting **5**(4), 559–583 (1989)
9. Creswell, A., Bharath, A.A.: Inverting the generator of a generative adversarial network. IEEE Trans. Neural Netw. Learn. Syst. **30**(7), 1967–1974 (2018)
10. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc.: Series B (Methodol.) **39**(1), 1–22 (1977)
11. Ding, F., Luo, F.: Clustering by directly disentangling latent space. arXiv:1911.05210 (2019)
12. Elgammal, A., Liu, B., Elhoseiny, M., Mazzone, M.: CAN: Creative adversarial networks, generating art by learning about styles and deviating from style norms. arXiv:1706.07068 (2017)
13. Gao, B., Yang, Y., Gouk, H., Hospedales, T.M.: Deep clustering with concrete k-means. In: Proc. ICASSP. pp. 4252–4256. IEEE (2020)
14. Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proc. ICCV (2017)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. NIPS. pp. 2672–2680 (2014)
16. Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: Proc. IJCAI. pp. 1753–1759 (2017)
17. Guo, X., Zhu, E., Liu, X., Yin, J.: Deep embedded clustering with data augmentation. In: Proc. Asian Conference on Machine Learning. pp. 550–565 (2018)
18. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580 (2012)
19. Huang, P., Huang, Y., Wang, W., Wang, L.: Deep embedding network for clustering. In: Proc. ICPR. pp. 1532–1537. IEEE (2014)
20. Hull, J.J.: A database for handwritten text recognition research. TPAMI **16**(5), 550–554 (1994)
21. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: An unsupervised and generative approach to clustering. arXiv:1611.05148 (2016)
22. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proc. CVPR (2019)
23. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. TPAMI **20**(3), 226–239 (1998)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proc. NIPS. pp. 1097–1105 (2012)
25. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
27. Liang, J., Yang, J., Lee, H.Y., Wang, K., Yang, M.H.: Sub-GAN: An unsupervised generative model via subspaces. In: Proc. ECCV (2018)
28. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. arXiv:1702.04782 (2017)
29. Lloyd, S.: Least square quantization in PCM. IEEE Trans. Inf. Theory (1957/1982) **18** (1957)
30. McConville, R., Santos-Rodriguez, R., Piechocki, R.J., Craddock, I.: N2d:(Not too) deep clustering via clustering the local manifold of an autoencoded embedding. arXiv:1908.05968 (2019)

31. McInnes, L., Healy, J., Astels, S.: HDBSCAN: Hierarchical density based clustering. Journal of Open Source Software **2**(11), 205 (2017)
32. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426 (2018)
33. Mrabah, N., Bouguessa, M., Ksantini, R.: Adversarial deep embedded clustering: on a better trade-off between feature randomness and feature drift. arXiv:1909.11832 (2019)
34. Mrabah, N., Khan, N.M., Ksantini, R., Lachiri, Z.: Deep clustering with a dynamic autoencoder: From reconstruction towards centroids construction. arXiv:1901.07752 (2019)
35. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: ClusterGAN: Latent space clustering in generative adversarial networks. arXiv:1809.03627 (2019)
36. Opitz, D.W., Maclin, R.F.: An empirical evaluation of bagging and boosting for artificial neural networks. In: Proc. ICNN. vol. 3, pp. 1401–1405. IEEE (1997)
37. Pearlmutter, B.A., Rosenfeld, R.: Chaitin-Kolmogorov complexity and generalization in neural networks. In: Proc. NIPS. pp. 925–931 (1991)
38. Perrone, M.P.: Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization. Ph.D. thesis (1993)
39. Ren, Y., Wang, N., Li, M., Xu, Z.: Deep density-based image clustering. Knowledge-Based Systems p. 105841 (2020)
40. Wang, Y., Zhang, L., Nie, F., Li, X., Chen, Z., Wang, F.: WeGAN: Deep image hashing with weighted generative adversarial networks. IEEE Trans. Multimed. (2019)
41. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques with Java implementations. ACM Sigmod Record **31**(1), 76–77 (2002)
42. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
43. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proc. ICML. pp. 478–487 (2016)
44. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: Proc. ICML,Volume 70. pp. 3861–3870. JMLR.org (2017)
45. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proc. CVPR. pp. 5147–5156 (2016)
46. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proc. ECCV. pp. 818–833. Springer (2014)
47. Zemel, R.S., Hinton, G.E.: Developing population codes by minimizing description length. In: Proc. NIPS. pp. 11–18 (1994)
48. Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for neural networks for image processing. arXiv:1511.08861 (2015)
49. Zhao, W., Wang, S., Xie, Z., Shi, J., Xu, C.: GAN-EM: GAN based EM learning framework. arXiv:1812.00335 (2018)
50. Zhou, P., Hou, Y., Feng, J.: Deep adversarial subspace clustering. In: Proc. CVPR (2018)
51. Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal **5**(5), 363–387 (2012)

7 Appendix A: Full Proofs

This appendix contains the full proofs of the results in Section 4.

7.1 More Accurate Pseudo-Labels Supplement

The only part omitted from the argument in the main paper is a proof for the claim about the entropy of the random variable X. This is supplied by the following proposition.

Proposition 1. *Given a categorical random variable X of the form*

$$\begin{aligned} p(X = c_0) &= t \\ \forall c \neq c_0, p(X = c) &= \frac{1-t}{C-1}\,, \end{aligned}$$

for some $1/C \leq t \leq 1$, the entropy $H(X)$ is a strictly decreasing function of t.

Proof.

$$\begin{aligned} H(X) &= -\,t \log t - (1-t) \log \frac{1-t}{C-1} \\ \frac{d(H(X))}{dt} &= -\log t - 1 - \frac{1}{1-t} + \log \frac{1}{C-1} + \\ &\quad + \frac{t}{1-t} + \log 1 - t \\ &= -\,2 - \log t - \log C - 1 + \log t - 1 \\ &= -\,2 - \log \left(\frac{t}{1-t}(C-1) \right). \end{aligned}$$

The argument to the log is clearly an increasing function of t for $t > 1$. Therefore, for $1/C \leq t < 1$, it is lower-bounded by setting $t = 1/C$. This gives

$$\begin{aligned} \frac{d(H(X))}{dt} &\leq -2 - \log \left(\frac{1/C}{1-1/C}(C-1) \right) \\ &< -\log \left(\frac{1/C}{1-1/C}(C-1) \right) = -\log 1 = 0\,. \end{aligned}$$

The derivative is always strictly negative with respect to t, so, as a function of t, $H(X)$ is always strictly decreasing.

7.2 Lemma 1 Supplement

The following is a proof for the claim that $u_{same} < u_{diff}$, as stated in Section 4.

Decomposing u_{same} according to the definition of variance (as the expectation of the square minus the square of the expectation) gives

$$\begin{aligned}&\underset{x,x'\sim T}{E}[w^T(x-x') - \eta w'(||x||^2 - ||x'||^2)]^2 + \\ &\quad \mathrm{Var}(w^T(x-x') - \eta w'(||x||^2 - ||x'||^2))\,.\end{aligned}$$

The expectation term equals 0, as

$$\begin{aligned}&w^T \underset{x,x'\sim T}{E}[(x-x')] - \eta w' \underset{x,x'\sim T}{E}[(||x||^2 - ||x'||^2)] = \\ &(w\mathbb{E}[T] - \mathbb{E}[T]) - \eta w'(\mathbb{E}[||T||^2] - \mathbb{E}[||T||^2]) = 0\,.\end{aligned}$$

By symmetry, we can replace covariances involving x' with the same involving x. The remaining term can then be rearranged to give

$$\begin{aligned}&u_{same} = 2\mathrm{Var}(w^T x - \eta w' ||x||^2) \\ &= 2w^T Cov(T) w + 2\eta w' Var(||x||^2) - 4\mathrm{Cov}(w^T x, \eta w' ||x||^2)\,.\end{aligned}$$

Now rewrite u_{diff}. Decomposing as above gives

$$\begin{aligned}&\underset{x,x'\sim T}{E}[w^T(x-x') - \eta w'(||x-x'||^2)]^2 + \\ &\quad \mathrm{Var}(w^T(x-x') - \eta w'(||x-x'||^2))\,,\end{aligned}$$

and here the expectation term does not equal 0:

$$\begin{aligned}&(w^T \underset{x,x'\sim T}{E}[(x-x')] - \eta w' \underset{x,x'\sim T}{E}[(||x-x'||^2)])^2 = \\ &\quad (\eta w')^2 \underset{x,x'\sim T}{E}[||x-x'||^2]^2\,.\end{aligned}$$

The variance term can be expanded to give:

$$\begin{aligned}&\mathrm{Var}(w^T(x-x') - \eta w'(||x-x'||^2)) = \\ &2w^T Cov(T) w + 2\eta w' \mathrm{Var}(||x-x'||^2) - \\ &\quad 4\mathrm{Cov}(w^T x, \eta w' ||x-x'||^2)\,.\end{aligned}$$

By comparing terms, we can see that this expression is at least as large as u_{same}. First, consider the covariance terms.

Claim. $\mathrm{Cov}(w^T x, \eta w' ||x-x'||^2) = \mathrm{Cov}(w^T x, \eta w' ||x||^2)$.

$$
\begin{aligned}
&\mathrm{Cov}(w^T x, \eta w' ||x - x'||^2) \\
&= \mathbb{E}[w^T x \eta w' ||x - x'||^2] - \mathbb{E}[w^T x]\mathbb{E}[\eta w' ||x - x'||^2] \\
&= \eta w' \mathbb{E}[w^T x ||x - x'||^2] - 0\mathbb{E}[\eta w' ||x - x'||^2] \\
&= \eta w' \mathbb{E}[w^T x ||x - x'||^2] \\
&= \eta w' \mathbb{E}[w^T x \sum_k x^2 - 2xx' + x^{'2}] \\
&= \eta w' \sum_k \mathbb{E}[w^T x x_k^2] - 2\mathbb{E}[w^T x x_k]\mathbb{E}[x'] + \mathbb{E}[w^T x]\mathbb{E}[x^{'2}] \\
&= \eta w' \sum_k \mathbb{E}[w^T x x_k^2] - 2\mathbb{E}[w^T x x_k]\vec{0} + 0\mathbb{E}[x^{'2}] \\
&= \eta w' \sum_k \mathbb{E}[w^T x x_k^2] \\
&= \eta w' \mathbb{E}[w^T x \sum_k x_k^2] \\
&= \eta w' \mathbb{E}[w^T x ||x||^2] \\
&= \eta w' \mathbb{E}[w^T x ||x||^2] - 0\mathbb{E}[\eta w' ||x||^2] \\
&= \mathbb{E}[w^T x \eta w' ||x||^2] - \mathbb{E}[w^T x]\mathbb{E}[\eta w' ||x||^2] \\
&= \mathrm{Cov}(w^T x, \eta w' ||x||^2)\,.
\end{aligned}
$$

So, we see the covariance terms are equal.

Next, compare the second variance terms

Claim. $\mathrm{Var}(||x - x'||^2) \geq \mathrm{Var}(||x||^2)$.

$$
\begin{aligned}
&\mathrm{Var}(||x - x'||^2) \\
=&\mathrm{Var}\left(\sum_{k=0}^{nz}(x)_k^2 + (x')_k^2 - 2(x)_k(x')_k\right) \\
=& \mathrm{Var}\left(\sum_{k=0}^{nz}(x)_k^2\right) + \mathrm{Var}\left(\sum_{k=0}^{nz}(x')_k^2\right) + 2\,\mathrm{Var}\left(\sum_{k=0}^{nz} x_k x'_k\right) \\
=& 2\mathrm{Var}\left(\sum_{k=0}^{nz}(x)_k^2\right) + 2\,\mathrm{Var}\left(\sum_{k=0}^{nz} x_k x'_k\right) \\
=& 2(\mathrm{Var}(||x||^2) + \mathrm{Var}(x^T x')) \\
\geq& \mathrm{Var}(||x||^2)\,.
\end{aligned}
$$

Assuming that the data are not all identical, this implies that u_{diff} is strictly greater than u_{same}.

$$
\begin{aligned}
& u_{diff} - u_{same} \\
&= (\eta w')^2 \underset{x,x' \sim T}{E}[||x - x'||^2]^2 + 2w^T \operatorname{Cov}(T) w + \\
&\quad + 2\eta w' \operatorname{Var}(||x - x'||^2) - 4 \operatorname{Cov}(w^T x, \eta w' ||x - x'||^2) - \\
&\quad - ((2w^T \operatorname{Cov}(T) w + 2\eta w' Var(||x||^2) \\
&\quad - 4 \operatorname{Cov}(w^T x, \eta w' ||x||^2))) \\
&= (\eta w')^2 \underset{x,x' \sim T}{E}[||x - x'||^2]^2 + \\
&\quad + 2\eta w' \left(\operatorname{Var}(||x - x'||^2) - \operatorname{Var}(||x||^2) \right) - \\
&\quad - 4 \left(\operatorname{Cov}(w^T x, \eta w' ||x - x'||^2) - \operatorname{Cov}(w^T x, \eta w' ||x||^2) \right) \\
&= (\eta w')^2 \underset{x,x' \sim T}{E}[||x - x'||^2]^2 + \\
&\quad + 2\eta w' \left(\operatorname{Var}(||x - x'||^2) - \operatorname{Var}(||x||^2) \right) \\
&\geq (\eta w')^2 \underset{x,x' \sim T}{E}[||x - x'||^2]^2 > 0 \,.
\end{aligned}
$$

7.3 Lemma 2 Supplement

The following is the complete proof of Lemma 2, which was omitted from the main paper.

Proof. $v_{diff} - v_{same}$

$$
\begin{aligned}
&= \mathbb{E}[(w^T(x - z) - w'(x - x')(x - z))^2] - \\
&\quad - \mathbb{E}[(w^T(x - z) - w'(x + x')(x - z))^2] \\
&= \mathbb{E}[(w^T(x - z) - w'(x - x')^T(x - z))^2 - \\
&\quad - (w^T(x - z) - w'(x + x')^T(x - z))^2] \\
&= \mathbb{E}[(w^T(x - z) - w'(x - x')^T(x - z) + \\
&\quad + w^T(x - z) - w'(x + x')^T(x - z)) \\
&\quad (w^T(x - z) - w'(x - x')^T(x - z) - \\
&\quad - w^T(x - z) - w'(x + x')^T(x - z))] \\
&= \mathbb{E}[(2w^T(x - z) - w'(x - z)^T(x - x' + x + x')) \\
&\quad (-w'(x - z)^T(x - x' - x - x'))] \\
&= \mathbb{E}[(2w^T(x - z) - 2w'(x - z)^T(x))(2w'(x - z)^T(x'))] \\
&= 2\mathbb{E}[(w^T(x - z) - w'(x - z)^T(x)) w'(x - z)^T] \mathbb{E}[x'] \\
&= 2\mathbb{E}[(w^T(x - z) - w'(x - z)(x)) w'(x - z)^T] \vec{0} = 0 \,.
\end{aligned}
$$

7.4 Lemma 3 Supplement

The following is the complete proof of Lemma 3, which was omitted from the main paper.

Proof.

$$\begin{aligned}
\mathrm{Var}(T) =& \frac{1}{2} \underset{x,x' \sim T}{E}[(x - x')^2] \\
=& \frac{1}{2} (\underset{x,x' \sim T}{E}[(x - x')^2 | y(x) = y(x')] P(y(x) = y(x')) + \\
& \underset{x,x' \sim T}{E}[(x - x')^2 | y(x) \neq y(x')] P(y(x) \neq y(x')) \\
=& \frac{1}{2} (sP(y(x) = y(x')) + rP(y(x) \neq y(x')) \\
=& \frac{1}{2} \left(s \frac{1}{C} + r \frac{C-1}{C} \right) .
\end{aligned}$$

Noting that $s = 2\mathbb{E}[\mathrm{Var}(T|C)]$, and using Eve's law, we have

$$\begin{aligned}
d =& \mathrm{Var}(T) - s \\
=& \frac{1}{2} \left(s \frac{1}{C} + r \frac{C-1}{C} \right) - s \\
=& \frac{C-1}{2C} r - \frac{2C-1}{2C} s .
\end{aligned}$$

7.5 Theorem 5 Supplement

The following is a more detailed version of the argument given in the main paper.

If $y(x) = y(x')$, then Lemma 2 means that the expected distance of the encodings of x and x' to any data point from another cluster is unchanged by whether the update was from points with the same or with different labels. Similarly, the distance between any two other points is unchanged by whether the update was from points with the same or with different labels. This establishes that $r_T = r_F$. As for the intra-cluster variance, it is smaller after the update with the same labels than with different labels. Lemma 1 shows that the expected distance between the encodings of the two points themselves is smaller if the labels were the same, and the same argument as above shows that all other expected distances within clusters are unchanged.

If $y(x) \neq y(x')$, then Lemma 2 means that the expected distance of the encodings of x and any data point from the same cluster is unchanged by whether the update was from points with the same or with different labels (and the same for x'). Similarly, the distance between any two other points is unchanged by whether the update was from points with the same or with different labels. This establishes that $s_T = s_F$. As for the *inter*-cluster variance, it is larger after the update with the same labels than with different labels. Lemma 1 shows that the

	Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine
HDBSCAN	6923	7878	6979	7095	6802	6290	6911	7384	6776	6962
GMM	6942	6958	6791	7885	6976	7096	7350	6294	6906	6802
Ground Truth	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000

Table 4: Sizes of predicted clusters for MNIST.

	Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine
HDBSCAN	1565	1272	933	819	856	706	833	787	693	834
GMM	1271	834	785	833	690	835	862	930	699	1559
Ground Truth	1553	1269	929	824	852	716	834	792	708	821

Table 5: Sizes of predicted clusters for USPS.

	Top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Boot
HDBSCAN	7411	6755	56	6591	21333	6046	3173	5666	3711	9258
GMM	6700	3111	16379	6807	6753	9127	7389	4482	8814	438
Ground Truth	7000	7000	7000	7000	7000	7000	7000	7000	7000	7000

Table 6: Sizes of predicted clusters for FashionMNIST.

expected distance between the encodings of the two points themselves is larger if the labels were different, and the same argument as above shows that all other expected distances within clusters are unchanged.

8 Appendix C: Extended Results

The results in the main paper report the central tendency of five different training runs for each dataset. Tables 4, 5, and 6 show the sizes of the clusters predicted by SPC for one randomly selected run out of these five. On MNIST and USPS, where the accuracy of SPC is $> 98\%$, the predicted sizes are close to the true sizes. On FashionMNIST, where the accuracy is $\sim 65\%$, there is a much greater variance. This accounts for the discrepancy in ACC and NMI for FashionMNIST. Most of the errors are put into one large cluster, specifically the cluster that was aligned to 'coat' is over three times larger than it should be. This hurts accuracy more than NMI, because the incorrect data points in the 'coat' cluster count for zero when calculating the accuracy, but they are not randomly distributed among the other classes, so the conditional entropy of a data point that was mis-clustered as a coat is $< \log(10)$. Actually, most of the mistakes in the 'coat' cluster are pullovers or shirts, and almost none of them are, for examples, boots or tops. Comparing the cluster sizes for SPC-HDBSCAN and SPC-GMM also accounts for the differences across ACC and NMI between these two settings on FashionMNIST: SPC-GMM produces more uniformly-sized clusters, so the difference between ACC and NMI is smaller.