Data-Efficient Contrastive Self-supervised Learning: Most Beneficial Examples for Supervised Learning Contribute the Least

Siddharth Joshi [1] **Baharan Mirzasoleiman** [1]

Abstract

Self-supervised learning (SSL) learns high-quality representations from large pools of unlabeled training data. As datasets grow larger, it becomes crucial to identify the examples that contribute the most to learning such representations. This enables efficient SSL by reducing the volume of data required. Nevertheless, quantifying the value of examples for SSL has remained an open question. In this work, we address this problem for the first time, by proving that examples that contribute the most to contrastive SSL are those that have the most similar augmentations to other examples, in expectation. We provide rigorous guarantees for the generalization performance of contrastive learning on such subsets. Through extensive experiments, we show that we can safely exclude 20% of examples from CIFAR100 and 40% from STL10 and TinyImageNet, without affecting downstream task performance. In general, subsets selected by our method outperform random subsets by over 3% across these datasets. Interestingly, we also discover the subsets that contribute the most to contrastive learning are those that contribute the least to supervised learning. Code available at https://github.com/bigml-cs-ucla/sas-data-efficient-contrastive-learning.

1. Introduction

Large datasets power modern machine learning models. However, a key question is: what data points are essential for learning and whether more data will always yield better performance? Answering this question is crucial as it can reduce the substantial costs of training on large datasets, boost performance of the trained models and guide data collection. This has motivated a body of recent research on finding the most essential subsets for supervised learning (Toneva et al., 2019; Paul et al., 2021; Mirzasoleiman et al., 2020; Mindermann et al., 2022; Sorscher et al., 2022; Swayamdipta et al., 2020). However, as datasets grow larger, obtaining high-quality labels for them becomes prohibitively expensive. As a result, there has been a surge in self-supervised (SSL) pretraining on large un-labeled dataset (Chen et al., 2020; Grill et al., 2020a; Chen & He, 2021; Zbontar et al., 2021). Nevertheless, finding the most important data points for SSL has remained an open question.

Finding the examples that contribute the most to SSL is indeed very challenging. When labels are available, the value of every example for learning can be quantified based on its loss (or confidence of the prediction) or gradient norm. Effectively, difficult-to-learn examples i.e. those with high loss or large gradient norm during training are the ones that contribute the most to minimizing the training loss. However, in the absence of labels, SSL methods cluster examples based on their similarity to the other data points. Therefore, the SSL loss and gradient of every example is tightly coupled with that of the other examples in the dataset. Hence, dropping an example affects the loss and gradient of all the other examples. This makes data selection inherently more challenging for SSL as compared to supervised learning.

In this work, we address the above challenge for the first time and find examples that provably contribute the most to SSL. In particular, we focus on *contrastive* SSL which learns representations by maximizing the alignment between augmented views of the same examples and minimizing the similarity between augmented views of different examples (Chen et al., 2020; Zbontar et al., 2021; Oord et al., 2018). We prove that examples that contribute the most to contrastive learning are those that have the highest expected similarity between their augmented views and the augmented views of other examples in their latent class. Effectively, such examples pull different groups of examples in a class together and enable the contrastive loss to maximally push away representations of examples in different classes. We show that such examples (1) ensure a high alignment between augmented views of examples in every class, and (2) preserve the centers of class representations learned by contrastive learning on the full data. We leverage

[1]Department of Computer Science, University of California Los Angeles, CA 90095, USA.. Correspondence to: Siddharth Joshi <sjoshi804@cs.ucla.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the above properties to provide a generalization guarantee for a linear classifier trained on the representations obtained by applying contrastive learning to the subset.

We observe that, perhaps surprisingly, examples that contribute the most to contrastive learning contribute the least to supervised learning. In particular, we quantify the difficulty of examples for supervised learning using confidence of the predictions as well as the forgetting score (Toneva et al., 2019), i.e. the number of times an examples is misclassified after being correctly classified during the training. We show that examples that contribute the most to contrastive learning are the easy examples with a high confidence and low forgetting score for supervised learning. Such examples can be safely excluded from a supervised learning pipeline, without harming the accuracy (Toneva et al., 2019). In contrast, difficult-to-learn examples that contribute the most to supervised learning can significantly hurt contrastive learning performance.

We extensively evaluate the performance of our method, *SAS*, which selects **S**ubsets that maximize **A**ugmentation **S**imilarity to the full data, on various datasets and using different contrastive learning methods. We first apply *SAS* to CIFAR10, CIFAR100 (Krizhevsky et al., 2009), STL10 (Coates et al., 2011a) and TinyImageNet (Le & Yang, 2015), with ResNet50 using SimCLR (Chen et al., 2020). We show that using *SAS*, up to 20% of examples from CIFAR100 and 40% from STL10 and TinyImageNet (Deng et al., 2009), can be safely excluded without harming the downstream performance. Similarly, for BYOL, using *SAS* to discard 20% of examples from STL10 can even outperform downstream performance of the full data by 2%. In general, *SAS* subsets outperform random subsets by over 3% across these datasets and methods including SimSiam (Chen & He, 2021), MoCo (He et al., 2020) and BYOL (Grill et al., 2020a). We also demonstrate that the subsets that contribute the most to SSL can be efficiently extracted can be efficiently extracted early-in-training or using a smaller proxy model.

2. Related Work

Contrastive Learning. Contrastive learning has recently emerged as a performant self-supervised framework to learn representations that capture semantically relevant information from the data. The key idea behind this family of algorithms is learning representations by maximizing agreement between augmented views of the same example (positive pairs) and minimizing agreement between augmented views of different examples (negative pairs) (Chen et al., 2020; Zbontar et al., 2021; Grill et al., 2020a; Chen & He, 2021; He et al., 2020). To improve the performance of contrastive learning, re-weighting the negative pairs in the contrastive loss (Chuang et al., 2020) or re-weighting the loss to emphasize the hard negatives (Robinson et al., 2020) has been recently explored. Here, we aim to find subsets of examples that contribute the most to contrastive learning. The above reweighting strategies are orthogonal to our work and can be applied to the subsets found by our method.

Contrastive Learning Theory. A recent line of theoretical works has studied contrastive learning. In particular, under conditional independence between positive pairs given the label, representations learned by contrastive learning algorithms can achieve small errors in the downstream linear classification task (Arora et al., 2019; Saunshi et al., 2019; Tosh et al., 2021). The independence assumption was relaxed by (HaoChen et al., 2021), which showed that minimizing spectral-based contrastive loss results in spectral clustering on the augmented distribution and provides generalization guarantee for linear evaluation. Wang & Isola (2020) proved that asymptotically, the contrastive loss optimizes alignment (similarity) of positive pairs and uniformity of the representations on the hypersphere, relating them to positive effects on downstream tasks. The recent result of (Huang et al., 2021) showed that contrastive learning using the more general InfoNCE (Oord et al., 2018) or cross-correlation loss (Zbontar et al., 2021) maximizes alignment of positive pairs as well as divergence of centers of the latent class representations. Here, we build on this work and show that subsets that contribute the most to contrastive learning introduce minimal error on the alignment and divergence of centers of class representations learned on the full data. Leveraging the above properties, we provide generalization guarantees for downstream performance of representations learned on such subsets.

Essential Subsets for Supervised Learning. There has been a recent body of efforts on finding the most important subsets for supervised learning. Empirical methods commonly rank examples from easiest to hardest—based on confidence, loss or gradient—and curate subsets preserving the hardest examples. Coleman et al. (2020) used a smaller trained proxy model to find the most uncertain examples to train a larger model. Toneva et al. (2019) selects examples with highest forgetting score, i.e., the number of times they transition from being classified correctly to incorrectly during training. Swayamdipta et al. (2020) selects examples with the highest variance of predictions during training. Paul et al. (2021) selects examples with the lowest expected gradient norm over multiple initializations. More theoretically motivated approaches iteratively select subsets by importance sampling based on gradient norm (Katharopoulos & Fleuret, 2018) or select weighted subset of examples which closely capture the full gradient (Mirzasoleiman et al., 2020; Pooladzandi et al., 2022; Killamsetty et al., 2021).

In contrast, we show, for the first time, that easy-to-learn examples with highest confidence and lowest forgetting score that contribute the least to supervised learning are the

most beneficial for unsupervised contrastive learning.

3. Problem Formulation

Assume we have a dataset $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i\in V}$ of $n=|V|$ training examples drawn i.i.d. from an unknown distribution. Each example belongs to one of the K latent classes i.e. $V = \{V_1 \cup \cdots \cup V_K\}$, but the corresponding class labels are not known at training time.

Contrastive Learning learns representations of examples in the training data, by learning an encoder f that maximizes agreement between representations of differently augmented views of the same example (i.e. positive pairs) and minimizes agreement between representations of augmented views of different examples (i.e. negative pairs). This is achieved by minimizing the following InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{cl}(V)=-\mathop{\mathbb{E}}_{i,j\in V}\mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)\\ \boldsymbol{x}^-\in A(\boldsymbol{x}_j)}}\log\frac{e^{f(\boldsymbol{x}_1)^T f(\boldsymbol{x}_2)}}{e^{f(\boldsymbol{x}_1)^T f(\boldsymbol{x}_2)}+e^{f(\boldsymbol{x}_1)^T f(\boldsymbol{x}^-)}}, \tag{1}$$

where $A(\boldsymbol{x})$ is the set of augmented views of example $\boldsymbol{x}$.

The performance of contrastive learning is evaluated by training a linear classifier on the learned representation using labels:

$$g_f^l(\boldsymbol{x}) = \arg\max_{k\in[K]}(\boldsymbol{W}f(\boldsymbol{x})+\boldsymbol{b})_k \tag{2}$$

However, to simplify the theoretical analysis, we follow (Huang et al., 2021) and consider a non-parametric nearest neighbor (NN) classifier:

$$g_f(\boldsymbol{x}) = \arg\min_{k\in[K]}\|f(\boldsymbol{x})-\boldsymbol{\mu}_k\|, \tag{3}$$

where $\boldsymbol{\mu}_k := \mathbb{E}_{i\in V_k}\mathbb{E}_{\boldsymbol{x}'\in A(\boldsymbol{x}_i)}[f(\boldsymbol{x}')]$ is the center of class V_k.

The linear classifier learned on the labels g_f^l is guaranteed to perform at least as well as the NN classifier g_f (Huang et al., 2021). Therefore, we use the classification error rate of the NN classifier to bound the worst-case performance of the linear classifier:

$$\xi(g_f(V)) = \sum_{k=1}^{K}\mathbb{P}[g_f(\boldsymbol{x}_i)\neq k, \forall i\in V_k]. \tag{4}$$

We note that in our experiments, we evaluate our method using the downstream accuracy of the *linear classifier*, and our theoretical guarantees on the NN classifier also upper-bound the error of the linear classifier.

Our goal is to find a subset $S\subseteq V$ of at most r training examples, such that the encoder $f^S = \arg\min_f \mathcal{L}_{cl}(S)$ obtained by minimizing the contrastive loss on the subset, allows the NN classifier to obtain a similar error on the *full data*. Formally, we aim to solve the following problem:

$$S^* = \mathop{\arg\min}_{S\subseteq V,|S|\leq r}[|\xi(g_{f^S}(V))-\xi(g_f(V))|]. \tag{5}$$

4. The Most Important Subsets for SSL

We start by investigating which properties the subset S^* must satisfy, such that the learned representations on the subset provide small downstream classification error. To do so, we rely on recent theoretical results on optimization and generalization of contrastive learning. In particular, the recent results of Huang et al. (2021) showed that the generalization performance of representations obtained with contrastive learning dependents on: (1) alignment of positive pairs, (2) divergence of class centers and (3) concentration of the augmented data. Alignment captures the similarity between representations of augmented views of examples, in expectation. Good alignment requires all augmented views of an example to have similar representations. Divergence of class centers captures how distant class centers $\boldsymbol{\mu}_l$ and $\boldsymbol{\mu}_k$ are. Good divergence results in large enough distance between all pairs of class centers, i.e., small $\boldsymbol{\mu}_l^T\boldsymbol{\mu}_k\ \forall l,k\in[K]$. Concentration of augmented data is determined by the data distribution and augmentation pipeline. Specifically, let $V_k^0\subseteq V_k$ be the subset of examples in every class $k\in K$ that share at least one very similar augmented view: $\sup_{\boldsymbol{x}_1,\boldsymbol{x}_2\in V_k}\min_{\boldsymbol{x}_1'\in A(\boldsymbol{x}_1),\boldsymbol{x}_2'\in A(\boldsymbol{x}_2)}\|\boldsymbol{x}_1-\boldsymbol{x}_2\|\leq\delta$ for small $\delta>0$. If for every latent class $k\in[K]$, V_k^0 is large enough ($|V_k^0|\geq\sigma|V_k|$ for large $\sigma\in(0,1]$), then the classes have sharp concentration of augmented data. In this case, good alignment and divergence guarantee good generalization performance for the downstream NN classifier.

While concentration of the augmentations is independent of the contrastive loss, minimizing the contrastive loss effectively aligns the augmented views of the examples and results in good divergence of the class centers. Formally, for a normalized encoder $\|f\|=1$, the InfoNCE loss in Eq. (1) can be written as:

$$\mathcal{L}_{cl}(V)=\frac{1}{2}\underbrace{[\mathop{\mathbb{E}}_{i\in V}\mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)}\|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\|^2]}_{\mathcal{L}_{align}(V)\text{: Related to Alignment}}-1 \tag{6}$$
$$+\mathop{\mathbb{E}}_{i,j\in V}\mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)\\ \boldsymbol{x}^-\in A(\boldsymbol{x}_j)}}\log\left(e^{f(\boldsymbol{x}_1)^T f(\boldsymbol{x}_2)}+\underbrace{e^{f(\boldsymbol{x}_1)^T f(\boldsymbol{x}^-)}}_{\text{Related to Divergence}}\right).$$

The first term in the RHS of Eq. (6) is closely related to the alignment and the second term in the RHS is related to the divergence of class centers.

Alignment. Minimizing the first term in the RHS of Eq. (6) aligns augmented views of the training examples in expectation, and results in a small probability $R_\epsilon(V)$ for examples to still have non-aligned augmented views, i.e, the largest distance between their augmented views is larger than ϵ:

$$R_\epsilon(V)=\mathbb{P}\Big[i\in V:\sup_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)}\|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\|>\epsilon\Big]. \tag{7}$$

In particular, for a L-Lipschitz continuous encoder f, we

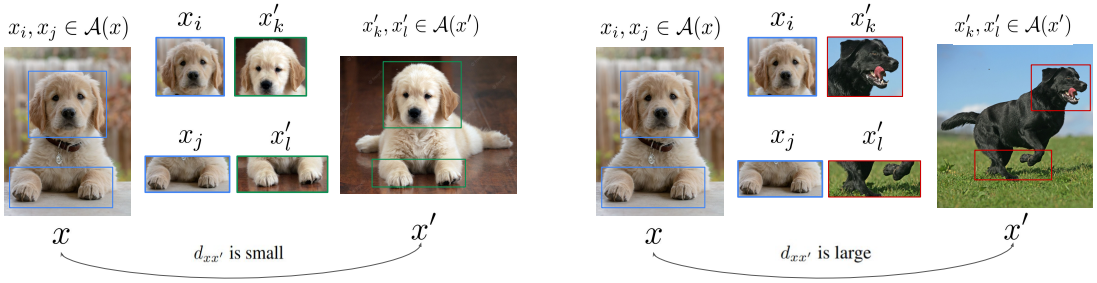


Figure 1. Visualizing Expected Augmentation Distance $d_{x,x'}$. Pair of examples on left shows two examples that are semantically very similar as seen by their augmentations being very similar to each other, thus the expected augmentation distance between them is small. In contrast, pair of examples on the right are not as semantically similar, thus have augmentations that are very dissimilar to each other.

have that (Huang et al., 2021):

$$R_\epsilon(V) \leq \eta(\epsilon) \cdot \sqrt{\mathcal{L}_{align}(V)}, \tag{8}$$

where $\mathcal{L}_{align}(V) = \mathbb{E}_{i \in V} \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\|^2$ is the alignment loss; $\eta(\epsilon) = \mathcal{O}(\frac{1}{\epsilon})$ is a function of ϵ and the transformations used for data augmentations.

Divergence. Minimizing the second term in RHS of Eq. (6) pushes away the class centers, i.e., expected representation of examples in a class, and yields a small $\boldsymbol{\mu}_k^T \boldsymbol{\mu}_l$ for all $k, l \in [K]$. Effectively, it maximizes the distance between different class representations.

Minimizing the InfoNCE loss in Eq. (6) minimizes both terms in the RHS, thus ensuring good alignment and divergence. With good alignment (small R_ϵ) and good divergence (small $\boldsymbol{\mu}_k^T \boldsymbol{\mu}_l$), the NN classifier g_f can correctly classify all the examples in the main part of every class that have concentrated and aligned augmented views. If the majority of examples in every class have a high concentration of augmented data is large (large σ), good generalization is guaranteed. Formally,

Theorem 4.1 (Huang et al. 2021)**.** *For any $l, k \in [K]$, if*

$$\boldsymbol{\mu}_k{}^T \boldsymbol{\mu}_l < \phi(\sigma, \delta, \epsilon), \tag{9}$$

then the downstream error rate of NN classifier is

$$\xi(g_f(V)) \leq (1 - \sigma) + R_\epsilon(V). \tag{10}$$

Exact form of $\phi(\sigma, \delta, \epsilon)$ is discussed in Appendix B.

4.1. Subsets that Preserve Alignment and Divergence

We rely on the above observations to find a subset that, when used to learn representations, provides similar generalization performance to that of the full data, for the downstream NN classifier. The key idea of our approach is to find a subset, such that minimizing the contrastive loss on this subset: (1) results in good alignment for all the examples, and (2) preserves the class centers of full data. In doing so, we ensure that the divergence of the class centers is preserved. If such a subset can be found, minimizing the contrastive loss in Eq. (1) on the subset results in good alignment and divergence on the full data, hence guarantees similar generalization performance for the downstream NN classifier.

Next, we introduce the notion of expected augmentation distance and discuss how it can be leveraged to find a subset that satisfies the above two conditions.

We start by defining the expected augmentation distance:

Definition 4.2 (**Expected augmentation distance**)**.** We define the expected augmentation distance between examples $i, j \in V$ as the expected l_2 norm between all pairs $(\boldsymbol{x}, \boldsymbol{x}')$ of augmented examples, such that $\boldsymbol{x} \in A(\boldsymbol{x}_i)$ and $\boldsymbol{x}' \in A(\boldsymbol{x}_j)$. Formally, for every pair of examples $i, j \in V$ we have:

$$d_{i,j} = \mathop{\mathbb{E}}_{\boldsymbol{x} \in A(\boldsymbol{x}_i), \boldsymbol{x}' \in A(\boldsymbol{x}_j)} \|\boldsymbol{x} - \boldsymbol{x}'\|. \tag{11}$$

Intuitively, expected augmentation distance captures the *semantic dissimilarity* between every pair of examples. That is, two examples that are semantically similar have a small expected augmentation distance. We visualize examples with small and large expected augmentation distance in Fig. 1.

4.2. Ensuring Good Alignment

First, we address finding a subset that, when used to minimize the contrastive loss, aligns the augmented views of all the training examples. From Eq. (8), we know that minimizing the alignment loss $\mathcal{L}_{align}$, directly minimizes the probability $R_\epsilon(V)$ of examples with non-aligned augmented views. That is $R_\epsilon(V) \leq \eta(\epsilon) \cdot \sqrt{\mathcal{L}_{align}(V)}$.

Here, we find a subset $S_k \subseteq V_k$ of examples from every latent class k that ensures small $R_\epsilon(V_k)$, i.e., probability that examples in V_k are not well-aligned. For every (arbitrary) subset $S_k \subseteq V_k$ of size $r_k = |S_k|$ selected from class k with $n_k = |V_k|$ examples, we can upper-bound the probability $R_\epsilon(V_k)$ based on the alignment loss of the subset i.e. $\mathcal{L}_{align}(S_k)$. In particular, using $R_\epsilon(V_k) \leq \eta(\epsilon) \cdot \mathbb{E}_{i \in V_k} \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq$

$\eta(\epsilon)\sqrt{\mathcal{L}_{align}(V)}$ (Huang et al., 2021), we can write:

$$
\begin{aligned}
&R_\epsilon(V_k) \\
&\leq \eta(\epsilon) \cdot \mathop{\mathbb{E}}_{i\in V_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\| \qquad (12)\\
&= \frac{\eta(\epsilon)}{n_k} \cdot \Bigg(\sum_{i\in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\| \\
&\qquad + \sum_{i\in V_k\setminus S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\| \Bigg) \qquad (13)\\
&\leq \frac{\eta(\epsilon)}{n_k} \cdot \Bigg(\sum_{i\in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\| \\
&\qquad + \sum_{i\in V_k\setminus S_k} [2 \min_{j\in S_k} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1\in A(\boldsymbol{x}_i),\\ \boldsymbol{x}_2\in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_1)-f(\boldsymbol{x}_2)\|] \Bigg), \qquad (14)
\end{aligned}
$$

Detailed steps of getting Eq. (14) from Eq. (13) can be found in the Appendix C. Note that the first term in Eq. (14) is exactly $\frac{\eta(\epsilon)}{n_k}\sqrt{\mathcal{L}_{align}(S_k)}$. Hence, for a L-Lipschitz continuous encoder f, where $\|f(\boldsymbol{x})-f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x}-\boldsymbol{x}'\|\ \forall \boldsymbol{x},\boldsymbol{x}'$, we have:

$$
R_\epsilon(V_k) \leq \frac{\eta(\epsilon)}{n_k}\left(r_k\sqrt{\mathcal{L}_{align}(S_k)} + 2L \sum_{i\in V_k\setminus S_k^k} \min_{j\in S_k} d_{i,j} \right).
$$

The alignment loss on the subset $\mathcal{L}_{align}(S_k)$ can be effectively minimized by contrastive learning on the subset using the InfoNCE loss. We also empirically show in Appendix A Fig. 8(a) that alignment loss on the subsets we find for contrastive learning is smaller than the alignment loss on the full data, i.e., $\mathcal{L}_{align}(S_k) \leq \mathcal{L}_{align}(V_k)$. Therefore, training on a subset $S_k \subseteq V_k$ introduces at most the following error on $R_\epsilon(V_k)$, i.e., the probability for any example in V_k to have a distance larger than ϵ between its augmented views:

$$
\nu_R^k \leq \frac{2L\eta(\epsilon)}{n_k} \sum_{i\in V_k\setminus S_k^k} \min_{j\in S_k} d_{i,j}, \qquad (15)
$$

Therefore, the subset $S_k \subseteq V_k$ with *smallest expected augmentation distance* $d_{i,j}$ (semantic similarity) to the *rest* of the examples in the class $V_k \setminus S_k$ can best align augmentations of all the examples in the class V_k.

Remark. Eq. (15) shows that the subset S_k that aligns augmented views of all the examples in a class V_k should have an element that is sufficiently similar to any other example in the class. In other words, the subset should contain examples that are *representative of every group* of examples in the class.

4.3. Preserving the Class Centers

Next, we discuss finding a subset that captures the center of every latent class $\boldsymbol{\mu}_k$.

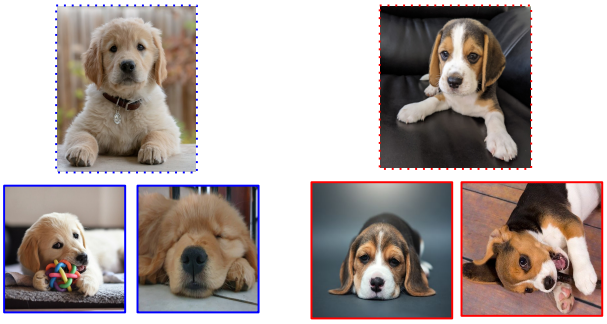


Figure 2. Most representative examples: examples in top row are each representative of their group (e.g. breed) in class *dog*.

For every (arbitrary) subset $S_k \subseteq V_k$ of size $r_k = |S_k|$ selected from class k with $n_k = |V_k|$ examples, we can write:

$$
\begin{aligned}
\boldsymbol{\mu}_k &= \mathop{\mathbb{E}}_{\substack{i\in V_k,\\ \boldsymbol{x}'\in A(\boldsymbol{x}_i)}} [f(\boldsymbol{x}')] \\
&= \mathop{\mathbb{E}}_{\substack{i\in V_k,\\ \boldsymbol{x}'\in A(\boldsymbol{x}_i)}} [f(\boldsymbol{x}')] - \mathop{\mathbb{E}}_{\substack{j\in S_k,\\ \boldsymbol{x}''\in A(\boldsymbol{x}_j)}} [f(\boldsymbol{x}'')] + \mathop{\mathbb{E}}_{\substack{j\in S_k,\\ \boldsymbol{x}''\in A(\boldsymbol{x}_j)}} [f(\boldsymbol{x}'')] \\
&= \frac{1}{n_k}\sum_{i\in V_k} \mathop{\mathbb{E}}_{\boldsymbol{x}'\in A(\boldsymbol{x}_i)} [f(\boldsymbol{x}')] - \frac{1}{r_k}\sum_{j\in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}''\in A(\boldsymbol{x}_j)} [f(\boldsymbol{x}'')] + \boldsymbol{\mu}_k^S \\
&= \frac{1}{n_k\cdot r_k}\Big[r_k \sum_{i\in V_k} \mathop{\mathbb{E}}_{\boldsymbol{x}'\in A(\boldsymbol{x}_i)} [f(\boldsymbol{x}')] - n_k \sum_{j\in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}''\in A(\boldsymbol{x}_j)} [f(\boldsymbol{x}'')] \Big] + \boldsymbol{\mu}_k^S \\
&= \frac{1}{n_k\cdot r_k} \sum_{i\in V_k}\sum_{j\in S_k} \Big[\mathop{\mathbb{E}}_{\boldsymbol{x}'\in A(\boldsymbol{x}_i)} [f(\boldsymbol{x}')] - \mathop{\mathbb{E}}_{\boldsymbol{x}''\in A(\boldsymbol{x}_j)} [f(\boldsymbol{x}'')] \Big] + \boldsymbol{\mu}_k^S \\
&= \frac{1}{n_k\cdot r_k} \sum_{i\in V_k}\sum_{j\in S_k} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}'\in A(\boldsymbol{x}_i),\\ \boldsymbol{x}''\in A(\boldsymbol{x}_j)}} [f(\boldsymbol{x}') - f(\boldsymbol{x}'')] + \boldsymbol{\mu}_k^S \qquad (16)
\end{aligned}
$$

Hence, for a L-Lipschitz continuous encoder f, where $\|f(\boldsymbol{x})-f(\boldsymbol{x}')\| \leq L\|\boldsymbol{x}-\boldsymbol{x}'\|\ \forall \boldsymbol{x},\boldsymbol{x}'$, we can upper-bound the normed difference between the center of class V_k and subset S_k as follows:

$$
\nu_\mu^k = \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^S\| \leq L \cdot \mathop{\mathbb{E}}_{\substack{i\in V_k,\\ j\in S_k}} [d_{i,j}]. \qquad (17)
$$

That is, the subset that preserves the center of class k, can be found by *minimizing the expectation of expected augmentation* distances (semantic similarity) between examples in the subset S_k and *all* the data points V_k in class k.

Remark. Eq. (17) implies that a subset S_k that captures the centre of class k, should be similar to all the examples in the class, in expectation. Such a subset contains examples from dense regions with sharp concentration of augmented data. Such examples *best represent the entire class*.

4.4. Minimizing the Alignment and Divergence Error

Based on Eq. (15) and (17), we find the subset that ensures alignment of all data points in class k and closely captures

the center of the class, by solving the following problem:

$$S_k^* = \underbrace{\arg\min_{S\subseteq V_k, |S|\leq r_k} \mathop{\mathbb{E}}_{\substack{i\in V_k,\\ j\in S_k}} [d_{i,j}]}_{\text{Captures class center}} \quad \text{s.t.} \quad \underbrace{\min_{j\in S_k} d_{i,j} \leq \delta \;\; \forall i \in V_k}_{\text{Ensures alignment}} . \tag{18}$$

Problem (18) is NP-hard as it involves calculating the value of the objective over an exponential number of subsets. To efficiently find a subset that captures the class center and contains representatives from different groups of examples in a class, we rely on the following objective which minimizes the *sum* of expected augmentation distance between examples in the subset $j \in S_k$ and the rest of examples in the class $V_k \setminus S_k$:

$$S_k^* = \arg\min_{S\subseteq V_k, |S|\leq r_k} \sum_{i\in V_k\setminus S_k} \sum_{j\in S_k} d_{i,j}. \tag{19}$$

By minimizing the sum of distances (dissimilarities) between the subset and the *rest* of examples, Eq. (19) finds examples that are similar to many other examples in their class. In doing so, it finds a subset that ensure alignment. At the same time, the selected examples are selected from *dense* regions with sharp concentration of augmented data. Hence, the subset closely preserves the class center.

The above minimization problem can be turned into maximizing the following non-monotone submodular[1] problem:

$$S_k^* = \arg\max_{S\subseteq V_k, |S|\leq r_k} \sum_{i\in V_k\setminus S_k} \sum_{j\in S_k} C - d_{i,j}, \tag{20}$$

where C is a big constant. $C - d_{i,j}$ captures the *similarity* between i and j.

Thus, we can find a nearly optimal subset using algorithms designed for maximizing non-monotone submodular functions under a cardinality constraint. First, we rely on the greedy algorithm to identify a subset and then refine it by using unconstrained submodular maximization (Mirzasoleiman et al., 2016). The greedy algorithm commences with an empty set $S_0 = \emptyset$, and at each step t, it selects an element $e \in V$ that maximizes the marginal utility $F(e|S_t) = F(S_t \cup \{e\}) - F(S_t)$. Formally, $S_t = S_{t-1} \cup \{\arg\max_{e\in V} F(e|S_{t-1})\}$. For unconstrained maximization, we utilize the double-greedy algorithm (Buchbinder et al., 2015), which initializes $S^\alpha = \emptyset$ and $S^\beta = S_T$, where S_T is the subset found by the final iteration of the greedy algorithm. It then computes $a_e = F(e|S^\alpha)$ and $b_e = F(S^\beta \setminus \{e\})$ for all $e \in V$. Subsequently, it adds examples for which $a_e \geq b_e$ to S^α and removes examples for which $a_e < b_e$ from S^β, eventually setting $S^\alpha = S^\beta$. The time complexity of the greedy algorithm is $\mathcal{O}(nk)$ to identify k out of n examples, which can be further expedited through lazy evaluation (Minoux, 2005). The double-greedy approach applied to the subset has a complexity of $\mathcal{O}(k)$. Thus, the subset can be found very efficiently.

[1] A set function $F : 2^V \to \mathbb{R}^+$ is *submodular* if $F(e|S) = F(S\cup\{e\}) - F(S) \geq F(T\cup\{e\}) - F(T)$, for any $S \subseteq T \subseteq V$ and $e \in V \setminus T$.

Algorithm 1 *SAS*: finding **S**ubsets that maximize the expected **A**ugmentation **S**imilarity

Input: Subset size B, proxy model f_p
Output: Subset S
$\{V_1, ..., V_K\} \leftarrow$ approximate latent classes (Sec. 4.5)
for all $V_k \in \{V_1, \cdots, V_K\}$ **do**
  for all $i, j \in V_k$ **do**
    $s_{i,j} = \langle f_p(\boldsymbol{x}_i), f_p(\boldsymbol{x}_j)\rangle$
  end for
  $S_k \leftarrow \{\}$
  $r_k \leftarrow \frac{|V_k|}{|V|} \cdot B$
  $F(S_k) = \sum_{i\in V_k\setminus S_k} \sum_{j\in S_k} s_{i,j}$
  while $|S_k| \leq r_k$ **do**
    $e \leftarrow \text{argmax}_{e\in V_k\setminus S_k} F(e|S_k)$
    $S_k \leftarrow S_k \cup \{e\}$
  end while
  $S_k \leftarrow$ double-greedy(S_k)
end for
return $S = \{S_1 \cup \cdots \cup S_K\}$

Remark. Intuitively, as the subsets selected from different classes have a small expected augmentation distance to all the different groups in the class, they pull together all the examples in a class during contrastive learning and let the representations of a class to cluster closely. At the same time, as they preserve the class centers, they allow the representations of different classes to be effectively pushed away from each other. In doing so, the subsets effectively minimize the contrastive loss on the full data. Note that as $d_{i,j}$ is a property of the data in the *input space*, the subset found by solving Problem (20) ensures good alignment and divergence *throughout contrastive learning*.

Fig. 2 presents a visualization of examples in the dog class. The examples found by Eq. (20) resemble those in the top row, i.e., they contain the core features of the class (e.g. the head and the paws of the puppies) with minimal noise (e.g. the non-standard poses of the puppies in the bottom row). Due to the standard and clear presentation of the core features of their respective groups, the examples in top row have smaller expected augmentation distance to many examples than examples in the bottom row, where some core features may be occluded (e.g. paws not visible) and/or presented in non-standard ways (e.g. open mouth).
Next, we provide a generalization guarantee for contrastive learning from the subset. The following theorem shows that if contrastive learning on the subset provides a small extra divergence on the center of examples selected from different classes compared to that of full data, the downstream NN

classifier will have a similar generalization performance to that of contrastive learning from full data.

Theorem 4.3. *Assume f is a normalized encoder and the subset S_k selected by Eq. (20) has ν_R^k error (Eq. 15) in capturing $R_\epsilon(f, V_k)$ and ν_μ^k error (Eq. 17) in capturing the center of class k. If for any pair of classes $k, l \in [K]$, we have:*

$$\boldsymbol{\mu}_k^{S^T} \boldsymbol{\mu}_l^S < \phi(\sigma, \delta, \epsilon) - \big(C\nu_R^k + 2(\max\{\nu_\mu^k, \nu_\mu^l\})^2 + 4\max\{\nu_\mu^k, \nu_\mu^l\}\big). \tag{21}$$

where $\phi(\sigma, \delta, \epsilon)$ is the requirement on divergence of full data class centers in Theorem 4.1 and C is a constant, then the generalization error of the model trained on the subset can be bounded by:

$$\xi(g_{f^S}(V)) \leq (1 - \sigma) + R_\epsilon + \nu_R. \tag{22}$$

Theorem 4.3 shows that if the subset captures the class centers and alignment closely (i.e. ν_R and ν_μ are small), then minimizing the contrastive loss on the subset provides a similar divergence to that of full data, and thus a similar downstream generalization performance for the NN classifier is guaranteed.

The proof can be found in Appendix B, where we also discuss that $C\nu_R$ is generally small. Fig. 8(b) in Appendix A confirms that divergence of *full data* class centers when training on sufficiently large subsets found by Eq. (20) is in fact better than that of training on the full data. This explains the similar or even superior generalization performance of models trained on *SAS* subsets to models trained on the full data.

4.5. SAS: Finding the Subset in Practice

Finally, we present our method, *SAS*, which finds subsets that minimize expected augmentation distance or equivalently maximize expected augmentation similarity, by approximately finding the latent classes and estimating the expected augmentation distance, without having the labels.

Approximately Finding the Latent Classes. Problem (20) requires selecting a subset from every class separately. Without the labels, we need to approximately find the latent classes. In practice, one can find latent classes by clustering the representations of a model trained with contrastive SSL. This approach requires no extra information and thus can generalize to contrastive learning in all domains. However, if an extra small subset of labeled data and a proxy model is available, we can find latent classes much more accurately. Specifically, if a small subset of labels are available, a proxy model can be used to approximately find the latent classes. In our experiments, we show that having as small as 1% of the labels, the pretrained CLIP (Radford et al., 2021) image encoder can be used to find the latent classes more accurately. Crucially, even without having access to any downstream labels, the pretrained CLIP can be used to find the latent classes. In our experiments, we show that using CLIP's image and text encoders, we can match image embeddings from STL10 to the closest text embeddings from ImageNet labels to obtain approximate latent classes for STL10. In practice, any *fine-grained* relevant set of labels provide a superior performance. This is because linearly separable representations for the fine-grained task will ensure linearly separable representations for the coarser-grained task. This is a practical way to use *SAS* for vision tasks as well as other domains with pretrained foundational models.

Estimating the Expected Augmentation Similarity. Expected augmentation distance captures similarity of examples in the input space. However, as pixel space is extremely high-dimensional, nearly all expected augmentation distances will be very large and extremely sensitive to small noise. Instead, using a proxy model can better capture the semantic similarities in practice. Note that the proxy model does not necessarily have to be the same as the model being trained with SSL. Indeed, the proxy model can be much smaller than the model being trained or can be partially trained with similar augmentations, as we confirm experimentally. Having a proxy model f_p, for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in V_k$, we estimate *expected augmentation similarity*, i.e., $C - d_{i,j}$ in Eq. (20) by $s_{i,j} = \langle f_p(\boldsymbol{x}_i), f_p(\boldsymbol{x}_j)\rangle$. The pseudocode of *SAS* is illustrated in Alg. 1.

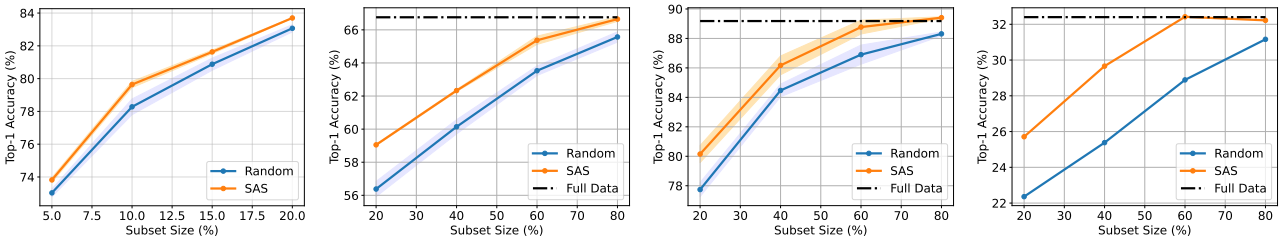
5. Experiments

In this section, we first evaluate the downstream generalization performance of the models trained by contrastive learning on the subsets found by *SAS* vs. random subsets, on CIFAR10, CIFAR100 (Krizhevsky & Hinton, 2009), STL10 (Coates et al., 2011b) and TinyImageNet (Deng et al., 2009). Then, we do an extensive ablation study on the effect of the approximate latent classes, and the proxy model used to estimate expected augmentation distance. Finally, we investigate the relation of these subsets to subsets that are important for supervised learning.

Training Setup We use SimCLR (Chen et al., 2020) as the contrastive learning method to train ResNet-50 (He et al., 2016) as the encoder architecture and a 2-layer MLP to project the representation to a 128-dimensional latent space. We use InfoNCE with temperature as our loss. Following the standard training pipeline in (Chuang et al., 2020; Robinson et al., 2020) we train for 400 epochs using the Adam optimizer with a learning rate of 0.001. Due to computational constraints, we use ResNet-18 as the encoder architecture for TinyImageNet and only have a single run per subset size. We also evaluate *SAS* on other contrastive learning methods, namely BYOL (Grill et al., 2020a), SimSiam (Chen & He, 2020) and MoCo (He et al., 2020), using ResNet-18 as the encoder architecture.

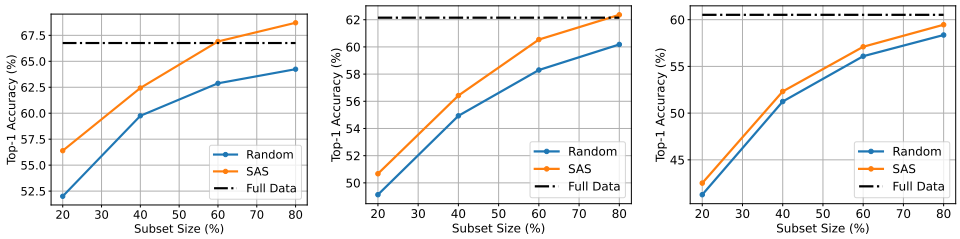
Data Augmentation For data augmentations, we use random crop, random resizing, random horizontal flips and color distortion, as is done in (Chen et al., 2020).

Evaluation. For evaluation, we use the widely used linear evaluation protocol (Chen et al., 2020; Chuang et al., 2020). That is, we train a linear classifier using the learned



(a) CIFAR10 (b) CIFAR100 (c) STL10 (d) TinyImageNet (ResNet-18)

Figure 3. Downstream Classification Accuracy of *SAS* Subsets vs. Random Subsets (reporting mean and std over 3 runs).



(a) BYOL (STL10) (b) SimSiam (CIFAR100) (c) MoCo (CIFAR100)

Figure 4. Evaluating *SAS* on other contrastive learning methods (training a ResNet-18).

representations of the training examples and their labels. Then, we evaluate the performance of the linear classifier on the test set representations and their corresponding labels. To ensure fair comparison, we compare *SAS* subsets with random subsets of the same size sampled from the same approximate latent classes.

5.1. Downstream Generalization Performance

First, we evaluate the downstream generalization performance of the model pre-trained on subsets of different sizes found by *SAS* vs. random subsets of the same size. Here, we use a pre-trained ResNet-50 as the proxy to calculate s_{ij}, as discussed in Sec. 4.5. For CIFAR100 and STL10, we consider all $s_{i,j} > 0$ and for CIFAR10 we consider all $s_{i,j} > 0.5$. As examples in CIFAR10 are generally more similar to each other, a larger threshold helps identifying representative examples better. To approximately find the latent classes, we train a linear classifier on the CLIP representations of the training data with a small randomly selected subset of training labels. In particular, for CIFAR10 and CIFAR100, we use 1% of the labels of training examples selected at random, and for STL10, we use all the labels ($< 5\%$) available labels. We use the trained linear classifiers to predict the latent class for all the training examples. In our ablation studies, we evaluate the performance when finding latent classes in other ways.

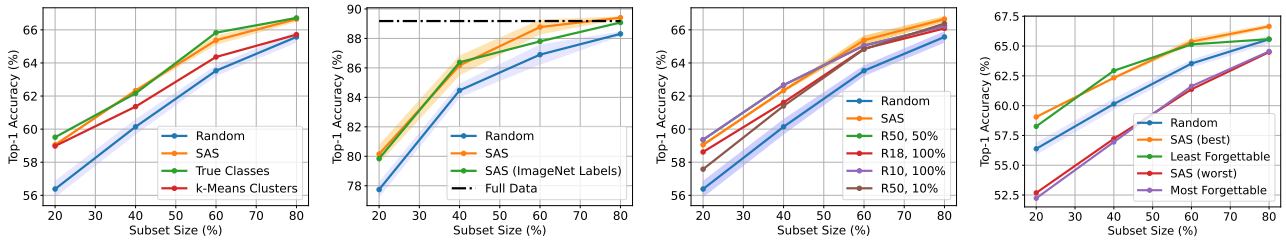
SimCLR Fig. 3 shows that training with SimCLR on subsets of various sizes found by *SAS* allows outperform random subsets by over 3% on CIFAR100 and STL10, and by up to 2% on CIFAR10. Critically, comparing the performance of the subsets with that of the full data, we can see that for CIFAR100, 20% of examples and for STL10 and TinyImageNet, 40% of examples, can be safely discarded without affecting downstream accuracy.

Other Contrastive Learning Methods. We validate that *SAS* can effectively find examples that contribute the most to contrastive learning across variety of contrastive learning methods. For these experiments, we train a ResNet-18 using BYOL (Grill et al., 2020b), MoCo (He et al., 2020) and SimSiam (Chen & He, 2020). Fig. 4(a) shows that training with BYOL on subsets of various sizes found by *SAS* from STL10 outperforms random subsets by more than 3%. Interestingly, with BYOL, subsets of size 80% outperform training on the full data by 2%. We also show that *SAS* allows us to discard 20% of examples on CIFAR100 when using SimSiam (Fig. 4(b)) and to achieve nearly 2% improvement over random subsets when using MoCo. (Fig. 4(c)).

5.2. Ablation Study

Next, we conduct an extensive ablation study on the effect of the approximate latent classes, and the proxy model used to estimate expected augmentation distance.

Finding Approximate Latent Classes. Fig. 5(a) compares the downstream performance on CIFAR100 when latent classes are obtained by training a linear classifier using 1% labeled training data on CLIP representations, to that of using the ground-truth class labels, and k-means clustering on the representations of a pretrained model. We see that approximately finding the latent classes using 1% of the labels works nearly as well as ground-truth labels. Notably, while the accuracy of the linear classifier trained with 1% of the labels of CIFAR100 is only 70.8%, this does not negatively affect the quality of subsets found by *SAS*. The latent classes help us avoid confusing examples that are



(a) Effect of approximate latent classes (CIFAR100) (b) ImageNet labels (STL10) (c) Effect of proxy model (CIFAR100) (d) Easy examples for SL are important for SSL(CIFAR100)

Figure 5. Ablation study on CIFAR100 and STL10.

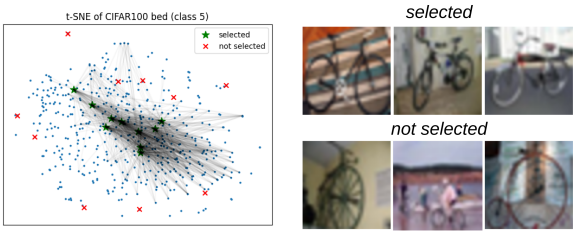
similar to examples across many latent classes; thus, even with relatively inaccurate latent classes, such examples can be filtered. Moreover, in the absence of any labels, using k-means clustering on the on the representations of a pretrained model performs equally well for smaller subsets and still provides a significant improvement for larger subsets.

Next, we consider using a different set of labels than the original labels of the training data to find the latent classes. In particular, we use a pretrained CLIP to label STL10 images by ImageNet labels, using the zero-shot approach. That is, we match every image in STL10 to one of the ImageNet labels, by finding the CLIP text embedding of the ImageNet label that is most similar to the CLIP image embedding. Fig. 5(b) compares the downstream performance on STL10, when using ImageNet labels to find latent classes using a zero-shot approach to that of using the available ($< 5\%$) STL10 labels to train a linear classifier on CLIP image representations. Notably, no label information about STL is used in the first case. The results clearly shows how *SAS* can entirely avoid the use of labels for approximating the latent classes. Crucially, any relevant and potentially finer-grained set of labels are enough to approximately find the latent classes and achieve a superior downstream performance.

Using Proxy Models to Estimate Expected Augmentation Distance. Fig. 5(c) shows estimating augmentation distance using various proxy models, such as a ResNet-50 that is partially trained for as few as 10% of epochs as well as smaller models such as a pre-trained ResNet-10, achieves a very similar downstream performance to that of using a fully pre-trained ResNet-50.

5.3. Investigating subsets found by *SAS*

Visualization. Fig. 6(a) use t-SNE to visualize examples that are selected by *SAS* vs those that are not selected, from the class "bed" in CIFAR100. Examples with small expected augmentation distance to selected and not selected examples are connected. We see that the selected examples have small distance to many other examples in the class. Fig. 6(b), illustrates some examples that are selected and not selected from the "bicycle" class. We see that the selected examples are representatives of the whole class, while those not selected present uncommon poses or views of the object.



(a) t-SNE of *bed* (pairs with small $d_{\boldsymbol{xx}'}$ are connected) (b) Examples from *bicycle*

Figure 6. Visualizing selected examples from CIFAR100

Easy Examples are the Most Important. Finally, we use the forgetting score (Toneva et al., 2019), i.e. the number of times an examples is misclassified after being correctly classified during *supervised* learning, to quantify the difficulty of an example. Importantly, least forgettable examples that can be safely discarded from supervised learning without harming the accuracy (Toneva et al., 2019). Fig. 5(d) shows that least forgettable examples can considerably outperform the random baseline and achieve a comparable performance to *SAS* for smaller subsets. On the other hand, the most forgettable examples that are most beneficial for supervised learning, perform significantly worse than the random baseline and similar to the subsets deemed worst by *SAS*. This illustrates how the subsets that contribute the most to contrastive learning are the least beneficial for supervised learning and vice-a-versa. Fig. 7 in Appendix A further shows that subsets found by *SAS* have low forgetting score and high confidence, in expectation. That is, they are easy for supervised learning. Effectively, the most important subsets for SSL are least important for supervised learning.

6. Conclusion

We identified subsets of examples that contribute the most to contrastive SSL. Theoretically, we characterized important subsets for contrastive learning with rigorous generalization guarantees for downstream performance. Empirically, we showed that using our method 20% - 40% examples can be discarded on CIFAR100, STL10 and TinyImageNet, observing no loss and even improvement in downstream accuracy. Surprisingly, we discovered that these important subsets are the least informative for supervised learning.

Acknowledgment. This research was supported by the National Science Foundation CAREER Award 2146492.

References

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Buchbinder, N., Feldman, M., Seffi, J., and Schwartz, R. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. February 2020. doi: 10.48550/arXiv.2002.05709. URL https://arxiv.org/abs/2002.05709v3.

Chen, X. and He, K. Exploring simple siamese representation learning, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased Contrastive Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8765–8775. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/63c3ddcc7b23daa1e42dc41f9a44a873-Abstract.html.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011a.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011b. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., and Zaharia, M. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020a.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised Learning, September 2020b. URL http://arxiv.org/abs/2006.07733. arXiv:2006.07733 [cs, stat].

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss, 2021. URL https://arxiv.org/abs/2106.04156.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning, March 2020. URL http://arxiv.org/abs/1911.05722. arXiv:1911.05722 [cs].

Huang, W., Yi, M., and Zhao, X. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021.

Katharopoulos, A. and Fleuret, F. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018.

Killamsetty, K., Durga, S., Ramakrishnan, G., De, A., and Iyer, R. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Le, Y. and Yang, X. S. Tiny imagenet visual recognition challenge. 2015.

Mindermann, S., Brauner, J. M., Razzak, M. T., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A. N., Morisot, A., Farquhar, S., and Gal, Y. Prioritized Training on Points that are Learnable, Worth Learning, and not yet Learnt. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 15630–15649. PMLR, June 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html. ISSN: 2640-3498.

Minoux, M. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, pp. 234–243. Springer, 2005.

Mirzasoleiman, B., Badanidiyuru, A., and Karbasi, A. Fast constrained submodular maximization: Personalized data summarization. In *International Conference on Machine Learning*, pp. 1358–1367. PMLR, 2016.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. Coresets for data-efficient training of machine learning models. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep Learning on a Data Diet: Finding Important Examples Early in Training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ac56f8fe9eea3e4a365f29f0f1957c55-Abstract.html.

Pooladzandi, O., Davini, D., and Mirzasoleiman, B. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pp. 17848–17869. PMLR, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive Learning with Hard Negative Samples. October 2020. doi: 10.48550/arXiv.2010.04592. URL https://arxiv.org/abs/2010.04592v2.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5628–5637. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/saunshi19a.html.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. S. Beyond neural scaling laws: beating power law scaling via data pruning, August 2022. URL http://arxiv.org/abs/2206.14486. arXiv:2206.14486 [cs, stat].

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746. URL https://aclanthology.org/2020.emnlp-main.746.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An Empirical Study of Example Forgetting During Deep Neural Network Learning, November 2019. URL http://arxiv.org/abs/1812.05159. arXiv:1812.05159 [cs, stat].

Tosh, C., Krishnamurthy, A., and Hsu, D. Contrastive estimation reveals topic posterior information to linear models. *J. Mach. Learn. Res.*, 22:281–1, 2021.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

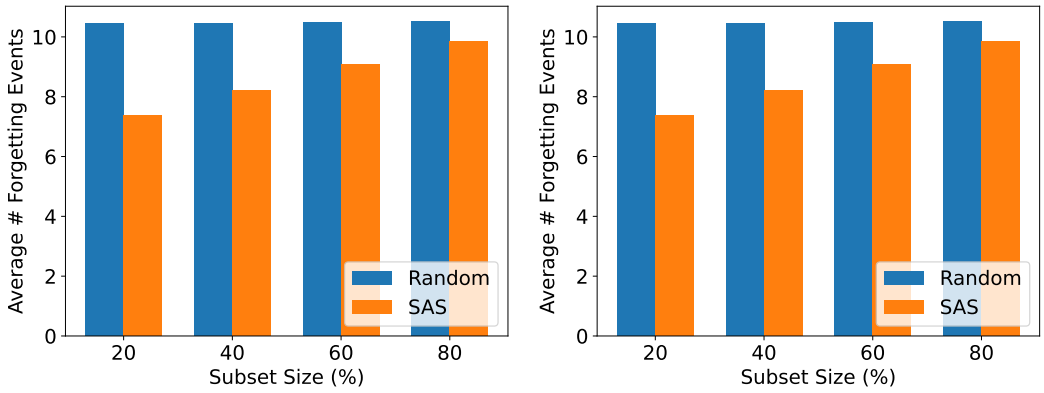
A. Extension to Experiments

A.1. Details for STL10 Investigatory Experiments

BYOL We consider a ResNet-18 trained for 40 epochs on STL10 with batch size 64 using SGD with learning rate of 0.001.

A.2. Easy Examples are Important

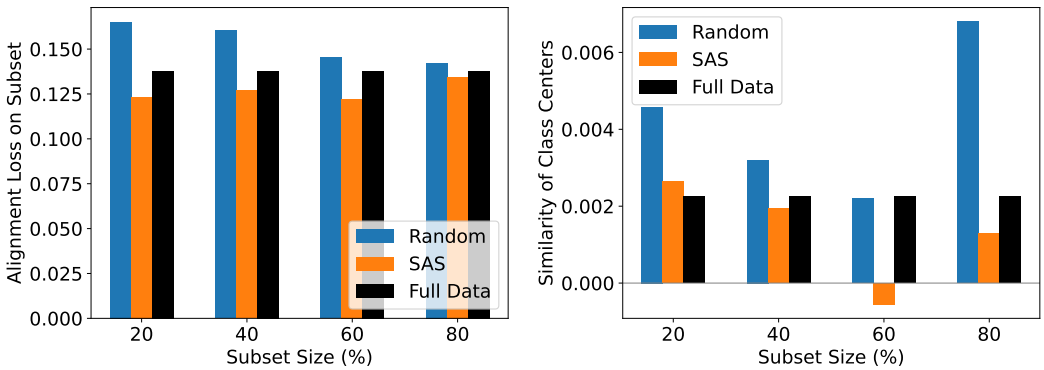
Here, we present results showing that the subsets *SAS* selects are easier for supervised learning by various metrics. We consider the number of forgetting events (Toneva et al., 2019) and the confidence of the prediction to quantify difficulty of a given example. Fig. 7 shows that *SAS* consistently picks examples with lower average forgetting events and higher confidence than the random subsets.



(a) Forgetting Score (b) Confidence

Figure 7. Examples found by *SAS* are easy (smaller number of forgetting events or higher confidence) for supervised learning.

A.3. Empirical Proof of Good Alignment and Divergence



(a) $\mathcal{L}_{align}(S)$ (b) Divergence

Figure 8. Empirically verifying we find subsets that achieve good alignment and divergence

In Fig. 8, we empirically measure $\mathcal{L}_{align}(S)$ and the mean similarity of class centers to show that the subsets chosen by *SAS* do indeed have better alignment and divergence than random subsets. Moreover, Fig. 8(a) also empirically verifies our claim that $\mathcal{L}_{align}(S_k) \leq \mathcal{L}_{align}(V_k)$

B. Proof for Theorem 4.3

Proof. First, we bound the discrepancy in divergence of subset class centers and divergence of full data class centers, relying on the discrepancy between class centers on the subset and full data.

Let $\nu_\mu^k = \|\boldsymbol{\mu}_k^S - \boldsymbol{\mu}_k\|$ and $\nu_\mu^l = \|\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_l\|$. Then,

$$\boldsymbol{\mu}_k^{S^T}\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_k^T\boldsymbol{\mu}_l = ((\boldsymbol{\mu}_k^S - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k)^T((\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_l) + \boldsymbol{\mu}_l) - \boldsymbol{\mu}_k^T\boldsymbol{\mu}_l \tag{23}$$

$$= (\boldsymbol{\mu}_k^S - \boldsymbol{\mu}_k)^T(\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_l) + (\boldsymbol{\mu}_k^S - \boldsymbol{\mu}_k)^T\boldsymbol{\mu}_l + \boldsymbol{\mu}_k^T(\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_l) + \boldsymbol{\mu}_k^T\boldsymbol{\mu}_l - \boldsymbol{\mu}_k^T\boldsymbol{\mu}_l \tag{24}$$

$$\leq \nu_\mu^k\nu_\mu^l + \nu_\mu^k\|\boldsymbol{\mu}_l\| + \nu_\mu^l\|\boldsymbol{\mu}_k\| \tag{25}$$

Thus, for a normalized encoder $\|f\| = r$ we get

$$\boldsymbol{\mu}_k^{S^T}\boldsymbol{\mu}_l^S - \boldsymbol{\mu}_k^T\boldsymbol{\mu}_l \leq r(\nu_\mu^k + \nu_\mu^l) + \nu_\mu^k\nu_\mu^l. \tag{26}$$

Next, we use Theorem B.1 to provide a generalization guarantee for the downstream NN classifier.

Let $V^\epsilon \subseteq V$ be the subset of examples of the full data that are well-aligned i.e. $\forall \boldsymbol{x}_i \in V^\epsilon$, s.t. $\sup_{\boldsymbol{x}_1,\boldsymbol{x}_2\in A(\boldsymbol{x}_i)}\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq \epsilon$

Recall $V_k^0 \subseteq V_k$ is the subset of examples with sharp concentration of augmented data in latent class k, i.e., $\sup_{i,j\in V_k^0}\min_{\boldsymbol{x}\in A(\boldsymbol{x}_j),\boldsymbol{x}'\in A(\boldsymbol{x}_j)}\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \delta$ and $|V_k^0| \geq \sigma|V_k|$ for $\sigma \in (0, 1]$

Theorem B.1 (Complete version of Theorem 4.1 (Huang et al., 2021))**.** *For any $l, k \in [K]$, if*

$$\boldsymbol{\mu}_k^T\boldsymbol{\mu}_l < \phi(\sigma, \delta, \epsilon) = r^2(1 - \rho_k(\sigma, \delta, \epsilon) - \sqrt{2\rho_k(\sigma, \delta, \epsilon)} - \frac{1}{2}\Delta_\mu), \tag{27}$$

then every example in $V_k^0 \cap V^\epsilon$ can be classified correctly by the NN classifier, where $\rho_k(\sigma, \epsilon, \delta) = 2(1 - \sigma) + \frac{R_\epsilon}{p_k} + (\sigma - \frac{R_\epsilon}{p_k})(\frac{L\delta}{r} + \frac{2\epsilon}{r})$, p_k = probability of an example being from latent class k and $\Delta_\mu = 1 - \min_k\|\boldsymbol{\mu}_k\|^2/r^2$.

If for any latent class $k \in [K]$, all examples in $V_k^0 \cap V^\epsilon$can be classified correctly by the NN classifier, then the downstream error rate of NN classifier

$$\xi(g_f(V)) \leq (1 - \sigma) + R_\epsilon(V) \tag{28}$$

The above Theorem cannot be directly used as the training on the subset introduces an additional error in capturing the alignment for latent class k, i.e., ν_R^k. Incorporating this, we get:

$$\boldsymbol{\mu}_k^T\boldsymbol{\mu}_l < r^2(1 - \rho_k'(\sigma, \delta, \epsilon) - \sqrt{2\rho_k'(\sigma, \delta, \epsilon)} - \frac{1}{2}\Delta_\mu), \tag{29}$$

where $\rho_k'(\sigma, \epsilon, \delta) = 2(1 - \sigma) + \frac{R_\epsilon + \nu_R^k}{p_k} + (\sigma - \frac{R_\epsilon + \nu_R^k}{p_k})(\frac{L\delta}{r} + \frac{2\epsilon}{r})$, and R_ϵ is the probability of examples not having aligned augmented views and ν_R^k is the alignment error on latent class k due to training on the subset.

From (26), we have:

$$\boldsymbol{\mu}_k^{S^T}\boldsymbol{\mu}_l^S + r(\nu_\mu^k + \nu_\mu^l) + \nu_\mu^k\nu_\mu^l < r^2\Big(1 - \rho_k'(\sigma, \epsilon, \delta) - \sqrt{2\rho_k'(\sigma, \epsilon, \delta)} - \frac{1}{2}\Delta_\mu\Big). \tag{30}$$

Then, as long as the following bound on the divergence of the class centers of the subset holds:

$$\boldsymbol{\mu}_k^{S^T}\boldsymbol{\mu}_l^S < r^2\Big(1 - \rho_k'(\sigma, \epsilon, \delta) - \sqrt{2\rho_k'(\sigma, \epsilon, \delta)} - \frac{1}{2}\Delta_\mu\Big) - r(\nu_\mu^k + \nu_\mu^l) - \nu_\mu^k\nu_\mu^l, \tag{31}$$

by Theorem B.1, we have that the NN classifier can correctly classify all the examples in $V_k^0 \cap V^\epsilon$ for any latent class $k \in [K]$

Thus, then incorporating our additional error in alignment ν_R into the generalization error bound in Theorem B.1, we get

$$\xi(g_{f^S}(V)) \leq (1-\sigma) + R_\epsilon(V) + \nu_R \tag{32}$$

□

Now, we can bound how much smaller the inner product of the class centers on the subset must be than that on the full data to achieve equivalent generalization guarantees (Eq. (32)), i.e. how much better the divergence on the subset should be than divergence on the full data.

Let $\varepsilon_{k,l} = r(\nu_\mu^k + \nu_\mu^l) + \nu_\mu^k \nu_\mu^l$. Then, comparing the bounds on divergence from Eq. (27) from Theorem B.1 and Eq. (31), we have

$$r^2(1-\rho_k(\sigma,\delta,\epsilon) - \sqrt{2\rho_k(\sigma,\delta,\epsilon)} - \frac{1}{2}\Delta_\mu) - r^2(1-\rho'_k(\sigma,\delta,\epsilon) - \sqrt{2\rho'_k(\sigma,\delta,\epsilon)} - \frac{1}{2}\Delta_\mu) + \varepsilon_{k,l} \tag{33}$$

$$= r^2\Big(\rho'_k(\sigma,\delta,\epsilon) - \rho_k(\sigma,\delta,\epsilon) + \sqrt{\rho'_k(\sigma,\delta,\epsilon)} - \sqrt{\rho_k(\sigma,\delta,\epsilon)})\Big) + \varepsilon_{k,l}. \tag{34}$$

Let $\zeta = \frac{\nu_R^k}{p_k}(1 - \frac{L\delta+2\epsilon}{r})$ where p_k is probability of an example being from latent class k.

Since $\sqrt{x+a} - \sqrt{x+b} \approx \frac{a-b}{2\sqrt{x}}$ for large x, we get:

$$\approx r^2\Big(\zeta + \frac{\zeta}{2\sqrt{\rho(\sigma,\delta,\epsilon)}}\Big) + {\nu_\mu^k}^2 + 2r\nu_\mu^k + r^2 + r(\nu_\mu^k + \nu_\mu^l) + \nu_\mu^k\nu_\mu^l \tag{35}$$

$$= C\nu_R^k + 2(\max\{\nu_\mu^k, \nu_\mu^l\})^2 + 4\max\{\nu_\mu^k, \nu_\mu^l\}. \tag{36}$$

where $C = \frac{r^2}{p_k}(1 - \frac{L\delta+2\epsilon}{r})(1 + \frac{1}{2\sqrt{\rho_k(\sigma,\delta,\epsilon)}})$.

Hence, we can rewrite Eq. (31) as

$${\boldsymbol{\mu}_k^S}^T \boldsymbol{\mu}_l^S < \phi(\sigma,\delta,\epsilon) - \big(C\nu_R^k + 2(\max\{\nu_\mu^k, \nu_\mu^l\})^2 + 4\max\{\nu_\mu^k, \nu_\mu^l\}\big) \tag{37}$$

When examples in every class have a high concentration of augmented data, i.e., when δ is small, $\rho(\sigma,\delta,\epsilon)$ is small and C is large. However, in this settings, picking a subset according the objective in Eq. (15) guarantees a very small ν_R^k. Therefore, $C\nu_R^k$ is small. On the other hand, when examples in every class do not have a high concentration of augmented data, δ is relatively large and hence C is small. As a result, $C\nu_R^k$ in Eq. (37) is small in both cases. Thus, for small ν_μ, the required divergence of subset class centers for the model trained on the subset is similar to the required divergence of full data class centers for the model trained on full data.

C. Detailed Steps to derive Eq. (14) from Eq. (13)

Let $j \in S_k$ and $\boldsymbol{x}_{j'} = \arg\min_{\boldsymbol{x}_{j_k} \in A(\boldsymbol{x}_j)} \mathbb{E}_{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_{j_k})\|$.

Then $\forall i \in V_k \setminus S_k$:

$$\mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \tag{38}$$

$$\leq \mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} [\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_{j'})\| + \|f(\boldsymbol{x}_{j'}) - f(\boldsymbol{x}_2)\|] \tag{39}$$

$$\leq \mathop{\mathbb{E}}_{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_{j'})\| + \mathop{\mathbb{E}}_{\boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_{j'}) - f(\boldsymbol{x}_2)\|. \tag{40}$$

But by definition of $\boldsymbol{x}_{j'}$, we have:

$$\leq \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i) \\ \boldsymbol{x}_{j_k} \in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_{j_k})\| + \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_2 \in A(\boldsymbol{x}_i) \\ \boldsymbol{x}_{j_k} \in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_{j_k}) - f(\boldsymbol{x}_2)\| \tag{41}$$

$$= 2 \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i) \\ \boldsymbol{x}_2 \in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| . \tag{42}$$

Since this inequality holds for any $j \in S$, we get:

$$\mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq 2 \min_{j \in S} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i) \\ \boldsymbol{x}_2 \in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| . \tag{43}$$

Thus, substituting the aforementioned bound to upper bound the second term (summation over $i \in V_k \setminus S_k$) Eq. (13), we get Eq. (14) i.e.:

$$\frac{\eta(\epsilon)}{n_k} \cdot \left(\sum_{i \in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| + \sum_{i \in V_k \setminus S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \right) \tag{44}$$

$$\leq \frac{\eta(\epsilon)}{n_k} \cdot \left(\sum_{i \in S_k} \mathop{\mathbb{E}}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \in A(\boldsymbol{x}_i)} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| + \sum_{i \in V_k \setminus S_k} [2 \min_{j \in S_k} \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}_1 \in A(\boldsymbol{x}_i), \\ \boldsymbol{x}_2 \in A(\boldsymbol{x}_j)}} \|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\|] \right). \tag{45}$$