DOES DECENTRALIZED LEARNING WITH NON-IID UNLABELED DATA BENEFIT FROM SELF SUPERVISION?

Lirui Wang, Kaiqing Zhang, Yunzhu Li, Yonglong Tian, Russ Tedrake
MIT CSAIL

ABSTRACT

The success of machine learning relies heavily on massive amounts of data, which are usually generated and stored across a range of diverse and distributed data sources. *Decentralized learning* has thus been advocated and widely deployed to make efficient use of the distributed datasets, with an extensive focus on supervised learning (SL) problems. Unfortunately, the majority of real-world data are *unlabeled* and can be highly *heterogeneous* across sources. In this work, we carefully study decentralized learning with unlabeled data through the lens of self-supervised learning (SSL), specifically contrastive visual representation learning. We study the effectiveness of a range of contrastive learning algorithms under decentralized learning setting, on relatively large-scale datasets including ImageNet-100, MS-COCO, and a new real-world robotic warehouse dataset. Our experiments show that the *decentralized* SSL (Dec-SSL) approach is *robust* to the heterogeneity of decentralized datasets, and learns useful representation for object classification, detection, and segmentation tasks, even when combined with the simple and standard decentralized learning algorithm of Federated Averaging (`FedAvg`). This robustness makes it possible to significantly reduce communication and to reduce the participation ratio of data sources with only minimal drops in performance. Interestingly, using the same amount of data, the representation learned by Dec-SSL can not only perform on par with that learned by centralized SSL which requires communication and excessive data storage costs, but also sometimes outperform representations extracted from decentralized SL which requires extra knowledge about the data labels. Finally, we provide theoretical insights into understanding why data heterogeneity is less of a concern for Dec-SSL objectives, and introduce feature alignment and clustering techniques to develop a new Dec-SSL algorithm that further improves the performance, in the face of highly non-IID data. Our study presents positive evidence to embrace *unlabeled data* in decentralized learning, and we hope to provide new insights into whether and why decentralized SSL is effective and/or even advantageous.[1]

1 INTRODUCTION

The success of machine learning hinges heavily on the access to large-scale and diverse datasets. In practice, most data are generated from different locations, devices, and embodied agents, and stored in a distributed fashion. Examples include a fleet of self-driving cars collecting a massive amount of streaming images under various road and weather conditions during everyday driving, or individuals using mobile devices to take photos of objects and scenery all over the world. Besides being large-scale, these datasets have two salient features: they are *heterogeneous* across data sources, and mostly *unlabeled*. For instance, images of road conditions, which are expensive to label, vary across cars driving on highways vs. rural areas, and under sunny vs. snowy weather conditions (Figure 11).

Methods that can make the best use of these large-scale distributed datasets can significantly advance the performance of current machine learning algorithms and systems. This has thus motivated a surge of research in *decentralized machine learning* (Konečnỳ et al., 2016; Hsieh et al., 2017; McMahan et al., 2017; Kairouz et al., 2021; Nedic, 2020), where usually a global model is trained on the distributed datasets using communication between the local data sources and a centralized server. The goal is typically to reduce or eliminate the exchanges of local raw data to save commu-

[1] Code is available at https://github.com/liruiw/Dec-SSL

nication costs and protect data privacy. How to mitigate the effect of *data heterogeneity* remains one of the most important research questions in this area (Zhao et al., 2018; Hsieh et al., 2020; Karimireddy et al., 2020; Ghosh et al., 2020; Li et al., 2021a), as it can heavily downgrade the performance of decentralized learning. Moreover, most existing decentralized learning studies focused on *supervised learning* (SL) problems that require data labels (McMahan et al., 2017; Jeong et al., 2020; Hsieh et al., 2020). Hence, it remains unclear whether and how decentralized learning can benefit from large-scale, heterogeneous, and unlabeled datasets typically encountered in the real world.

On the other hand, people have developed effective methods of learning purely from unlabeled data and demonstrated impressive results. Self-supervised learning (SSL), a technique that learns *representations* by generating supervision signals from the data itself, has unleashed the power of unlabeled data and achieved tremendous successes for a wide range of downstream tasks in computer vision (He et al., 2020; Chen et al., 2020; He et al., 2021b), natural language processing (Devlin et al., 2018; Sarzynska-Wawer et al., 2021), and embodied intelligence (Sermanet et al., 2018; Florence et al., 2018). These SSL algorithms, however, are usually trained in a *centralized* fashion by pooling all the unlabeled data together, without accounting for the heterogeneous nature of the decentralized data sources. Very recently, there have been a few contemporaneous/concurrent attempts (He et al., 2021a; Zhuang et al., 2021; 2022; Lu et al., 2022; Makhija et al., 2022) that bridged unsupervised/self-supervised learning and decentralized learning, with focuses on *designing better algorithms* that mitigate the data heterogeneity issue. In contrast, we revisit this new paradigm and ask the question:

Does decentralized learning with non-IID unlabeled data really benefit from SSL?

We focus on *understanding* the use of SSL in decentralized learning when handling unlabeled data. We aim to answer whether and when decentralized SSL (Dec-SSL) is effective (even combined with simple and off-the-shelf decentralized learning algorithms, e.g., `FedAvg` (McMahan et al., 2017)); what are the unique inherent properties of Dec-SSL compared to its SL counterpart; how do the properties play a role in decentralized learning, especially with highly heterogeneous data? We also aim to validate our observations on large-scale and practical decentralized datasets. We defer a more detailed comparison with these most related works to §A.

In this paper, we show that unlike in decentralized (supervised) learning, data heterogeneity can be *less concerning* in decentralized SSL, with both empirical and theoretical evidence. This leads to more communication-efficient and robust decentralized learning schemes, which can sometimes even outperform their supervised counterpart that assumes the availability of label information. Among the first studies to bridge decentralized learning and SSL, our study provides positive evidence to embrace unlabeled data in decentralized learning, and provides new insights into this setting. We detail our contributions as follows.

Contributions. (i) We show that decentralized SSL is a viable learning paradigm to handle relatively large-scale unlabeled datasets, even when combined with the simple `FedAvg` algorithm. Moreover, we also provide both experimental evidence and theoretical insights that decentralized SSL can be inherently *robust* to the data heterogeneity across different data sources. This allows more local updates, and can significantly improve the *communication efficiency* in decentralized learning. (ii) We provide further empirical and theoretical evidences that even when *labels* are available and decentralized supervised learning (and associated representation learning) is allowed, Dec-SSL still stands out in face of highly non-IID data. (iii) To further improve the performance of Dec-SSL, we design a new Dec-SSL algorithm, `FeatARC`, by using an iterative feature alignment and clustering procedure. Finally, we validate our hypothesis and algorithm in practical and large-scale data and task domains, including a new real-world robotic warehouse dataset.

2 Preliminaries and Overview

Consider a decentralized learning setting with K different data sources, which might correspond to different devices, machines, embodied agents, or datasets/users that can generate and store data locally. The goal is to collaboratively solve a learning problem, by exploiting the decentralized data from all data sources. More specifically, consider each data source $k \in [K]$ has local dataset $D_k = \{x_{k,i}\}_{i=1}^{|D_k|}$, and $x_{k,i} \in \mathcal{X} \subseteq \mathbb{R}^d$ are identically and independently distributed (IID) samples from probability distribution $\mathcal{D}_k$, i.e., $x_{k,i} \sim \mathcal{D}_k$. Note that the distributions $\mathcal{D}_k$ is in general different across data sources k, yielding an overall *heterogeneous* (i.e., non-IID) data distribution

for the data from all the sources. Let $D = \bigcup_{k\in[K]} D_k$ denote the set of all data samples. Moreover, we are interested in situations where no label is provided alongside the data x. To effectively utilize the large-scale *unlabeled* data, we resort to self-supervised learning approaches.

Specifically, SSL approaches extract representations from these unlabeled data, by finding an embedding function $f_w : \mathcal{X} \to \mathbb{R}^m$, where w is the parameter of the embedding function. $z = f_w(x)$ is the representation vector that can be useful for downstream tasks, e.g., classification or segmentation. We summarize several popular SSL approaches here that will be used later in the paper.

Self-supervised representation learning. Now consider a given data source $k \in [K]$. There are two popular methods in the SSL community. In contrastive learning (Chen et al., 2020; He et al., 2020) specifically, a sample x is used to provide supervision signals along with two generated *positive* samples x^+ and x (overloaded for notational simplicity) and (possibly multiple) *negative* samples x^- sampled from the training batch. The goal of SSL is to find an embedding f_w that makes x and x^+ close, while keeping x and x^-s apart, if negative samples are used.

One commonly used loss for SSL is the InfoNCE loss (Oord et al., 2018), which has been used in popular SSL approaches as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020):

$$\mathcal{L}_k(w) := \frac{1}{|D_k|}\sum_{i=1}^{|D_k|} -\log\left(\frac{\exp(-\mathbb{D}(f_w(x_{k,i}), f_w(x^+_{k,i}))/\tau)}{\exp(-\mathbb{D}(f_w(x_{k,i}), f_w(x^+_{k,i}))/\tau) + \sum_j \exp(-\mathbb{D}(f_w(x_{k,i}), f_w(x^-_{k,j}))/\tau)}\right) \tag{2.1}$$

where $\tau > 0$ is a temperature hyperparameter, j is the index for negative samples, $\mathbb{D}(\cdot,\cdot)$ is a distance function such as the cosine distance, i.e., $\mathbb{D}(z_1, z_2) = -\frac{z_1 \cdot z_2}{||z_1|| ||z_2||}$. Some other effective SSL approaches, such as BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021), remove the terms related to negative samples in (2.1). These methods also add an additional function g, the *feature predictor*, which only applies to x to create an asymmetry and to avoid the collapsed solutions. This usually leads to the following objective: $\mathcal{L}_k(w) := \frac{1}{|D_k|}\sum_{i=1}^{|D_k|} \mathbb{D}\big(g(f_w(x_{k,i})), f_w(x^+_{k,i})\big)$. In our experiments, we make use of both losses and the SSL approaches associated with them.

Decentralized SSL. To exploit the heterogeneous data distributed at different locations/devices, decentralized SSL optimizes the following global objective:

$$\min_w \sum_{k\in[K]} \frac{|D_k|}{|D|}\mathcal{L}_k(w), \tag{2.2}$$

which can be solved using many existing decentralized learning algorithms. For instance, `FedAvg` (McMahan et al., 2017) is one of the most representative, easy-to-implement, and communication-efficient decentralized learning algorithms which optimizes this objective without data-sharing among data sources. At each iteration t, the server first samples a set of data sources $\mathcal{M}_t$ with size $|\mathcal{M}_t| = \rho K$ and run δ local update steps on each of the local dataset. Then, each data source $k \in \mathcal{M}_t$ sends back the updated local model weight $w_k^{t,\delta}$ to the central server, and the server averages them to be the global model $w^{t+1} = \frac{1}{|\mathcal{M}_t|}\sum_{k\in\mathcal{M}_t} w_k^{t,\delta}$ for the next round $t+1$. The server then broadcasts the global model to each data source to reset $w_k^{t+1,0}$ as w^{t+1}. The number of local updates (δ) determines the communication efficiency (larger δ means less communication); in the experiments, we use E to denote the number of epochs of local updates (as a surrogate for δ). Both E and the participation rate ρ are important factors that determine the efficiency of decentralized learning. The learned representation $f_w(x)$ can then be used in downstream supervised learning tasks. There are many real-world applications of decentralized SSL, including self-driving cars, warehouse robots, and mobile devices. A further discussion can be found in Appendix §D.

2.1 Overview of Our Study

Terminology & setup. We separate our experiment pipeline into **representation learning** (pre-training phase) and **downstream evaluation** (evaluation phase). Our main focus is on the aforementioned **Dec-SSL** approach. We use `FedAvg` (McMahan et al., 2017) with SimCLR (Chen et al., 2020) as the default method. Moreover, we will also compare with settings where the *label information* is available, i.e., the classical decentralized (supervised) learning, which intuitively should be more favorable for learning. See Figure 1 for a summary of different settings. The first setting is **Dec-SL**: we simply run `FedAvg` on the decentralized labeled data, for end-to-end classification.Dec-SL does not learn *representations* explicitly, and serves as a natural baseline when labels are available.

The second setting is *representation learning* from Dec-SL, where we train supervised learning with `FedAvg`, and then use the feature extractor network as the backbone for downstream tasks. This way, we can also learn the representation from decentralized labeled data, and make the comparison with Dec-SSL more fair, since both are learning features for various downstream tasks. We term this setting as **Dec-SLRep**.

The evaluation phase tests the representations from Dec-SSL or Dec-SLRep. We consider two protocols in the evaluation phase: **linear probing** for image classification (Zhang et al., 2016) and **finetuning** for object detection/segmentation (Doersch et al., 2015). For classification, we train a linear classifier on top of the frozen pretrained network and evaluate the top-1 classification accuracy. For object detection/segmentation, we finetune the network by using the pretrained weights as initialization and training in an end-to-end fashion, and then we evaluate the mean Average Precision (mAP) metric. Downstream tasks are performed on centralized train and test dataset. Please refer to Appendix §C.1 for implementation details and Table 3 for experiment setups.

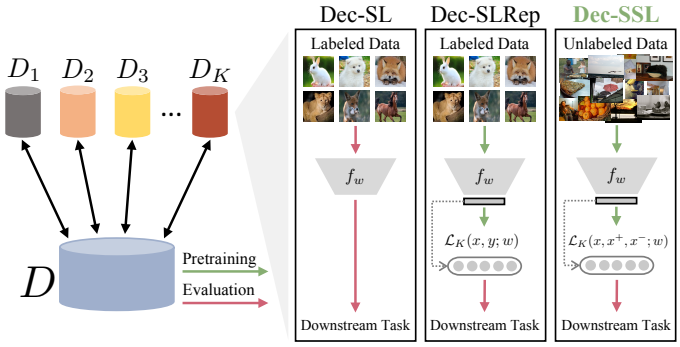

Figure 1: Comparisons among Dec-SL, Dec-SLRep, and Dec-SSL.

Questions of interest. Through extensive experiments on large-scale datasets, and theoretical analysis in simplified settings, we seek to answer the following questions: (i) How well can decentralized SSL, even instantiated with the simple `FedAvg` algorithm, rival the performance of its centralized counterpart, and handle the non-IIDness of decentralized unlabeled data? (ii) Is there any unique and inherent property of Dec-SSL, compared to its supervised learning counterpart; how and why may the property benefit decentralized learning, even when the label information is available? (iii) Is there a way to further improve the performance of Dec-SSL in face of highly non-IID data? Our hypothesis is that SSL, whose objective is not particularly dependent on the x to y mappings, learns a relatively *uniform* representation across decentralized and heterogeneous unlabeled datasets, thus leading to more efficient and robust decentralized learning schemes. We aim to validate this hypothesis and answer these questions in the following sections.

3 Dec-SSL is Efficient and Robust to Data Heterogeneity

We first seek to address question (i) in §2.1 – how well decentralized SSL performs, in face of non-IID and decentralized unlabeled data. To this end, we first introduce the notion of *data heterogeneity* in decentralized learning, which is usually categorized as *input heterogeneity, label distribution heterogeneity, and the heterogeneity in the relationships between the features and labels*, respectively (Hsieh et al., 2020). We create *label heterogeneity* by distributing each data source with different proportion of classes; we construct the heterogeneity via either sampling from a Dirichlet process with hyperparameter α or via skewness partitioning (Hsieh et al., 2020) with hyperparameter β. We also create *input heterogeneity* by leveraging the feature space of a pretrained network on the data. See §C.2 for more details on how we create data heterogeneity across data sources.

3.1 Experimental Observations

CIFAR classification under different types of non-IIDness. In this experiment, we construct input and label non-IIDness using 5 data sources in the CIFAR-10 (Krizhevsky et al., 2009) dataset based on the Dirichlet Process. The sources of non-IIDness are the feature clusters and labels, respectively. We control parameter α to create datasets from very IID (each data source has roughly a uniform distribution over 10 classes / 5 feature clusters) to very non-IID (each data source has data from 2 classes / 1 feature clusters). Recall that E denotes the number of epochs for local updates and ρ denotes the participation ratio of data sources at each round. We use $E = 50$ epochs of local updates in this experiment, which is equivalent to around $\delta = 1000$ iterations, i.e., each local data source updates 50 epochs independently before averaging. The results are shown in Figure 2. Surprisingly, the performance of downstream classification, with representations trained using decentralized SSL, is very *insensitive* to the non-IIDness across the datasets and only bears a slight performance drop.

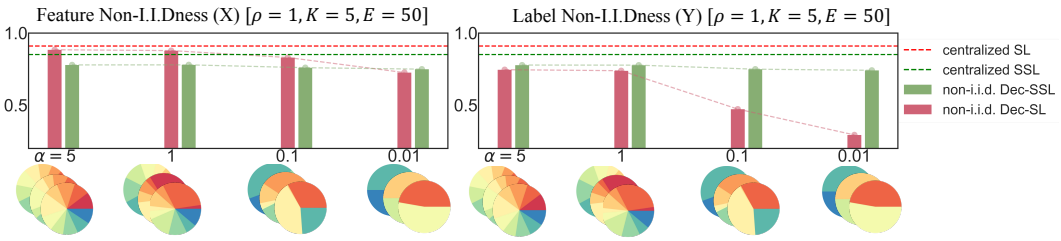

Figure 2: **SSL objective is robust to different types of X and Y heterogeneity on the CIFAR-10 dataset.** In the pie chart below, each pie denotes one data source, and color denotes the sample number of one source of non-IIDness (left to right, more non-IID). We observe that Dec-SSL is surprisingly robust to the non-IIDness in both input (X) and label (Y) and also behaves closer to its centralized counterpart. Y-axis denotes accuracy.

This robustness over data non-IIDness is encouraging, and stands in sharp contrast with most existing decentralized supervised learning algorithms, which are known to suffer from the data heterogeneity in general (Hsieh et al., 2020). As a baseline, we consider the classical decentralized SL approach of `FedAvg`, trained over the same non-IID data, but with label information. Indeed, the performance of decentralized SL can drop significantly as the non-IIDness increases. Finally, we note that the simple use of `FedAvg` in SSL can achieve performance comparable to the centralized SSL, showing that Dec-SSL is an effective decentralized learning scheme to handle unlabeled data.

Finetuning ImageNet representation for COCO detection. In this experiment, we finetune the representations learned from ImageNet to COCO detection benchmark (Lin et al., 2014) with the Detectron pipeline (Girshick et al., 2018). Specifically, we use ImageNet-100 with ResNet-18 and $1\times$ training schedule for Mask R-CNN (He et al., 2017) with a ResNet18 FPN being the backbone. Compared to the contemporary works (Zhuang et al., 2022; Lu et al., 2022) on federated self-supervised learning, our setup is more relevant to real-world applications, as it works on larger-scale and more practical datasets and tasks.

We run Dec-SSL on ImageNet-100 dataset with 5 data sources, and with $E = 1$ epoch of local updates, which corresponds to around $\delta = 500$ local updates, to learn the global representation using `FedAvg`. On Table 1 left, we observe that the representation from Dec-SSL almost reaches the performance of the representation from centralized SSL and improves upon baselines that train the model from scratch, i.e., the *no pretrain* row. This conveys that SSL can learn useful representations in decentralized settings, avoiding the heavy communication cost of centralized learning.

Decentralized SSL for real-world package segmentation. The issue of data heterogeneity and communication efficiency is significant for real-world applications such as those in Amazon warehouses, whose fleets of working robots can generate millions of images per day (see Figure 13 for an illustration). We provide details about the Amazon dataset in §D.1. We use data from one sample warehouse site at Amazon, and split the data based on the session ID (which is usually a sequence of days). Each decentralized learner is only allowed to access the local data at one session, which is equivalent to the non-IID case where skewness $\beta = 0$. We then deploy decentralized self-supervised learning on a subset of the enormous warehouse data, which has around 80000 images with contour labels output by the Amazon work-cells. We use SimCLR with `FedAvg` and communication efficiency $E = 1$ number of local update epochs, as the pretraining method.

On the right subtable of Table 1, we compare different ways to initialize weights for finetuning, and show that the representations learned from decentralized SSL outperforms training from scratch and even matches centralized SSL on the Amazon dataset. We also experiment with finetuning segmentation task using Mask R-CNN on different fractions of the data, and show that Dec-SSL can further improve the performance of training from scratch, when there is no as much labeled data.

3.2 Theoretical Insights

We now provide some theoretical insights into why the objective of Dec-SSL leads to more robust performance in face of data heterogeneity. In particular, we analyze the property of the solutions to the local and global objectives of Dec-SSL in a simplified setting, and show that the global objective is not affected significantly by the heterogeneity of local datasets. Our setup is inspired by the very recent work (Liu et al., 2021), where the effect of imbalanced data in centralized SSL was studied. In particular, we generalize the centralized and 3-way classification setting to a decentralized and

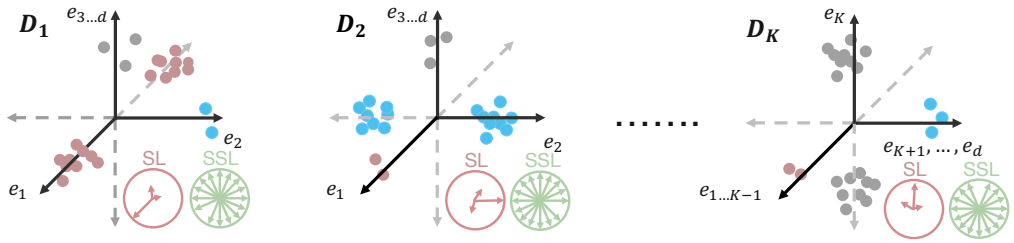

Figure 3: **The learned feature space of SSL is more insensitive to heterogeneity under the linear settings.** In §3.2, we consider a decentralized learning setting where each local dataset has a skewed distribution with most data points (each color is a class) concentrated on one axis. Each basis vector inside the sphere denotes how well it is represented in the learned subspace. For contrastive objectives, the learned feature space (green sphere) of the local model is more uniform and close to the global model. On the other hand, the SL objective (red sphere) tends to overfit to local dataset, and the learned feature spaces become heterogeneous.

ImageNet-100 Pretrain	MS-COCO AP^{bb}	MS-COCO AP^{mk}
no pretrain	20.5	19.4
Central-SLRep	21.2 (+0.7)	20.1 (+0.7)
Central-SSL	23.2 (+2.7)	22.1 (+2.7)
Dec-SLRep	19.8 (−0.7)	19.7 (+0.3)
Dec-SSL	22.1 (+1.6)	20.7 (+1.3)

Amazon Pretrain	Amazon (AP^{mk}) 100%	10%	1%
no pretrain	60.8	59.2	47.0
Central-SSL	61.6 (+0.8)	60.4 (+1.2)	49.5 (+2.5)
Dec-SSL	61.2 (+0.4)	60.1 (+0.9)	48.8 (+1.8)

Table 1: **Left). Object detection and semantic segmentation finetuned on COCO.** The model is pretrained on ImageNet-100 (Tian et al., 2020a) dataset and then finetune on MS-COCO with metrics bounding-box mAP (AP^{bb}) and mask mAP (AP^{mk}). **Right). Finetuning results on the Amazon package segmentation dataset with representations pretrained on the Amazon dataset.** We observe that Dec-SSL reaches similar performance (AP^{mk}) as centralized SSL and also outperforms training from scratch. Note that $100\%, 10\%, 1\%$ denote the portion of the data used for finetuning.

$2K$-way one, carefully design the generation of data distribution across data sources, and establish analyses for both local and global objectives in decentralized SSL. We also design new metrics to characterize the performance adapted to the decentralized setting. Due to space limitation, we include an abridged introduction here, and defer more details to Appendix §E.

Setup. Consider a Dec-SSL problem with K data sources. Similar to the SimSiam approach, we first augment x, an anchor sample from the dataset, by sampling $\xi, \xi' \sim \mathcal{N}(0, I)$ IID from the Gaussian distribution. Consider the linear embedding function $f_w(x) = wx$, where $w \in \mathbb{R}^{m \times d}$ and $m \geq 2K$. The SSL objective for data source k is given by

$$\mathcal{L}_k(w) := -\widehat{\mathbb{E}}\big[(w(x_{k,i} + \xi_{k,i}))^\top (w(x_{k,i} + \xi'_{k,i}))\big] + \frac{1}{2}\|w^\top w\|_F^2, \tag{3.1}$$

where $\widehat{\mathbb{E}}$ is taken expectation over the empirical dataset $x_{k,i} \sim D_k$, and the randomness of $\xi_{k,i}$ and $\xi'_{k,i}$. Moreover, recall the global objective is given in (2.2). Note that (3.1) instantiates SimSiam loss with the negative inner-product $\langle a, b \rangle$ as the distance function $\mathbb{D}(a, b)$ and no feature predictor, and with a regularization term for mathematical tractability, as in Liu et al. (2021).

Data heterogeneity. The K data sources collaboratively solve (2.2) to learn a representation for a $2K$-way classification task. The K local datasets are generated in a way that for each fixed $k \in [K]$, the labels are skewed in that data from classes $2k - 1$ and $2k$ constitute the majority of the data, while other classes are rare, or even unseen. More details on the specifications of data heterogeneity can be found in §E.1. We visualize the heterogeneity of the data distributions in Figure 3.

To compare the representations learned across data sources and that learned from jointly solving (2.2), we introduce the following definition on the representability of the representation space.

Definition 3.1 (Representability vector)**.** Let $\mathcal{S} \subseteq \mathbb{R}^d$ be the subspace spanned by the rows of the learned feature matrix $w \in \mathbb{R}^{m \times d}$, where the embedding function $f_w(x) = wx$. The *representability* of $\mathcal{S}$ is defined as a vector $\boldsymbol{r} = [r_1, \cdots, r_d]^\top \in \mathbb{R}^d$, such that $r_i = \|\Pi_{\mathcal{S}}(e_i)\|_2^2$ for $i \in [d]$, where $\Pi_{\mathcal{S}}(e_i) \in \mathbb{R}^d$ is the projection of standard basis e_i onto $\mathcal{S}$, and thus $r_i = \sum_{j=1}^{s} \langle e_i, v_j \rangle^2$ where $s = \dim(\mathcal{S})$ and $\{v_1, \cdots, v_s\}$ is a set of orthonormal bases for $\mathcal{S}$.

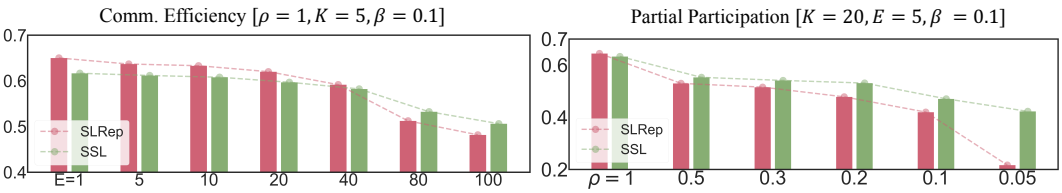

Figure 4: **Dec-SSL performance on ImageNet-100 dataset**. Compared to supervised learning, we observe that under non-IID settings, decentralized SSL can perform better under communication constraints (left) and partial participation constraints (right).

The intuition of this definition is that a good feature space should have the property that many standard unit bases among $e_1, \cdots, e_d$, which can be used to represent any vectors in $\mathbb{R}^d$, can be represented well by the feature space, i.e., have large projections onto it. Note that as a vector, $\boldsymbol{r}$ provides a quantitative way to compare the representability of two feature spaces across different directions (i.e., different unit basis). In the following theorem, we compare the representability learned by local objectives and the global one, for Dec-SSL.

Theorem 3.2 (Representability of local v.s. global objectives for Dec-SSL)**.** For decentralized SSL in the setting described above, with high probability, the representability vector learned from any local objective of source k, denoted by $\boldsymbol{r}^k = [r_1^k, \cdots, r_d^k]^\top$, satisfies that $1 - O(d^{-4/5}) \leq r_i^k \leq 1$ for all $i \in [K] \backslash \{k\}$. Moreover, the representability vector learned from the global objective, denoted by $\bar{\boldsymbol{r}} = [\bar{r}_1, \cdots, \bar{r}_d]^\top$, satisfies that $1 - O(d^{-4/5}) \leq \bar{r}_i \leq 1$ for all $i \in [K]$.

Theorem 3.2 states that the feature spaces learned from local SSL objectives are relatively *uniform*, in the sense that for the K basis directions $e_1, \cdots, e_K$ that generate the data, any two data sources have similar representability in *all of them but two* directions, especially when the dimension d of the data is large. Furthermore, when solving the global objective (2.2), the learned representation is also uniform, and its representability differs *at most one* direction from that of each local data source. Note that the results hold with highly heterogeneous data across data sources. In other words, Dec-SSL is not affected significantly by the non-IIDness of the data, justifying the empirical observations in §3.1. Illustration of the results can also be found in Figure 3.

Intuition & implication. The main intuition behind Theorem 3.2 is that, the objective of SSL is not *biased* by the heterogeneous distribution of labels at each local dataset, and tends to learn uniform representations. Related arguments have also been made in the recent works on the theoretical understanding contrastive learning/SSL (Wang & Isola, 2020; Liu et al., 2021). In the decentralized setting, this insensitivity to data heterogeneity becomes even more relevant, as it potentially allows each local data source to perform much more local updates, without drifting the iterates significantly. This enables more communication-efficient decentralized learning schemes, in contrast to most existing ones that are vulnerable to data non-IIDness. We validate these points next.

4 Dec-SSL Can be Favorable Even When Labels are Available

We here seek to address question (ii) in §2.1 – how does the unique property of Dec-SSL, such as the robustness to data heterogeneity, benefit decentralized learning? While lack of labels seems a limitation, we show that this might not be the case in decentralized learning with heterogeneous data. First, it is known that decentralized SL in general performs poorly when the data is highly heterogeneous (Zhao et al., 2018; Hsieh et al., 2020). Further, even in the decentralized representation learning setting when labels are available, Dec-SSL still stands out in face of highly non-IID data.

To make a fair comparison, we mainly compare Dec-SSL with Dec-SLRep (recall the definition in §2.1), which are both decentralized *representation learning* approaches. We defer the comparison with Dec-SL to Appendix §B. We conduct experiments on both ImageNet and CIFAR-10 datasets, and evaluate the performance of the learned representations in terms of the variations of two commonly used metrics in decentralized learning – the number of local updates epochs E, and the participation ratio of data sources ρ. We observe consistently that Dec-SSL indeed outperforms Dec-SLRep in learning representations in terms of communication efficiency and participation ratio, especially with highly non-IID data. We remark that such observations are also consistent with those on object detection and semantic segmentation given in Table 1.

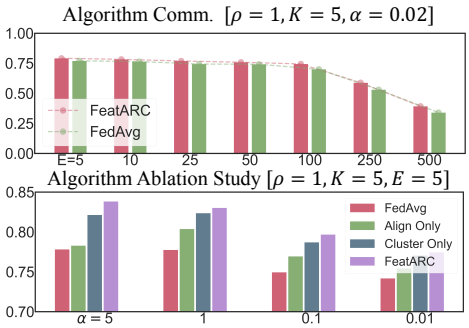

Figure 5: **Ablation study on the `FeatARC` algorithm.** We observe that under non-IIDness and communication constraints, `FeatARC` outperforms the baseline variants of the algorithm and `FedAvg`.

Method / Setting	IID	non-IID
FURL (Zhang et al., 2020a)	71.25	68.01
EMA (Zhuang et al., 2022)	86.26	83.34
Per-SSFL (He et al., 2021a)	N/A	83.10
FEDU (Zhuang et al., 2021)	83.96	80.52
`FeatARC` (Ours)	**86.74**	**84.63**

CIFAR-100 Pretrain	CIFAR-10 100%	CIFAR-10 10%	CIFAR-10 1%
no pretrain	0.31	0.27	0.25
Dec-SLRep IID	0.65	0.60	0.47
Dec-SSL IID	**0.71**	**0.67**	**0.57**
Dec-SLRep Non-IID	0.43	0.35	0.32
Dec-SSL Non-IID	**0.70**	**0.66**	**0.57**

Table 2: **Top). Algorithm performance comparison. Bottom). CIFAR-10 Linear probing on the representation of CIFAR-100.** Our algorithm surpasses previous works on federated SSL both in the IID and non-IID settings.

4.1 Experimental observations

In this experiment, we train and evaluate the feature backbone on ImageNet-100 in a decentralized setting. We create non-IIDness across the local datasets based on label skewness and use $\beta = 0.1$ (each data source has only 10% of its data coming from the uniform class distributions).

Communication efficiency under high non-IIDness. In Figure 4, we show that under the non-IID scenario, averaging weights with an infrequent communication schedule causes less trouble to Dec-SSL than to Dec-SLRep. In `FedAvg`, the idea of averaging weights after multiple epochs might sound sub-optimal, but we notice that decentralized SSL is very robust with respect to this parameter. Intuitively, the robustness of Dec-SSL allows each local model to drift longer, leading to a lower communication frequency for decentralized learning.

Participation ratio under high non-IIDness. In this experiment, we split ImageNet-100 into 20 data sources and use local update $E = 5$ epochs. We measure the performance of decentralized learning algorithms with respect to the participation ratio of data sources at each round. For instance, when $\rho = 1$, at each round, all data sources update their local weights and upload to the server, while $\rho = 0.05$ means that each round a single random data source is selected for update. On the right of Figure 4, we show that with non-IID data, the convergence of Dec-SSL is more stable to less participants compared to Dec-SLRep. This allows more efficient decentralized learning, especially when deployed with extremely large number of data sources and unstable communication channels.

4.2 Theoretical insights

To shed lights on the above observations, we provide analysis for the feature spaces learned by the local objective of Dec-SLRep, under the same setup as in §3.2. For Dec-SLRep and each data source k, we consider learning a two-layer linear network $g_{u_k,v_k}(x) := v_k u_k x$ as classifier, where $u_k \in \mathbb{R}^{m\times d}$ and $v_k \in \mathbb{R}^{c\times m}$, and use $u_k x$ as the learned representation for downstream tasks. The network is learned by minimizing $\|(u_k)^\top u_k\|_F^2 + \|(v_k)^\top v_k\|_F^2$ subject to the margin constraint that $[g_{u_k,v_k}(x)]_y \geq [g_{u_k,v_k}(x)]_{y'} + 1$ for all data (x, y) in the local dataset k with all $y' \neq y$. . We now have the following proposition on the representations learned by Dec-SLRep across data sources.

Proposition 4.1 (Representations learned by Dec-SLRep across heterogeneous data sources)**.** With high probability, the features $u_k = [u_{k,1}, \cdots, u_{k,m}]^\top \in \mathbb{R}^{m\times d}$ learned from the local dataset D_k satisfies that $\sum_{i=1}^m \langle u_{k,i}, e_j\rangle^2 \leq O(d^{-\frac{1}{10}})$, for $j \in [K]\backslash\{k\}$; while $\sum_{i=1}^m \langle u_{k,i}, e_k\rangle^2 \geq 1 - O(d^{-\frac{1}{20}})$. In other words, the correlation between the learned features in w_k and e_j is small for all $j \in [K] \setminus \{k\}$, while the correlation between the features and e_k is large.

The proposition suggests that the feature spaces learned by Dec-SLRep differ significantly across local data sources, given the highly heterogeneous data. More specifically, we show that most of the unit bases in $\{e_1, \cdots, e_K\}$ have small correlations with the features learned at each local data source, while these feature spaces themselves vary significantly across data sources. The unit bases that are not learned might be significant for various other downstream tasks, making the learned representations less favorable. This heterogeneity among local solutions is not in favor of *local updates*, as too many local updates would drift the iterates towards its local solution, and the iterates

Algorithm 1 Feature Alignment Regularization and Clustering (`FeatARC`)

Input: Cluster number C, initialization of cluster and local models $\{\theta_j\}_{j\in[C]}$ and $\{\widetilde{\theta}_i\}_{i\in[K]}$
Parameters: Number of local updates E, number of total rounds T, distance function $\mathbb{D}$, learning rate γ, local datasets $D_1, ..., D_K$
for $t = 0, ..., T-1$ **do**
  Central server: Broadcast cluster parameters $\{\theta_j\}_{j\in[C]}$; choose $\mathcal{M}$, a random subset of data sources to participate at round t
  for Data source $i \in \mathcal{M}$ in parallel **do**
    Initialize local and global feature sets $z_{i,1}, ..., z_{i,C}, \widetilde{z}_i$
    for $j \in [C]$ **do**
      for Data sample $x_k \in D_i$ **do**
        Compute **global feature**: $z_{i,j} \leftarrow z_{i,j} \cup \{f_{\theta_j}(x_k)\}$
        Compute **local feature** : $\widetilde{z}_i \leftarrow \widetilde{z}_i \cup \{f_{\widetilde{\theta}_i}(x_k)\}$
      end for
      Compute average feature alignments: $A_{i,j} = \frac{1}{|z_{i,j}|}\sum_{k=1}^{|D_i|}\mathbb{D}(z_{i,j,k}, \widetilde{z}_{i,k})$
    end for
    Estimate cluster identity: $I_i \leftarrow \arg\min_{j\in[C]} A_{i,j}$
    Update **local model**: $\widetilde{\theta}_i \leftarrow \texttt{LocalUpdate-FAR}(E, \gamma, \theta_{I_i})$
    Send back $\widetilde{\theta}_i$ and the one-hot vector $s_i = \{s_{i,j}\}_{j\in[C]}$ with $s_{i,j} = \mathbf{1}_{\{j=I_i\}}$
  end for
  Central server: Update cluster model $\theta_j \leftarrow \frac{\sum_{i\in\mathcal{M}} s_{i,j}\widetilde{\theta}_i}{\sum_{i\in\mathcal{M}} s_{i,j}}$ for all $j \in [C]$
end for

would become too far away from each other, hurting the convergence of decentralized learning. Hence, compared with the Dec-SSL case and Theorem 3.2, Dec-SLRep can be less robust to data heterogeneity and less communication-efficient. We note that the advantage of Dec-SSL does not come from *using more data*, since we use exactly the same data for training Dec-SLRep and Dec-SSL. The intuition is also illustrated in Figure 3.

5 Algorithm `FeatARC` (Feature Alignment and Clustering)

Although Dec-SSL tends to learn relatively uniform features that are robust across datasets, the uniformity itself might not imply the alignment of features across datasets: the representation network from different local data sources can still map the same data point to different regions in the feature space. This misalignment becomes more significant when the data is highly non-IID and can have an adverse effect on the model aggregation process in decentralized learning (Zhang et al., 2020a). To mitigate this issue and address question (iii) in §2.1, we propose to use the same feature distance loss as an auxiliary local objective to align the local models with the global model. The alignment between two features is defined as the negative cosine distance metric $\mathbb{D}(z_1, z_2) = -\frac{z_1 \cdot z_2}{||z_1||||z_2||}$.

To further improve the Dec-SSL algorithm, we propose to learn *multiple models* using clustering-based approach. In particular, instead of learning a single global model as in (2.2), we learn C models and separate the K data sources into C clusters. The update of C models and the assignment of data sources to C clusters are conducted alternatively. When $C = K$, the algorithm reduces to learning K local models; when $C = 1$, it reduces to learning a single global one. The clustering approach intuitively learns multiple models to interpolate the performance between learning a *single global* model and K *local* models, thus achieving a good bias-variance tradeoff when testing on each local dataset (Mansour et al., 2020; Ghosh et al., 2020). However, unlike the supervised learning case, we do not use the loss of the decentralized learning (i.e., (2.1)) as the metric for clustering. This is because for contrastive learning, it has been observed that the SSL loss might not be indicative enough for the performance of the representation on downstream tasks (Robinson et al., 2021). Hence, we here again use the feature alignment distance $\mathbb{D}(\cdot, \cdot)$ as the metric for clustering.

We adopt the alignment regularization and clustering techniques, and developed a new Dec-SSL algorithm `FeatARC`, summarized in Algorithm 1 and Algorithm 2 with more details in §C.3. We show the performance of `FeatARC` in Figure 5, in comparison with different baselines including `FedAvg`, under different levels of data heterogeneity and communication frequency. It is shown

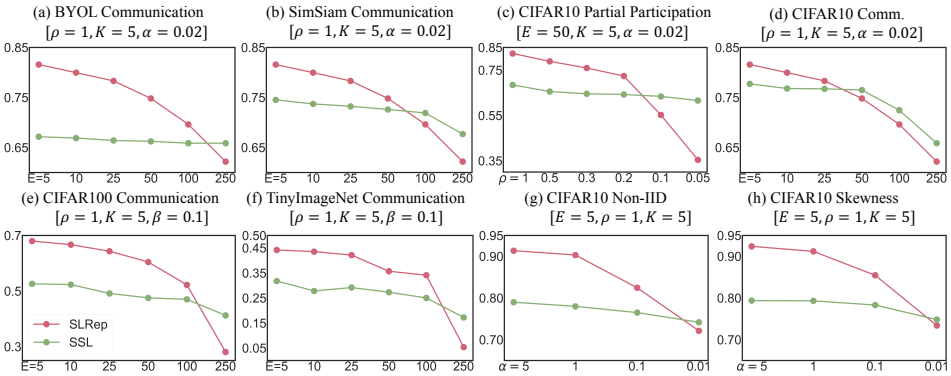

Figure 6: **SSL method and dataset ablation study.** We conduct ablation study on SSL methods SimSiam and BYOL, as well as on datasets CIFAR-10, CIFAR-100, and TinyImageNet.

that `FeatARC` outperforms the baselines consistently, including the variants that only uses alignment ("Align Only") or clustering ("Cluster Only"). Moreover, on the top of Table 2, we show that `FeatARC` also outperforms other recent decentralized self-supervised learning algorithms on CIFAR-10 dataset.

6 Ablation Study

Ablation on SSL algorithms. We ablate on the learning algorithms in the Dec-SSL setting. We experiment with SSL methods SimSiam (Chen & He, 2021) and BYOL (Grill et al., 2020) in addition to SimCLR to learn representations. From Figure 6 (a,b), we have consistent observations on the robustness to data non-IIDness, and the stable performance when reducing the communication frequency. These observations confirm that the SSL objectives are in general leading to relatively uniform features, and are less vulnerable to data heterogeneity with communication constraints.

Ablation on dataset. Furthermore, we ablate decentralized learning on standard datasets such as CIFAR-100 and Tiny-ImageNet (Le & Yang, 2015) and observe that Dec-SSL outperforms Dec-SLRep with communication constraints and non-IIDness (Figure 6 (e,f)). For CIFAR-10, we also found similar robustness to non-IIDness and skewness (Figure 6 (g,h)) as well as partial participation and communication constraints (Figure 6 (c,d)). On Table 2 Bottom, we show that the learned representations from Dec-SSL on one data source (CIFAR-100) can transfer to other data sources (CIFAR-10). Additional ablation study can be found in §B.

7 Conclusion

We propose the framework of decentralized SSL that learns representations from non-IID unlabeled data and conduct an empirical study on the robustness of Dec-SSL to different types of heterogeneity, communication constraints, and partial participation of data sources. We also provides some findings and theoretical analyses of Dec-SSL compared to its supervised learning counterpart, as well as investigate a new algorithm to further address the high heterogeneity in decentralized datasets.

Acknowledgement. This work is supported in part by Amazon.com Services LLC, PO2D-06310236 and Defense Science & Technology Agency, DST00OECI20300823. L.W. was supported by the MIT EECS Xianhong Wu Graduate Fellowship. K.Z. also acknowledges support from Simons-Berkeley Research Fellowship. We thank MIT Supercloud for providing compute resources. The authors would like to thank many helpful discussions from Phillip Isola at MIT and Andrew Marchese at Amazon.

References

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportu-

nities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2959–2968, 2019.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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

Adam Coates, Andrew Ng, and Honglak Lee. 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, 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021.

Chaoyang He, Zhengyu Yang, Erum Mushtaq, Sunwoo Lee, Mahdi Soltanolkotabi, and Salman Avestimehr. Ssfl: Tackling label deficiency in federated learning via personalized self-supervision. *arXiv preprint arXiv:2110.02470*, 2021a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021b.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.

Kevin Hsieh, Aaron Harlap, Nandita Vijaykumar, Dimitris Konomis, Gregory R Ganger, Phillip B Gibbons, and Onur Mutlu. Gaia: {Geo-Distributed} machine learning approaching {LAN} speeds. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pp. 629–647, 2017.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2020.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2018.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pp. 577–593. Springer, 2016.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021a.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021b.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL https://openreview.net/forum?id=vUz4JPRLpGx.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

Nan Lu, Zhao Wang, Xiaoxiao Li, Gang Niu, Qi Dou, and Masashi Sugiyama. Federated learning from only unlabeled data with class-conditional-sharing clients. In *International Conference on Learning Representations*, 2022.

Disha Makhija, Nhat Ho, and Joydeep Ghosh. Federated self-supervised learning for heterogeneous clients. *arXiv preprint arXiv:2205.12493*, 2022.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Angelia Nedic. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in Neural Information Processing Systems*, 34, 2021.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020a.

Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *European Conference on Computer Vision*, pp. 266–282. Springer, 2020b.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pp. 1179–1206. PMLR, 2021.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

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

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pp. 7252–7261. PMLR, 2019.

Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020a.

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020b.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4912–4921, 2021.

Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *International Conference on Learning Representations*, 2022.

Supplementary Materials for "Does Decentralized Learning with Non-IID Unlabeled Data Benefit from Self Supervision?"

CONTENTS

A Detailed Related Work

We here provide a more detailed review of the literature.

Self-supervised learning. Self-supervised learning aims to learn useful representations from data without human annotations. A breadth of methods has been proposed such as colorization (Larsson et al., 2016), impainting (Pathak et al., 2016), and denoising autoncoder (Vincent et al., 2008). One promising approach is contrastive learning, where the core idea is to find a representation space that makes positive pairs close and negative pairs apart. (Oord et al., 2018; Wu et al., 2018; Tian et al., 2020a; Chen et al., 2020; He et al., 2020; Chen & He, 2021; Grill et al., 2020; Caron et al., 2020) used a self-supervised pretraining objective for transformation-invariant representation, and demonstrated good performance on standard datasets such as ImagetNet classification (Krizhevsky et al., 2012), COCO detection and segmentation (Lin et al., 2014), and uncurated dataset (Caron et al., 2019). Despite that no label is needed, the representation learned by SSL has been shown to be robust to distribution shift (Liu et al., 2021), generally applicable in embodied agent tasks (Florence et al., 2018; Hendrycks et al., 2019), and adapt quickly in low-data regime (Tian et al., 2020b). More importantly, SSL models have been deployed in many large systems (foundation models (Bommasani et al., 2021)) in NLP (Brown et al., 2020) and intersection of language and vision (Radford et al., 2021; Jia et al., 2021). Other significant examples of SSL include masked auto-encoding in language (Devlin et al., 2018) and vision (He et al., 2021b). There has also been a growing literature on the theoretical understandings of SSL, with representative examples (Arora et al., 2019; Lee et al., 2021; Tosh et al., 2021; HaoChen et al., 2021). Very recently, (Liu et al., 2021) observed that self-supervised learning is more robust to dataset imbalance, more specifically, the data label imbalance. Interestingly, their observations are aligned with ours with decentralized heterogeneous data (though we focus on not only the label skewness among data sources), and their analysis also provide important insights into our observations. While all these works consider SSL in a centralized setting, our goal is to further understand and unlock the power of SSL in a decentralized setting, a practical while relatively underexplored one where *large-scale unlabeled* data is more relevant.

Decentralized machine learning. With massive amounts of data generated in a distributed fashion, decentralized learning has achieved increasing attention in the literature (Konečnỳ et al., 2016; Lian et al., 2017; McMahan et al., 2017; Hsieh et al., 2017; 2020; Karimireddy et al., 2020; Kairouz et al., 2021; Nedic, 2020), where a global model is trained over distributed data sources, addressing research questions on communication-efficiency (between the worker and the server or among workers) and privacy of the data. In addition, addressing the *heterogeneity/non-IIDness* of data distributions across sources has been, and remains to be the most important research question in the area (Zhao et al., 2018; Hsieh et al., 2020; Karimireddy et al., 2020; Ghosh et al., 2020; Li et al., 2021a). Most existing decentralized learning studies have extensively focused on *supervised learning* setting, where the labels of the data samples are required.

Federated unsupervised/self-supervised learning. To the best of our knowledge, there have only been a few contemporaneous/concurrent attempts (Zhang et al., 2020a; He et al., 2021a; Zhuang et al., 2021; 2022; Lu et al., 2022; Makhija et al., 2022) that bridged *unsupervised/self-supervised learning* with unlabeled data and decentralized learning, more specifically federated learning (FL), and proposed various algorithms to *mitigate* the effect of data heterogeneity. In particular, the works He et al. (2021a); Zhuang et al. (2021; 2022); Lu et al. (2022); Makhija et al. (2022) are closest to ours. He et al. (2021a), also motivated by the label-deficiency issue in federated learning, developed a series of self-supervised FL algorithms that incorporated the advances of supervised FL, especially those algorithms with *personalization*, to handle the heterogeneity in data. Zhuang et al. (2021) developed unsupervised representation learning algorithms from unlabeled data, mainly with the motivation of privacy-preserving, by designing communication protocol and divergence-aware predictor update rules that are *specific to* Siamese architecture. Zhuang et al. (2022) further improved the results by generalizing to other SSL approaches, proposing a new divergence-aware update rule, and ablating on how the components of these SSL approaches affect the performance. Later, Lu et al. (2022) also aimed to address the data-deficiency issue in FL, by training a *modified* model using *supervised* FL over the *surrogate labeled* data transformed from the unlabeled ones. The transformation requires knowledge of the class priors at each data source, and the approach is not relevant to self-supervised/contrastive learning, the focus of our paper. Finally, Makhija et al. (2022) pro-

Experiment	Pretrain	K	E	ρ	α	β	Evaluation
Figure 2	CIFAR-10	5	50	1	*	–	CIFAR-10
Table 1 Left	ImageNet-100	5	1	1	0.2	–	MS-COCO
Table 1 Right	Amazon	5	1	1	–	0	Amazon
Figure 4 Left	ImageNet-100	5	*	1	–	0.1	ImageNet-100
Figure 4 Right	ImageNet-100	20	5	*	–	0.1	ImageNet-100
Figure 5 Left Top	CIFAR-10	5	*	1	0.02	–	CIFAR-10
Figure 5 Left Bottom	CIFAR-10	5	5	1	*	–	CIFAR-10
Table 2 Top	CIFAR-10	5	5	1	–	–	CIFAR-10
Table 2 Bottom	CIFAR-100	5	5	1	–	–	CIFAR10
Figure 6 (a,b)	CIFAR-10	5	*	1	0.02	–	CIFAR-10
Figure 6 (c)	CIFAR-10	5	50	*	0.02	–	CIFAR-10
Figure 6 (d)	CIFAR-10	5	*	1	0.02	–	CIFAR-10
Figure 6 (e)	CIFAR-100	5	*	1	–	0.1	CIFAR-100
Figure 6 (f)	TinyImageNet	5	*	1	–	0.1	TinyImageNet
Figure 6 (g)	CIFAR-10	5	50	1	*	–	CIFAR-10
Figure 6 (h)	CIFAR-10	5	50	1	–	*	CIFAR-10
Figure 7 (a)	CIFAR-10	5	50	1	*	–	CIFAR-10
Figure 7 (b)	CIFAR-10	5	50	1	–	*	CIFAR-10
Figure 7 (c)	CIFAR-10	5	*	1	0.02	–	CIFAR-10
Figure 7 (d)	CIFAR-10	20	5	*	0.02	–	CIFAR-10
Figure 8 (c)	CIFAR-10	5	5	1	*	–	CIFAR-10
Figure 8 (d)	CIFAR-10	5	*	1	0.02	–	CIFAR-10
Table 4	CIFAR-10	5	5	1	–	–	STL-10

Table 3: **Table of experiment setups**. Note that the full experiment pipeline has two steps: pretrain and evaluate. There are two datasets, the pretrain dataset, and the evaluation dataset. For detection tasks on MS-COCO and Amazon, the pretrained encoder is also updating through the transfer learning procedure and for other datasets, only a single linear layer is trained with the pretrained encoder being frozen. $*$ denotes the control variable in each experiment and $-$ denotes the variables that are not used.

posed a self-supervised federated learning algorithm to handle the heterogeneity in data, by adding a proximal term that measures the distance between the local representations and those obtained on other clients in the local objective. The algorithm requires the server to directly access the unlabeled datasets, and also requires some *datasets* for representation alignment to be transmitted between the server and the clients.

Our focus. To be specific, our focus is *not* on finding better algorithms to handle/mitigate data heterogeneity in decentralized learning with unlabeled data, but on *understanding* the use of self-supervised learning approaches, in particular contrastive learning, in decentralized learning – whether and when decentralized SSL is effective and/or even advantageous (even combined with simple and off-the-shelf decentralized learning algorithms, e.g., `FedAvg`); what are the unique and inherent properties of decentralized SSL (compared to its SL counterpart); how may the properties play a role in decentralized learning (especially with highly heterogeneous unlabeled data)? Moreover, except Lu et al. (2022); Makhija et al. (2022), which contained some convergence analysis for the algorithms they developed, these contemporaneous works usually did not provide theoretical insights about why decentralized SSL is used to handle decentralized unlabeled data, and when it is effective/advantageous (even sometimes the labels are available). Finally, our goal is to advocate the Dec-SSL *framework*, and the approach is not specific to certain network architecture (as e.g., Zhuang et al. (2021; 2022)), and does not require transmitting datasets (e.g., Makhija et al. (2022)). Finally, our empirical observations are thoroughly verified on *larger-scale* datasets compared to these works, e.g., ImageNet, MS-COCO, and real-world robotic warehouse datasets, which are more relevant to practical applications.

B Additional Experiments

We present the experiment details of Dec-SSL on CIFAR-10, similar to those in Section §4 on ImageNet with implementation details in the section §C.1 Note that we also include the baseline **Dec-SL** where the algorithm directly runs `FedAvg` on the downstream classification tasks, without

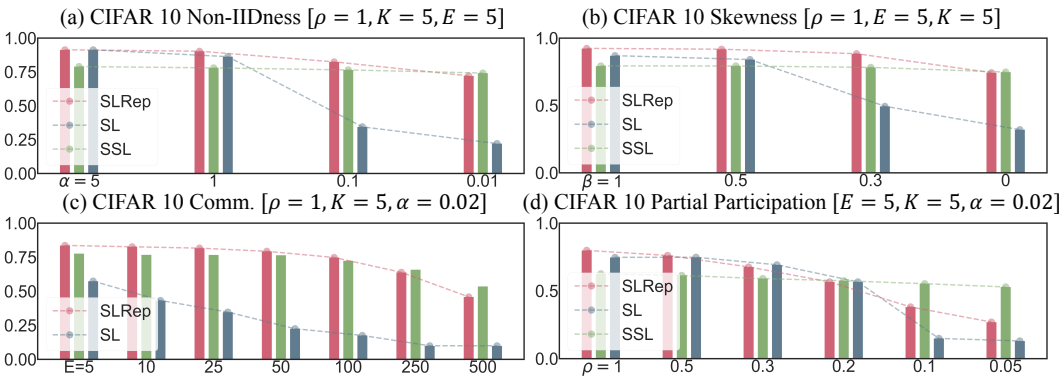

Figure 7: **CIFAR-10 Experiments.** SSL is more robust to non-IIDness, communication efficiency, and participation ratios on CIFAR-10 Dataset.

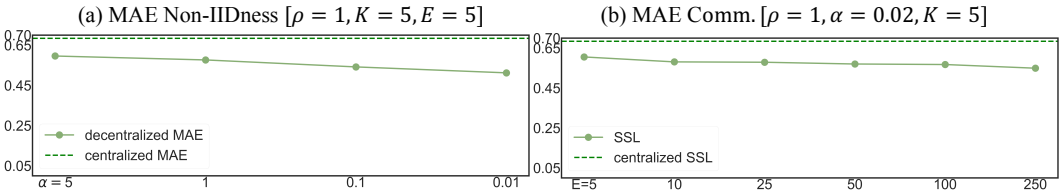

Figure 8: **Masked Autoencoder (MAE) Experiments.** We show that the more recent SSL approaches based on masked autoencoder (He et al., 2021b) and vision transformer (Dosovitskiy et al., 2020) are also robust to data non-IIDness and communication constraints. This supports that the advantages of Dec-SSL is not restricted to contrastive approaches and convolutional networks.

explicitly learning a representation. It is known (Hsieh et al., 2020) that the non-IIDness is particularly challenging to deal with in decentralized supervised learning, and we also confirm it in our experiments.

Decentralized SSL with non-IID data. On Figure 7 (a), we show that Dec-SSL is more robust than Dec-SLRep when we apply dirichlet label shift to create non-IIDness at different levels. We also observe that Dec-SLRep outperforms Dec-SL in this decentralized setting. This observation on the CIFAR-10 dataset is consistent with the ImageNet-100 dataset in Section §4.

Decentralized SSL can have better communication efficiency under non-IIDness. We use $\alpha = 0.02$ in this experiment. Under two different notions of non-IIDness, Dec-SSL is much more robust to communication efficiency compared to Dec-SLRep and Dec-SL. While the idea of averaging weights after multiple steps sounds challenging, it is surprising to see how robust Dec-SSL is with respect to the communication frequencies E in Figure 7 (b,c). Similar to the ImageNet experiments, Dec-SLRep is less robust to the communication frequencies, and Dec-SL is more brittle to less communication. For CIFAR-10 experiments, each epoch has around 50 iterations.

Decentralized SSL allows less participation under non-IIDness. We use $\alpha = 0.02$ in this experiment and fix the total number of epochs to be 500. We use $K = 20$ data sources in this experiment and want to measure the convergence of decentralized algorithms with respect to the participation of data sources at each round. In Figure 7 (d), we show that with non-IID data, SSL is much more robust to less participant each round compared to Dec-SL.

Learning representation transferable to different data sources. The idea of transfer learning has been used in the self-supervised learning literature (He et al., 2020; Pathak et al., 2016) and we apply similar ideas to the decentralized learning setting. In this case, the new data distribution could be treated as *a new user/data source*, which we want to perform well and adapt quickly on. We have additional results of linear probing from CIFAR-10 dataset to STL-10 dataset (Coates et al., 2011) in Table 4. We found a strong correlation of the downstream classification performance and the transfer learning performance, as they both rely on the representation capacity of the pretrained network.

CIFAR-10 Pretrain	STL-10 100%	10%	1%
no pretrain	0.25	0.24	0.13
Dec-SSL IID	**0.65**	**0.61**	**0.48**
Dec-SLRep IID	**0.65**	0.60	0.47
Dec-SSL Non-IID	0.60	0.54	0.36
Dec-SLRep Non-IID	0.31	0.30	0.25
Dec-SSL Non-IID Less Comm.	0.33	0.28	0.17

Table 4: Linear Probing from CIFAR-10 to STL10. We observe that pretraining on non-IID data can negatively affect the performance of transfer learning with different amounts of data. The learned representation from Dec-SSL can improve both the downstream tasks on the same dataset and help transfer to a new dataset.

Dec-SSL with masked autoencoder. In this experiment, we run the more recent SSL approach, masked autoencoder, on the CIFAR-10 dataset to investigate its robustness to data non-IIDness as well as communication efficiency. We use Vit-Tiny (Dosovitskiy et al., 2020) with the AdamW (Loshchilov & Hutter, 2018) optimizer for 1000 epochs with batch size 256. We note that the linear probing performance of MAE is not as good as contrastive learning. However, as shown in Figure 8, we still observe a similar stable trend in terms of the downstream performance, as the non-IIDness and the number of local updates increase. This indicates that the advantage of Dec-SSL is not restricted to contrastive approaches and convolutional neural networks.

C Supplementary Details

C.1 Implementation details

In representation learning, we aim to *pretrain* a network model on a dataset with some pretext tasks and *transfer* the weights to another problem, potentially a new dataset and a specific downstream task. The most widely and practically used representation is the pretrained weights supervised learning on ImageNet (He et al., 2016), as an initialization for finetuning or training on downstream tasks such as classification and detection. Recently, self-supervised representation learning has attracted increasing attention. It is common to study the performance and behavior of representations through evaluating on downstream tasks. We follow the same setup and try to understand the visual representation learning under the decentralized learning setting. For reference, Table 3 shows a list of datasets used for different experiments in the paper.

Unless otherwise noted, we use ResNet18 (He et al., 2016) throughout the experiments and train for 500 epochs with the Adam optimizers (Kingma & Ba, 2014), learning rate 0.001, and batch size 256. We use SimCLR (Chen et al., 2020) as the default SSL algorithm due to its simplicity. For masked autoencoder (He et al., 2021b) experiment on CIFAR-10, we use Vit-Tiny (Dosovitskiy et al., 2020) with AdamW optimizer for 1000 epochs with batch size 256. We note that the linear probing performance of MAE is not as good as contrastive learning. Note that in all experiments, the unit for local update number is epoch instead of iterations (e.g., $E = 5$ means each local data source would update 5 epochs, about 200 iterations, before averaging). Note that each epoch on CIFAR-10 for $K = 5$ data sources is $\delta = 50$ iterations and we fix the number of total epochs for all experiments. For ImageNet experiment, we use a learning rate of 0.005 with $E = 200/\rho$ epochs, where ρ is the participation ratio of data sources. For SimSiam and BYOL, we use a learning rate of 0.03 with the SGD optimizer. We consider the standard classification benchmark dataset such as CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), ImageNet (Krizhevsky et al., 2012), TinyImageNet (Le & Yang, 2015), STL-10 (Coates et al., 2011) and detection dataset such as COCO (Lin et al., 2014) and a real-world package detection dataset that comes from Amazon. We only use a subset of the Amazon dataset which has around 80000 RGB images with contour labels predicted by the Amazon systems. We use the SimCLR image augmentation for all view augmentation without Gaussian blurring on CIFAR and the standard version on ImageNet. The temperature for SimCLR is fixed to be 0.5. For classification tasks, to evaluate the learned representation, we initialize a linear classifier after the feature encoder and train it until convergence on the centralized training set, and then evaluate it on the centralized test set.

For finetuning detectron (Girshick et al., 2018) on COCO and Amazon datasets, we use the default schedule with 90000 iterations and the FPN backbone, batch size 16, and learning rate 0.02. We

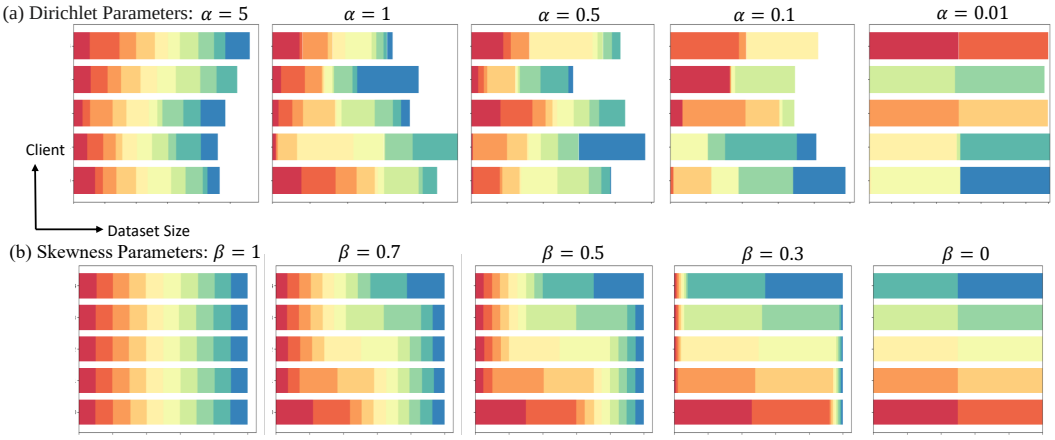

Figure 9: **Visualization of the label distribution shift for CIFAR-10.** Each horizontal bar represents the data for one data source where one color indicates one class in CIFAR-10 (10 classes in total), and the vertical axis represents different data sources. To study the effect of non-IIDness on decentralized learning, we use Dirichlet process (with parameter α) and a skewness ratio (with parameter β) to split the data. We can observe from left to right that the data becomes more and more non-IID as we adjust the parameters.

use a centralized dataset whose distribution is the union of all local data sources. For Amazon experiments, recall that each session is considered as a local data source, and we run pretrain with Dec-SSL with each session trained individually and then communicate. For the evaluation phase on Amazon package detection / segmentation tasks, we train on a subset of 10000 images of the unlabeled data for 20000 iterations to show the benefits of representation learning. For this segmentation task, we use the outputs of the Amazon systems as the "ground-truth", but we note that they can be inaccurate. For detection and segmentation tasks, the training and evaluation setups follow those in the Detectron (Girshick et al., 2018) pipeline, with only the initialization weight being replaced.

For both `FedAvg` (McMahan et al., 2017) and `FeatARC`, we use 5 data sources ($K = 5$) with evenly split number of data per data source. Each round we use full participation ($\rho = 1$) with 5 local update epochs ($E = 5$). We use step scheduler to gradually decay the learning rates and reset all local optimizer states for each round in the CIFAR-10 experiments, and do not average the BatchNorm layers (Li et al., 2021b). In `FeatARC`, we find 2 clusters to be sufficient to achieve good performance and also use hyperparameter $\lambda = 1$. During evaluation, we test on each local dataset using the corresponding cluster model, and average the best performance as the classification accuracy. All experiments run on one V100 GPU and finish within a day. We use a customized ResNet to process the CIFAR image, and these experiments take much less resource and time. Note that although we typically compare Dec-SLRep and Dec-SSL on the same dataset, in practice the unlabeled dataset has much larger diversity and quantity.

C.2 Data heterogeneity creation details

In this section, we discuss how we construct the non-IIDness of datasets on CIFAR-10. The same procedure applies to other datasets used in the paper. Assume that we split the dataset into N partitions (Note that it is different from the number of data sources K), and these $D_1, .., D_N$ are based on some sources of the heterogeneity. Once we have these N partitions, we use two different ways to create the data non-IIDness across data sources. The first method is to use a Dirichlet distribution to split $D_1, ..., D_N$ (Yurochkin et al., 2019). As a multivariate generalization of the Beta distribution, Dirichlet distribution generates sample $p_k \sim Dir_N(\alpha)$ and assigns a portion $p_{k,j}$ of the class k to data source j. Note that α represents a concentration parameter. When α increases to the limit of ∞, the distribution becomes more and more IID (each data source has roughly a uniform distribution). Empirically for CIFAR-10 with 50000 data points and 10 classes (Figure 9), $\alpha = 5$ implies a reasonably uniform distribution over 10 classes and $\alpha = 0.01$ implies an non-IID case each data source has data from mostly 2 classes and a small amount comes from other uses. Another way to create non-IIDness is through skewness partitioning (Hsieh et al., 2020). In this case, we separate the entire dataset into (β) fraction that would split uniformly to each partition and $(1 - \beta)$ fraction that would split in a skewed way. Assume we have N partitions, then each

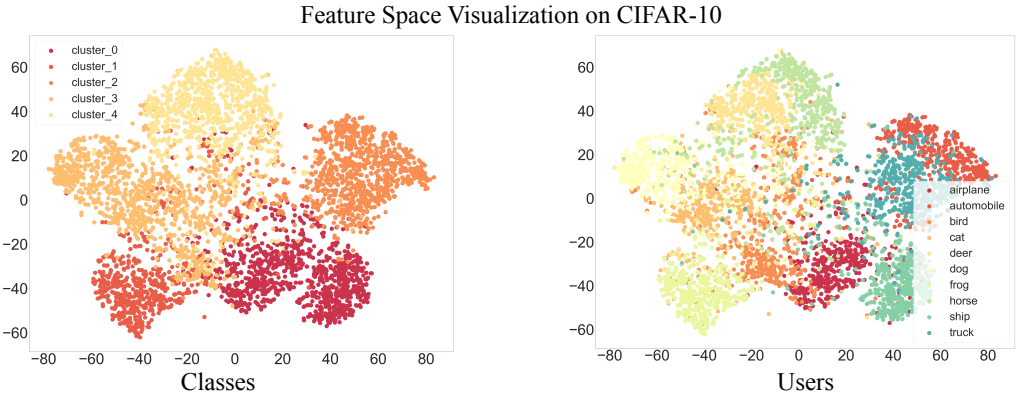

Figure 10: **Feature distribution shift.** Heterogeneity created with a pretrained feature extractor. We use a pretrained network to map images to a feature space and then do clustering to split the dataset.

data source would have β fraction of its data coming from the IID distribution of the dataset, and $(1-\beta)$ fraction that comes from $\lfloor N/K \rfloor$ of the partitions exclusively. As we decreases β from 1 to 0, the dataset becomes more heterogeneous. To see this, observe that $\beta = 1$ implies that the data is completely uniform from 10 classes and $\beta = 0$ means that each dataset has exclusive data from $\lfloor N/K \rfloor$ of the partitions (2 classes from CIFAR). Note that for these two approaches, the non-IIDness level is parametrized by α, β and in the experiment, we consider a range of $\alpha \in [0.01, 5]$ and $\beta \in [0, 1]$.

Label distribution shift. This source of data heterogeneity comes from the class labels of the data samples. Since CIFAR-10 has 10 classes, we can separate the whole datasets into $D_1, ...D_{10}$ as each D_i contains only the 5000 images from one class. For instance, D_1 can be all cat images and D_5 can be all truck images. On Figure 9, we visualize the created non-IIDness on the y distribution (label) by these two approaches.

Feature distribution shift. This source of data heterogeneity comes from the feature space of a pretrained network (Zhang et al., 2020b). Specifically, we first train a pretrained network on classification task on CIFAR-10 with a ResNet50 and use the 2048-dimensional latent vector as a representation of the image feature. After that, we further use Principal Component Analysis (PCA) to reduce the dimension to 30 and do clustering. Treating each feature space cluster as a partition, we create 5 clusters and visualize the cluster ID and the class ID, in Figure 10.

C.3 Algorithm Details

We here introduce more details about the algorithms we proposed in §5. Our new algorithm `FeatARC` is summarized in Algorithm 1; The subroutine of feature alignment regularization in the local updates is tabulated in Algorithm 2. `FeatARC` is based on the idea of clustering in decentralized learning (Ghosh et al., 2020; Mansour et al., 2020), which alternates identifying the cluster identities for each local data source, and using the assigned cluster to do a `FedAvg` step that averages local models. In federated learning, clustering-based approach is often used as an interpolation between learning local (K) models and learning (a single) global model, in order to tradeoff the bias and variance in learning from heterogeneous datasets (Mansour et al., 2020). In the highly non-IID scenarios, classic `FedAvg` with a single global model often fails to capture the heterogeneity of local data distributions, which motivates the use of multiple models, under the assumption that there is some underlying clustering structure of the data (e.g. according to geographic regions, ethnic groups, etc.). Specifically, we denote the sets of cluster models and local models as $\{\theta_j\}_{j\in[C]}$ and $\{\widetilde{\theta}_i\}_{i\in[K]}$, respectively, where C and K denote the number of cluster models and local data sources respectively. At each round, we compute an assignment I_i for each local data source i based on matrix $A \in \mathbb{R}^{K\times C}$ where $A_{i,j}$ denotes the "closeness" of data source i to cluster j. This "closeness" is defined based on how aligned the features are, measured by $\mathbb{D}(\cdot,\cdot)$, between the local feature $f_{\widehat{\theta}_i}(x_k)$ and global feature $f_{\theta_j}(x_k)$ for each data point x_k in the local dataset D_i (Line 8 to 12 in Algorithm 1). Now to estimate the cluster identity for data source i, we use the argmin over the

Algorithm 2 LocalUpdate with Feature Alignment Regularization (`LocalUpdate-FAR`)

Input: Local iteration number E, step size γ, model θ
Parameters: SSL objective L_{SSL}, feature distance metric $\mathbb{D}$, random augmentation function `Aug`, balance parameter λ, local dataset D
Set $\widetilde{\theta} \leftarrow \theta$ as the initialization of the **local model**
for $t = 0, ..., E-1$ **do**
  Sample data pair: x from D and x^- from D independently
  Compute **global feature**: $z_g \leftarrow f_\theta(x)$
  Augment views: $x^+ \leftarrow \texttt{Aug}(x)$
  Compute **local feature**: $(z_l^+, z_l^-, z_l) \leftarrow (f_{\widetilde{\theta}}(x^+), f_{\widetilde{\theta}}(x^-), f_{\widetilde{\theta}}(x))$
  Predict feature: $(p_l^+, p_l^-, p_l) \leftarrow (g_{\widetilde{\theta}}(x^+), g_{\widetilde{\theta}}(x^-), g_{\widetilde{\theta}}(x))$
  Compute loss: $L(\widetilde{\theta}) \leftarrow L_{\text{SSL}}(p_l^+, p_l^-, z_l^+, z_l^-) + \lambda \cdot (\frac{1}{2}\mathbb{D}(p_l^+, z_g) + \frac{1}{2}\mathbb{D}(p_l, z_g))$
  Update **local model**: $\widetilde{\theta} \leftarrow \widetilde{\theta} - \gamma \nabla L(\widetilde{\theta})$
end for
Return: $\widetilde{\theta}$

cluster of the average of feature distance for all data points in the dataset. The assigned cluster model would be sent to the local data source, and is locally updated with the subroutine in Algorithm 2. Throughout the paper, the distance metric (or the alignment as its negative) between features used in SSL loss, auxiliary loss, and clustering identification, is all defined based on a cosine distance metric $\mathbb{D}(z_1, z_2) = -\frac{z_1 \cdot z_2}{||z_1|| ||z_2||}$. We use this distance metric, instead of the SSL loss as the metric, since it has been shown that the SSL loss might not be indicative enough for the performance on downstream tasks (Robinson et al., 2021).

In Algorithm 2, we propose to add the distance of the features from the local model to the features from the global model as an auxiliary loss in the local SSL training, which can be viewed as distilling global model to the local model, or as a trust-region update that restricts the drift of local models. Note that we here refer to the cluster model as the "global model" in this local subroutine. In particular, in addition to the original self-supervised learning loss L_{SSL} that takes in positives x^+ and potentially negatives x^-, we add a weighted auxiliary loss. The loss is defined as the cosine distance metric on the prediction output p of the local model on data point positives x^+, and the feature output z of the global model on data point x (similar to how SimSiam is implemented). This way, when there are many local updates without explicit communication among local data sources, the global model features can still *regularize* the local ones to be close to the global one.

Compared to other concurrent/contemporaneous methods, Zhuang et al. (2022) requires an extra memory bank and a customized update rule for local model, Makhija et al. (2022) requires access to an unlabeled public dataset for all models to measure the distance, and requires the communication of some datasets; He et al. (2021a) experiments with multiple methods to do personalization in decentralized learning, and our clustering-based approach can be viewed as new instance of it. In `FeatARC`, the auxiliary regularization loss and the clustering procedure between global and local models are simple to add and are general enough to be compatible with any SSL algorithm in decentralized n our experiment, we sweep over the hyperparameters and choose the balancing hyperparameter $\lambda = 1$ and the number of cluster to be $C = 2$.

D Real-world Decentralized Unlabeled Data Examples

In this section, we enumerate several real-world motivating scenarios where Dec-SSL with heterogeneous and unlabeled data is relevant and using unlabeled data can play a significant role (Figure 11). These examples are naturally related to fleets of devices, where model adaptation and data sharing become a central question, and thus require an efficient way to extract information from the decentralized datasets. Note that different from the "big and diverse data" motivation for SSL in the centralized setting, decentralized setting emphasizes that the data come from very distinct data sources, and the bandwidth in many cases simply cannot afford raw data communications.

Self-driving fleet. Self-driving cars are naturally deployed around the world with very distinct data distributions. For instance, the traffic rules in Berlin can be very different from the traffic rules

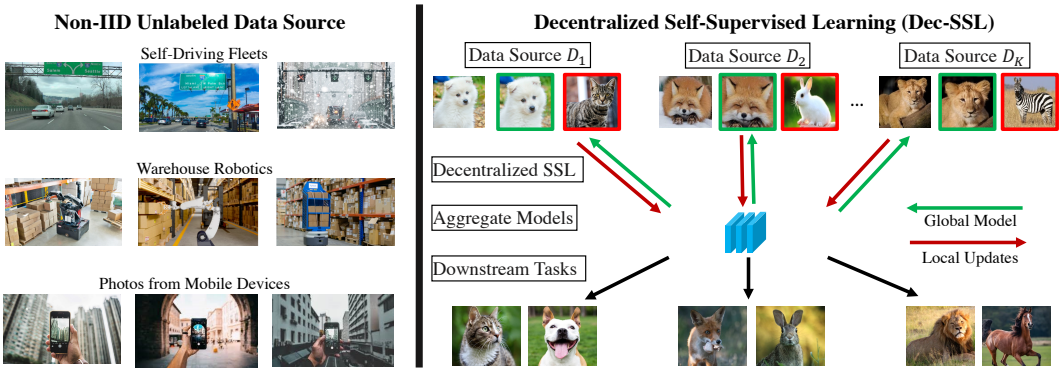

Figure 11: **Decentralized self-supervised learning (Dec-SSL).** In the real world, large amounts of unlabeled data are generated and stored in a distributed fashion with high heterogeneity. In this work, we study decentralized self-supervised learning and apply it to real-world visual representation learning problems.

in China. The camera observations on a freeway is very different from those on a crowded city road. Despite that data sharing might not be a problem, labeling all masks for images can be a prohibitive tasks and sharing all data can be very inefficient. The data is inherently skewed in terms of quantity n, features x, as well as labels y. For instance, we can have imbalanced number of classes for an object detector trained to deploy on the freeway that often sees trucks and one trained to deploy on the street that often sees people and cyclists.

Mobile edge devices. Decentralized supervised learning on the edge devices such as medical diagnosis, object detection, and sentence completion have been used in the real systems. However, with growing interests and importance, decision making and interactions with the environment in the wild are more likely to generate unlabeled datasets. With external sensors, one can collect data for agents participating in some tasks such as cooking, doing sports, and working, but we cannot easily provide labels for these settings and these settings can sometimes be privacy sensitive. Take cooking for instance, it can be very difficult to label the masks for all the ingredients and food on the table. Moreover, the data from only each single user might not be enough to learn a generalizable representations through self-supervised learning, thus motivating each user to join a federation, and jointly learn a global model. Thus, it would be very interesting for the community to investigate decentralized self-supervised learning to acquire useful representation from these by nature distributed and diverse data.

Warehouse/Household robots. A bottleneck in robotics has been the availability of high-quality and large-scale real-world data. As robotic systems are deployed more and more at scale in both warehouse and households settings, large-scale datasets are becoming increasingly available. In Section §D.1, we present a detailed example from the actual Amazon warehouse to motivate decentralized self-supervised learning. Similar to self-driving cars, a single robotic work-cell can generate millions of images per year; however, it is impractical to label data at this scale. Moreover, each local data distribution can be narrow and thus the model learned from each local dataset can hardly generalize. Considering a model trained on data from a warehouse that only sees boxes and then trying to operate this model in a warehouse that sees a variety of package types. To address the overfitting issue, it is useful to learn a *common representation* that can be quickly specialized for each local data source. Our Dec-SSL framework provides an efficient and robust way to do representation learning. In addition, due to the communication budget, it is desirable to have longer local updates E and arbitrary participation ratio ρ during the learning process. These methods, taken to full fruition, can enable local systems to efficiently share information and continually improve with significantly fewer labels. Similarly, a fleet of home robots that are deployed at diverse homes across the world can generate terabytes of raw data that are infeasible to share on cloud databases, due to limits on both the privacy and the network bandwidth. Moreover, the data that is commonplace for robots at one place can be out-of-distribution for robots at other places, causing challenges on deploying robots in homes, warehouses, and other human environments. In the next section, we provide more details about the robotics dataset example from Amazon warehouse.

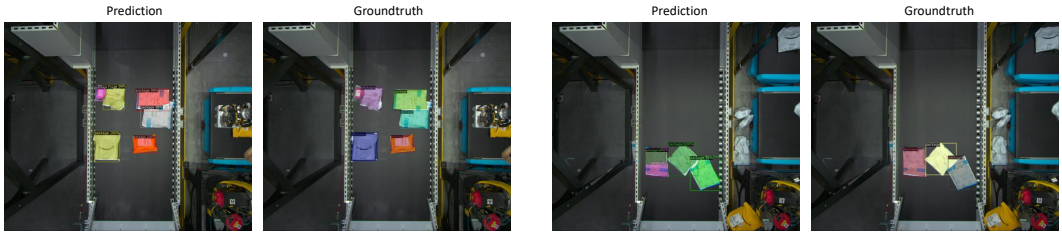

Figure 12: **Qualitative instance segmentation results on the real world Amazon Data.** We applied our decentralized self-supervised learning (Dec-SSL) framework to the real-world data collected in the Amazon warehouses. For each pair of images, the left shows the instance segmentation results of Dec-SSL using a backbone pretrained on the same data, while the right shows the predictions from the Amazon system (used as the ground-truth labels for finetuning). Our method achieves a decent result and outperforms the baseline trained from scratch.

D.1 A REAL-WORLD NON-IID DATASET IN ROBOTICS

Robin is a robotic manipulation work-cell at Amazon designed to induct packages into a sortation system. Packages are fed to the robot by means of a conveyor belt and other up-stream material handling equipment. An advanced sensing and perception system on Robin acquires images of the scene, detects and segments packages, and determines what package to pick and how. A custom End of Arm Tool (EoAT) and motion planning and control software robustly execute the pick and place the package on an outbound drive unit. The large-scale deployment of Robin in production provides millions of visual and interaction data. These data are largely unlabeled. As deployments of systems like Robin scale, centrally aggregating data from the entire fleet becomes costly if not infeasible due to bandwidth limits. An additional challenge to continual learning on Robin is that distributions shift at both the individual work-cell level as well as the facility (or site) level, and these shifts present trade-offs in generalization vs. specialization. Said differently, its not clear simply pooling the data is advantageous. The following are some notable ways the Robin dataset is diverse along with several factors that drive this diversity.

Package mix. Robin handles many different types of parcels; for example, cardboard boxes, paper bags, poly bags, jiffy mailers, items shipped in their own packaging, etc. A particular facility may see a particular distribution of package types based on its purpose in the network. For example, many sites handle a diversity of package types, weights, and sizes; whereas, other sites may handle predominately only one or two package types or have restrictions based on weight or size. There are also temporal factors that produce shifts in package type distribution. For example, the introduction of recyclable materials or the use of less packaging material over time.

Package density. As is notable in Figure 12, the density of package presentation varies between facilities and over time. Different sites may have different up-stream material handling systems (e.g. conveyance) that feed Robin packages in different ways. On one extreme, scenes can consist of a single package, and on the other extreme packages are presented in a dense pile with significant overlap and occlusion. During certain times of year (e.g., holiday season) there is increased volume in the network and this can produce denser scenes.

Hardware configuration. Robin work-cells can vary in arm, EoAT, and sensor types throughout the network. Additionally, the size and types of collision geometry in the work-cell area can change at both the work-cell and facility level. These differences mean that even if the input distribution (scenes) are the same, robots may see and move in different ways to accomplish the same task. This further contributes to diversity in both visual and interaction data.

E THEORETICAL ANALYSIS

Notation. For any positive integer k, we use $[k]$ to denote the set $\{1, 2, \cdots, k\}$. We use $\mathcal{N}(\mu, \Sigma)$ to denote the Gaussian distribution with mean μ and covariance matrix Σ. We use $\langle x, y \rangle$ to denote the inner product of two vectors $x, y \in \mathbb{R}^d$. For two non-negative integers k, n, we use $k \bmod n$ to denote the arithmetic remainder of k divided by n. For positive integer $d > 0$, we use e_i with $i \in [d]$

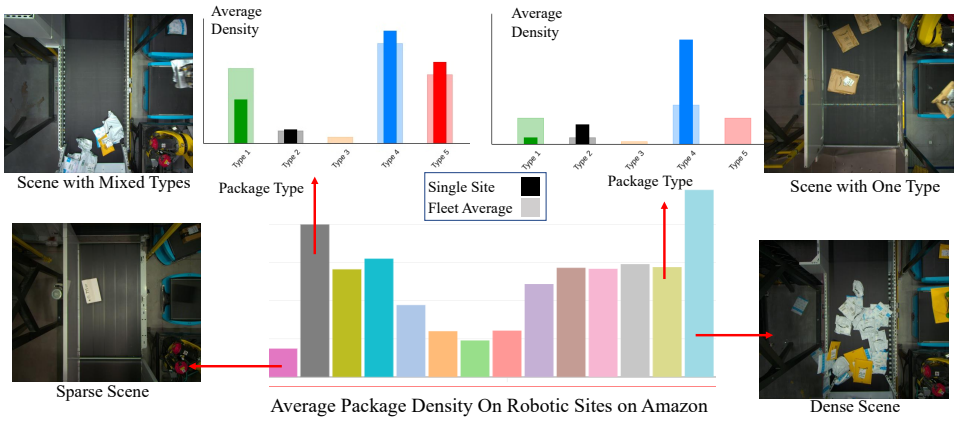

Figure 13: **Heterogeneity naturally occurs in real-world unlabeled data.** In real-world robotic settings such as those in Amazon warehouses, data distribution shifts emerge as a result of differences in location and type of facility, time of year (e.g. holidays), upstream material handling systems, and robotic work-cell configuration (e.g., arm, gripper, and sensor types), etc. The plot shows two axis of non-IIDness across different sites in Amazon: the density of the packages among scenes and the package type distribution within scene compared to the average.

to denote the d basis vectors in $\mathbb{R}^d$ Euclidean space. For a real $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ and $\lceil x \rceil$ to denote the floor and ceiling integers of x, i.e., $\lfloor x \rfloor = \max\{k \in \mathbb{Z} \,|\, k \le x\}$ and $\lceil x \rceil = \min\{k \in \mathbb{Z} \,|\, x \le k\}$.

In this section, we aim to shed some lights on the robustness of decentralized SSL approaches to data heterogeneity, and their comparison to decentralized supervised learning.

E.1 Deferred details and proof in Section §3.2

Setup. Consider a decentralized SSL problem with K data sources. To model non-IIDness across them, we use a common type of data heterogeneity, i.e., the label heterogeneity (see our discussions in §C.2). Indeed, label heterogeneity has been recognized as a fundamental and pervasive problem for decentralized learning, causing significant performance loss across many applications (Hsieh et al., 2020). This setting also corresponds to one non-IIDness we used in previous subsections (see e.g., §3.1). Similar to the SimSiam approach (Chen et al., 2020), we first augment x, an anchor sample from the dataset to have two positive samples, by sampling $\xi, \xi' \sim \mathcal{N}(0, I)$ IID from the Gaussian distribution. Consider the linear embedding function $f_w(x) = wx$, where $w \in \mathbb{R}^{m \times d}$ and $m \ge 2K$. The local SSL objective for data source k is given by

$$\mathcal{L}_k(w) := -\widehat{\mathbb{E}}\big[(w(x_{k,i} + \xi_{k,i}))^\top (w(x_{k,i} + \xi'_{k,i}))\big] + \frac{1}{2}\|w^\top w\|_F^2, \tag{E.1}$$

where $\widehat{\mathbb{E}}$ is taken expectation over the dataset $x \sim D_k$, and the randomness of $\xi_{k,i}$ and $\xi'_{k,i}$. Moreover, recall the global objective is given in (2.2). Note that (E.1) instantiates the SimSiam loss with the negative inner-product $\langle a, b \rangle$ as the distance function $\mathbb{D}(a, b)$ and no feature predictor for simplicity. We also add a regularization term $\|w^\top w\|_F^2/2$ to improve the mathematical tractability of the objective (which in practice corresponds to the weight decay method in the optimizer). The K data sources collaboratively minimize (2.2), and evaluate the learned representation on a $2K$-way classification task.

Data heterogeneity. Choose $K = \Theta(d^{1/20})$. The K local datasets are generated as follows. For a fixed data source k, the labels are skewed in that data from classes $2k-1$ and $2k$ constitute the majority of the data, while other classes are rare, or even unseen. Specifically, let $e_1, \cdots, e_d$ denote the standard unit basis of $\mathbb{R}^d$, and let $n_{k,j}$ for $j \in [2K]$ denote the number of data for class $j \in [2K]$ in this dataset k. For class $2k-1$, data is generated following $x^{(2k-1)} = e_k - \sum_{i \ne k, i=1}^{K} q^{(2k-1,i)} \tau e_i + \mu \xi^{(2k-1)}$, where $q^{(2k-1,i)}$ are sampled uniformly from $\{0, 1\}$, $\xi^{(2k-1)} \sim \mathcal{N}(0, I)$, and both $\tau = d^{1/5}$ and $\mu = d^{-1/5}$ are positive hyperparameters. Similarly, for class $2k$, $x^{(2k)} = -e_k - \sum_{i \ne k, i=1}^{K} q^{(2k,i)} \tau e_i + \mu \xi^{(2k)}$. The amounts of data from classes $2k-1$ and $2k$ are equal and both of order $\mathtt{poly}(d)$. For classes $2i-1$ with $i \ne k$, $x^{(2i-1)} = e_i + \mu \xi^{(2i-1)}$, and there

is no data for classes $2i$ in data source k. The amounts of data in classes $2i-1$ with $i \neq k$ are the same and of order sublinear in d, i.e., $O(d^\beta)$ for some $\beta \in (0,1)$, such that $O(Kd^\beta) \leq O(d^{1/5})$. Note that this leads to that $O(Kd^\beta/(2n_{k,2k})) \leq O(d^{-4/5}) \leq O(1)$, and implies that the sum of the data from all the infrequent classes $2i-1$ and $2i$ for $i \neq k$ are less than the data in the frequent classes $2k-1$ and $2k$. All K local datasets are assumed to contain the same amount of data, i.e., $|D_1| = |D_2| = \cdots = |D_K|$. We visualize the heterogeneous data distribution in Figure 3.

Proof of Theorem 3.2:

For local dataset k. We first analyze the solution to minimizing the local objective (E.1), using only local dataset D_k. Define

$$X_k := \widehat{\mathbb{E}}_{x \sim D_k}(xx^\top) = \frac{1}{|D_k|}\sum_{i=1}^{|D_k|} x_{k,i}x_{k,i}^\top$$

to be the empirical data covariance matrix for dataset k. Notice that

$$\begin{aligned}
&\mathbb{E}(X_k) \qquad\qquad (E.2)\\
&= \mathrm{diag}\Big(\underbrace{\tau^2 + O(d^{-2/5}),\ \tau^2 + O(d^{-2/5}), \cdots,\ \underbrace{1 + O(d^{-2/5})}_{k\text{-th term}}, \cdots, \tau^2 + O(d^{-2/5}),}_{K \text{ terms}}\\
&\qquad \underbrace{O(d^{-2/5}), \cdots, O(d^{-2/5})}_{d-K \text{ terms}} \Big) \qquad\qquad (E.3)\\
&= \mathrm{diag}\Big(d^{2/5} + O(d^{-2/5}), \cdots,\ 1 + O(d^{-2/5}),\ \cdots,\ d^{2/5} + O(d^{-2/5}),\ O(d^{-2/5}), \cdots, O(d^{-2/5}) \Big).
\end{aligned}$$

By matrix concentration bounds, e.g., (Vershynin, 2018), we have that with probability at least $1 - \frac{1}{2}e^{-d^{1/10}}$, $\|X_k - \mathbb{E}(X_k)\| \leq O(d^{-2/5})$. By Weyl's inequality we have that with high probability,

$$|\lambda_{k,i} - \lambda_i(\mathbb{E}(X_k))| \leq \|X_k - \mathbb{E}(X_k)\|_2 \leq O(d^{-2/5}) \qquad (E.4)$$

for all $i \in [d]$, where we denote $\lambda_{k,i} := \lambda_i(X_k)$ as the i-th largest eigenvalue of X_k.

On the other hand, as $|D_k| \geq \mathtt{poly}(d)$, for any e_j with $j \in [K] \setminus \{k\}$, we have that with probability at least $1 - \frac{1}{2}e^{-d^{1/10}}$, at least $1/3$ (where at least $2/3$ data come from classes $2k-1$ or $2k$, and $1/2$ of them) satisfy that either $q^{(2k-1,j)}$ or $q^{(2k,j)}$ is 1, and $\sum_{i=1}^{|D_k|} |\langle \xi_{k,i}, e_k \rangle| / |D_k| \leq O(d^{1/10})$[2] (see (Liu et al., 2021, Lemma E.1)). Hence, we have that

$$e_j^\top X_k e_j = \widehat{\mathbb{E}}_{x \sim D_k}\big[(e_j^\top x)^2\big] \geq \big[\widehat{\mathbb{E}}_{x \sim D_k}(e_j^\top x)\big]^2 \geq \Big(\frac{1}{3}\tau - \mu \sum_{i=1}^{|D_k|} \frac{1}{|D_k|}\big|e_j^\top \xi_{k,i}\big|\Big)^2 = \Omega(\tau^2) = \Omega(d^{2/5}), \qquad (E.5)$$

with probability at least $1 - \frac{1}{2}e^{-d^{1/10}}$, where we use the fact that $\mu = d^{-1/5}$.

Now notice that the local objective in (E.1) can be equivalently re-written as

$$\min_w \ \|X_k - w^\top w\|_F^2, \qquad (E.6)$$

which, by Eckart-Young-Mirsky theorem (Eckart & Young, 1936), yields that the span of the rows of optimal w (an $m \times d$ matrix) is the span of the eigenvectors of the first m eigenvalues of X_k. Let $\{v_{k,1}, \cdots, v_{k,d}\}$ denote the set of d orthonormal eigenvectors of X_k, then $X_k = \sum_{i=1}^d \lambda_{k,i} v_{k,i} v_{k,i}^\top$, where recall that $\lambda_{k,i}$ is the i-th largest eigenvalue of X_k. Hence, by (E.5), we have

$$\lambda_{k,1} \sum_{i=1}^d (e_j^\top v_{k,i})^2 \geq e_j^\top X_k e_j = \sum_{i=1}^d \lambda_{k,i}(e_j^\top v_{k,i})^2 \geq \Omega(d^{2/5}) \qquad (E.7)$$

[2]Note that we here slightly abuse the notation by denoting the noise in generating the data point $x_{k,i}$ by $\xi_{k,i}$, which should not be confused with the augmentation noise in the SSL objective (E.1).

with high probability. In fact, by (E.4) and (E.2), we can have finer bounds of $e_j^\top X_k e_j$ as

$$\begin{aligned} e_j^\top X_k e_j &= e_j^\top \mathbb{E}(X_k) e_j + e_j^\top \big[X_k - \mathbb{E}(X_k)\big] e_j \\ &\geq d^{2/5} + O(d^{-2/5}) - \|X_k - \mathbb{E}(X_k)\| \geq d^{2/5} - O(d^{-2/5}), \end{aligned} \tag{E.8}$$

$$\begin{aligned} e_j^\top X_k e_j &= e_j^\top \mathbb{E}(X_k) e_j + e_j^\top \big[X_k - \mathbb{E}(X_k)\big] e_j \\ &\leq d^{2/5} + O(d^{-2/5}) + \|X_k - \mathbb{E}(X_k)\| \leq d^{2/5} + O(d^{-2/5}), \end{aligned} \tag{E.9}$$

where we use the fact that $\|X\|_{\max} \leq \|X\|$ for symmetric X.

Furthermore, by (E.4) and (E.2), we know that

$$\begin{aligned} d^{2/5} - O(d^{-2/5}) &\leq \lambda_1(\mathbb{E}(X_k)) - O(d^{-2/5}) \leq \lambda_{k,1} \\ &\leq \lambda_1(\mathbb{E}(X_k)) + O(d^{-2/5}) = d^{2/5} + O(d^{-2/5}) \end{aligned} \tag{E.10}$$

showing that $\lambda_{k,1} = d^{2/5} \pm O(d^{-2/5})$. Combining (E.8) and (E.10), we obtain that

$$\sum_{i=1}^{d} (e_j^\top v_{k,i})^2 \geq \frac{d^{2/5} - O(d^{-2/5})}{d^{2/5} + O(d^{-2/5})} \geq 1 - O(d^{-4/5}), \tag{E.11}$$

which completes the proof with $\alpha = 4/5$.

For global dataset. Recall the global objective given in (2.2):

$$\min_w \sum_{k\in[K]} \frac{|D_k|}{|D|} \mathcal{L}_k(w). \tag{E.12}$$

As the local objective in (E.1) can be equivalently re-written as (E.6), we can also re-write the global objective as

$$\min_w \; g(w) := \sum_{k\in[K]} \frac{|D_k|}{|D|} \|X_k - w^\top w\|_F^2. \tag{E.13}$$

Further, note that the gradient of $g(w)$ in (E.13) at any w is the same as that of the following objective:

$$\widetilde{g}(w) := \Big\| \underbrace{\sum_{k\in[K]} \frac{|D_k|}{|D|} X_k}_{\bar{X}} - w^\top w \Big\|_F^2. \tag{E.14}$$

Thus, these two objectives share the same minimizer. Note that it is the minimizer that we care about (as it determines the feature mapping), and minimizing (E.14) is equivalent to minimizing the SSL objective over the global dataset $D = \bigcup_{k\in[K]} D_k$, with the empirical data covariance matrix

$$\bar{X} := \widehat{\mathbb{E}}_{x\sim D}(xx^\top) = \frac{|D_k|}{|D|} \sum_{k\in[K]} \frac{1}{|D_k|} \sum_{i=1}^{|D_k|} x_{k,i} x_{k,i}^\top = \frac{1}{|D|} \sum_{i=1}^{|D|} x_i x_i^\top. \tag{E.15}$$

Hence, (E.13) is equivalent to solving

$$\min_w \; \widetilde{g}(w) = \|\bar{X} - w^\top w\|_F^2. \tag{E.16}$$

We can now follow the analysis above. First, by (E.2) and the linearity of expectation, we have

$$\begin{aligned} &\mathbb{E}(\bar{X}) \\ &= \operatorname{diag}\Big(\underbrace{d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20}), \cdots, d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20})}_{K \text{ terms}}, O(d^{-2/5}), \cdots, O(d^{-2/5}) \Big), \end{aligned} \tag{E.17}$$

where we have used the fact that

$$\frac{(K-1)\cdot d^{2/5} + 1}{K} = (1 - \Theta(d^{-1/20}))\cdot d^{2/5} + O(d^{-1/20}) = d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20}).$$

Then, by similar arguments from (E.4)-(E.9), we have that for all $j \in [K]$ (without excluding any k),

$$e_j^\top \bar{X} e_j \geq d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20}) - O(d^{-2/5}), \tag{E.18}$$

$$\lambda_1(\bar{X}) \leq \lambda_1(\mathbb{E}(\bar{X})) + O(d^{-2/5}) = d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20}) \tag{E.19}$$

leading to that

$$\sum_{i=1}^{d} (e_j^\top \bar{v}_i)^2 \geq \frac{d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20}) - O(d^{-2/5})}{d^{2/5} - \Theta(d^{7/20}) + O(d^{-1/20})} \geq 1 - 2 \cdot O(d^{-4/5}),$$

for large enough d such that $1 - O(d^{-1/20}) \geq 1/2$, where $\{\bar{v}_1, \cdots, \bar{v}_d\}$ denote the d orthonormal eigenvectors of $\bar{X}$. This completes the proof. □

E.2 Deferred results and proof in Section §4

Setup. The data are generated as in §E.1. For each local dataset k, consider a supervised learning algorithm that uses a two-layer linear network $g_{u_k,v_k}(x) := v_k u_k x$ as classifier, where $u_k \in \mathbb{R}^{m \times d}$ and $v_k \in \mathbb{R}^{c \times m}$ for some $m \geq c = 2K$ are weight matrices. Note that $u_k x$ can be viewed as the feature learned by this classifier, which can be used in the downstream tasks. This is exactly the protocol of **Dec-SLRep** on the local objective. Following Liu et al. (2021), we consider the approach of learning the network with minimal norm $\|(u_k)^\top u_k\|_F^2 + \|(v_k)^\top v_k\|_F^2$ subject to the margin constraint that $[g_{u_k,v_k}(x)]_y \geq [g_{u_k,v_k}(x)]_{y'} + 1$ for all data (x, y) in the local dataset k with all $y' \neq y$. Note that such a solution can be found in direction via gradient descent using logistic loss (Ji & Telgarsky, 2018). Now we are ready to show the following result, based on the techniques in Liu et al. (2021).

Proposition E.1 (Representations learned by Dec-SLRep across heterogeneous data sources)**.** With high probability, the feature matrix $u_k = [u_{k,1}, \cdots, u_{k,m}]^\top \in \mathbb{R}^{m \times d}$ learned from the local dataset D_k has the following properties:

$$\sum_{i=1}^{m} \langle u_{k,i}, e_j \rangle^2 \leq O(d^{-\frac{1}{10}}),$$

for $j \in [K] \setminus \{k\}$; while

$$\sum_{i=1}^{m} \langle u_{k,i}, e_k \rangle^2 \geq 1 - O(d^{-\frac{1}{20}}).$$

In other words, the correlation between the learned features in w_k and e_j is small for all $j \in [K] \setminus \{k\}$, while the correlation between the features and e_k is large.

The proposition suggests that the learned features for each local dataset k overfit the its skewed data, and does not learn the feature directions, e.g., other unit vector directions e_j for $j \in [K]$ and $j \neq k$, that might generalize well to the data in other datasets. The result can be viewed as a multi-class generalization of the first part of Theorem 3.1 in Liu et al. (2021) . The intuition is also illustrated in Figure 3. This way, the feature space learned from various local datasets differ significantly, in that most of the directions among $\{e_1, \cdots, e_K\}$ are uninformative, while their possibly informative feature directions are all different. This heterogeneity between local solutions is not in favor of *local updates*, as too many local updates would drift the iterates towards its local optimum, and the iterates would become too far away from each other, hurting the convergence of classic decentralized learning algorithms as `FedAvg`. Hence, compared with the Dec-SSL case and Theorem 3.2, Dec-SLRep can be less communication-efficient as it does not allow large number of local updates.

Proof of Proposition E.1. Without loss of generality, we show the result for dataset D_1, i.e., when $k = 1$. The proof follows mostly from the proof of Theorem 3.1 in Liu et al. (2021), and for conciseness, we only layout the key differences. For convenience, we remove the index k in the notation whenever it is clear from the context. First, note that the local SL problem is equivalent to the following one:

$$\min_{w} \sum_{i=1}^{c} \|\widetilde{w}_i\|_2^2 \quad s.t. \quad \langle \widetilde{w}_y, x \rangle \geq \langle \widetilde{w}_{y'}, x \rangle + 1, \ \forall (x, y) \in D_1, \quad y' \in [2K], \ y' \neq y, \tag{E.20}$$

where $\widetilde{w} = [\widetilde{w}_1, \cdots, \widetilde{w}_c]^\top$. We then establish the following lemma.

Lemma E.2 (Margin & norm bounds)**.** Given the data generated above. Construct a solution to (E.20) as $w_1^* = e_1, w_2^* = -e_1$, and for $i \in \{2, \cdots, K\}$, $w_{2i-1}^* = \frac{1}{\mu d}\sum_{j=1}^{n_{2i-1}} \xi_j^{(2i-1)}$ and $w_{2i}^* = \mathbf{0}$. Then we have that for large enough d, with probability at least $1 - e^{-d^{1/10}}$, the margin of $\{w_1^*, \cdots, w_{2K}^*\}$ is at least $1 - O(d^{-1/10})$. Moreover, we have $\|w_j^*\|_2^2 \le O(d^{-3/10})$ for $j \in [2K] \setminus \{1, 2\}$.

Proof sketch. The proof follows from the proof of Lemma E.2 in Liu et al. (2021). The argument for the data in classes 1 and 2 is similar; the argument for that in classes $2i-1$ for $i \in \{2, \cdots, K\}$ is similar to that for class 3 in the proof therein. Note that the total number of data in the rare classes here equals that of the rare class 3 therein, which is $O(d^{1/5})$. So one needs to replace the n_3 therein by $O(d^{1/5})/K = O(d^{3/20})$ (recall that $K = \Theta(d^{1/20})$), which is a smaller number that validates the arguments in the proof therein, and in fact, makes the norm of $\|w_j^*\|_2$ smaller, i.e., $\|w_j^*\|_2 \le O(d^{-3/20})$. Also, note that there is no margin constraints corresponding to classes $2i$ with $i \in \{2, \cdots, K\}$, as there is no data belong to these classes in this local dataset. Finally, note that for any data (x, y) in the dataset, $x^\top w_{2i}^* = 0$, which does not affect the margin between other classes and $2i$. The remaining of the proof follows from the proof therein. □

Then, similar to the argument in the proof of Theorem 3.1 in Liu et al. (2021) (supervised learning part), one can show that by normalizing the solution in Lemma E.2 by its margin, denoted by $\alpha \ge 1 - O(d^{-1/10})$, the solution to (E.20) (which should have no-larger norm) satisfies

$$\sum_{i=1}^{2K} \|\widetilde{w}_i\|_2^2 \le \sum_{i=1}^{2K} \left\|\frac{w_i^*}{\alpha}\right\|_2^2 = \frac{2 + (2K-2)\cdot O(d^{-3/10})}{\alpha^2} \le 2 + O(d^{-\frac{1}{10}}). \tag{E.21}$$

On the other hand, continue to follow the argument of Eq. (21)-(28) in the proof of Theorem 3.1 in Liu et al. (2021), we know that for any $\ell \in [2K] \setminus \{1, 2\}$,

$$\langle \widetilde{w}_1, e_1\rangle^2 + \langle \widetilde{w}_2, e_1\rangle^2 + \langle \widetilde{w}_\ell, e_1\rangle^2 \ge 2 - O(d^{-1/10}). \tag{E.22}$$

Note that this also implies

$$\sum_{i=1}^{2K} \|\widetilde{w}_i\|_2^2 \ge 2 - O(d^{-1/10}). \tag{E.23}$$

By (E.21) and (E.22), we know that

$$\sum_{j=2}^{d} \left(\langle \widetilde{w}_1, e_j\rangle^2 + \langle \widetilde{w}_2, e_j\rangle^2 + \langle \widetilde{w}_\ell, e_j\rangle^2 \right) \le O(d^{-1/10})$$

which further leads to the fact that

$$\sum_{j=2}^{d} \left(\langle \widetilde{w}_1, e_j\rangle^2 + \langle \widetilde{w}_2, e_j\rangle^2 + \sum_{\ell \in [2K]\setminus\{1,2\}} \langle \widetilde{w}_\ell, e_j\rangle^2 \right) \tag{E.24}$$

$$\le \sum_{\ell \in [2K]\setminus\{1,2\}} \sum_{j=2}^{d} \left(\langle \widetilde{w}_1, e_j\rangle^2 + \langle \widetilde{w}_2, e_j\rangle^2 + \langle \widetilde{w}_\ell, e_j\rangle^2 \right) \le 2K \cdot O(d^{-1/10}) \le O(d^{-1/20}).$$

The rest of the proof follows that of Theorem 3.1 in Liu et al. (2021), with the number of classes 3 therein being replaced by $c = 2K$ (as Lemma E.3 in Liu et al. (2021) still holds). By applying the argument therein for all e_j with $j = 2, \cdots, d$, we have

$$\sum_{j=2}^{d}\sum_{i=1}^{m} \langle u_i, e_j\rangle^2 \le \sum_{j=2}^{d} \sum_{\ell \in [2K]} \langle \widetilde{w}_\ell, e_j\rangle^2 \le O(d^{-\frac{1}{20}}). \tag{E.25}$$

Furthermore, notice that by Lemma E.3 in Liu et al. (2021), $u(u)^\top = (v)^\top v$ at the solution and $\|\widetilde{w}\|_F^2 = 2 \cdot \|u(u)^\top\|_F^2$. Hence

$$\left(\sum_{i=1}^{m} \|u_i\|_2^2\right)^2 = \|u\|_F^4 \ge \|u(u)^\top\|_F^2 = \|\widetilde{w}\|_F^2/2 \ge 1 - O(d^{-1/10}),$$

where the last inequality uses (E.23). This leads to the final result that

$$\sum_{i=1}^{m}\langle u_i, e_1\rangle^2 \geq \sum_{i=1}^{m}\|u_i\|_2^2 - O(d^{-\frac{1}{20}}) \geq 1 - O(d^{-1/20}),$$

where we use $K = \Theta(d^{1/20})$ and (E.25). Note that the proof above also holds for other dataset $k \neq 1$. This completes the proof. □