Information Dissimilarity Measures in Decentralized Knowledge Distillation: A Comparative Analysis

Mbasa Joaquim Molo[1,2], Lucia Vadicamo[2], Emanuele Carlini[2], Claudio Gennaro[2], and Richard Connor[3]

[1] Department of Computer Science, University of Pisa, Pisa, Italy, joaquim.molo@phd.unipi.it,
[2] Institute of Information Science and Technologies, CNR, Via G.Moruzzi 1, 56124, Pisa, Italy {lucia.vadicamo,emanuele.carlini,claudio.gennaro}@isti.cnr.it
[3] School of Computer Science, University of St Andrews, St Andrews, KY16 9SS, Scotland rchc@st-andrews.ac.uk

Abstract. Knowledge distillation (KD) is a key technique for transferring knowledge from a large, complex "teacher" model to a smaller, more efficient "student" model. Although initially developed for model compression, it has found applications across various domains due to the benefits of its knowledge transfer mechanism. While Cross Entropy (CE) and Kullback-Leibler (KL) are commonly used in KD, this work investigates the applicability of loss functions based on underexplored information dissimilarity measures, such as Triangular Divergence (TD), Structural Entropic Distance (SED), and Jensen-Shannon Divergence (JS), for both independent and identically distributed (iid) and non-iid data distributions. The primary contributions of this study include an empirical evaluation of these dissimilarity measures within a decentralized learning context, i.e., where independent clients collaborate without a central server coordinating the learning process. Additionally, the paper assesses the performance of clients by comparing pairwise distillation averaging among clients to conventional peer-to-peer pairwise distillation. Results indicate that while dissimilarity measures perform comparably in iid settings, non-iid distributions favor SED and JS, which also demonstrated consistent performance across clients.

Keywords: Information dissimilarity measure · Divergence Function · Knowledge Distillation · Distributed intelligence.

1 Introduction

The integration of Artificial Intelligence in edge processing has led to the emergence of an interdisciplinary field known as *Distributed Intelligence* or *Edge Intelligence*, which aims to develop systems composed of software agents, robots, sensors, and computer systems that can collaborate effectively [22,23,15]. In this field, *Knowledge distillation* (KD) has been employed to facilitate knowledge transfer between edge devices, enhancing the development of more efficient and accurate models [28]. KD is a machine learning technique designed to transfer knowledge from a large, complex model (the *teacher*) to a smaller, more efficient one (the *student*) [10,8,7]. In addition to its

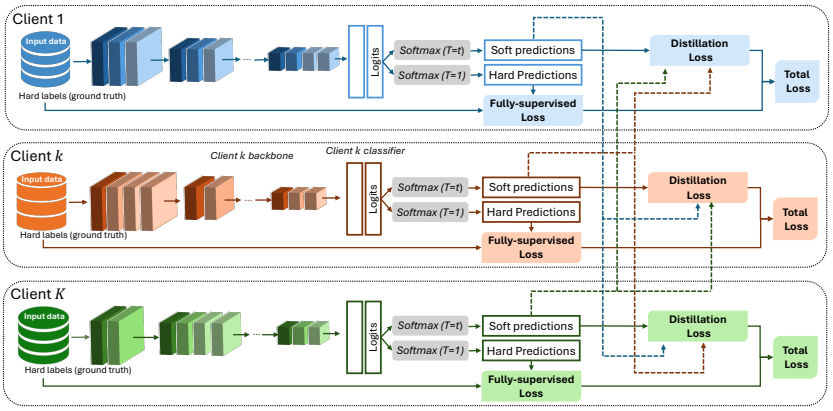


Fig. 1: KD-based decentralized network consisting of K clients, where distillation is performed using soft predictions for effective knowledge transfer.

primary role in model compression, it has started to find applications in other areas, including distributed intelligence [3] and continual learning [4].

In KD-based distributed learning framework, *clients* exchange information to enhance their learning process, where each client operates both as learner and source of knowledge for other clients. These clients are part of a decentralized system where no single model acts as the central teacher. Instead, each client trains on its local dataset and shares knowledge with others. As illustrated in Fig. 1, this information exchange is achieved through a combination of two types of losses. The first loss component, indicated as "fully-supervised loss", is usually the cross-entropy (CE) with "hard" targets derived by the ground-truth labels of the input samples. The second component is the "distillation loss" designed to ensure that each learning client mimics the output of other remote clients [25]. This loss is typically implemented by comparing the probability distributions of the models involved, where one model acts as the student and others take turns serving as teachers. This encourages the student's output probabilities to closely match those of the teacher. The model's output probabilities are typically computed using a softmax layer. Adjusting the softmax temperature during training has proven to be crucial in metric learning and distillation processes. In the context of distributed intelligence, this technique is also employed to generate soft predictions for effective distillation. Hence, the distillation loss is expressed as minimizing the gap between the soft predictions of one client with respect to the soft predictions of all other clients [1,2,27].

Given that the softmax function transforms an array of logits into an array of positive values summing to 1, various information dissimilarity measures can theoretically be used to implement the distillation loss. However, in practice, it is predominantly realized using CE, in addition to Kullback-Leibler (KL) Divergence, and Mean Squared Error (MSE) [13]. These methods have been extensively studied and proven effective for knowledge transfer in diverse machine learning tasks, while a wide range of information distance functions remain unexplored in the literature related to distributed learning.

In this work, we break new ground by investigating alternative dissimilarity measures – specifically, Triangular Divergence (TD), Structural Entropic Distance (SED), and Jensen-Shannon (JS) divergence – in the context of KD for decentralized learning scenarios. Recently, the correlations among these measures and the commonly used CE have been examined in [6] for independent and identically distributed (iid) data. Our work aims to expand the understanding of how these dissimilarity measures can enhance KD techniques, particularly in settings where data distribution may vary across learning clients (with a non-iid data distribution). To the best of our knowledge, our study is the first to empirically evaluate the effectiveness of TD, SED, and JSD for KD in a decentralized learning framework, offering novel insights and expanding the potential of KD applications beyond conventional CE and KL-based approaches. Our main contributions include designing a distributed KD environment suitable for investigating the aforementioned information dissimilarity measures and examining the performance of a set of clients by comparing pairwise distillation averaging among clients to the conventional peer-to-peer pairwise distillation, considering the various information dissimilarity measures.

The rest of this article is structured as follows. Section 2 provides background and related works on knowledge distillation and the dissimilarity measures utilized. Section 3 details the fully decentralized learning model employed in our study. Section 4 presents our experimental setup, while Section 5 discusses the results. Section 6 provides the conclusions.

2 Background and Related Works

2.1 Information Dissimilarity Measures and Statistical Divergences

Information distance refers to a measure that quantifies the dissimilarity between two sources of information (e.g., two finite objects). This concept is distinct but also related to statistical divergences, which quantify the dissimilarity between two probability distributions. For example, some information distances, including SED, can be used to compare also probabilities, while statistical divergence can be interpreted as information distances when the source of information are probability distributions. In the following, we provide formal definitions of the divergence functions and information distances used in this paper[4].

Kullback-Leibler Divergence. The KL divergence measures the difference between two probability distributions as the amount of information lost when one distribution is used to approximate the other. Given two distributions $\mathbf{q}$ and $\mathbf{p}$, it is defined as

$$KL(\mathbf{q} : \mathbf{p}) = \sum_{i=1}^{N} q_i \log \frac{q_i}{p_i} \tag{1}$$

[4]Please note that, as in [6], we use the delimiter ':' as argument separator of non-symmetric divergence instead of the double bar notation '‖' used in information theory.

Jensen-Shannon Divergence. The JS divergence, historically introduced in [26], is a "smoothed, symmetrised" version of KL divergence and can be interpreted as the total KL divergence relative to the average distribution $\frac{\mathbf{q}+\mathbf{p}}{2}$ [21]. In this paper, we use the following definition:

$$JS(\mathbf{q},\mathbf{p}) = \frac{1}{2}\left(KL\left(\mathbf{q} : \frac{\mathbf{q}+\mathbf{p}}{2}\right) + KL\left(\mathbf{p} : \frac{\mathbf{q}+\mathbf{p}}{2}\right)\right) \tag{2}$$

Structural Entropic Distance. SED [20] is an information-theoretic measure that compares the Shannon entropy H of two probability vectors with that of their arithmetic mean, where $H(\mathbf{p}) = -\sum_i^N p_i \ln p_i$ represents the amount of information needed to describe the probability vector $\mathbf{p} = [p_1, \ldots, p_N]$ [5]. Considering two probability vectors $\mathbf{p}$ and $\mathbf{q}$, SED can be calculated as the ratio of the complexity of the mean vector to the geometric mean of the complexities of individual vectors:

$$SED(\mathbf{q},\mathbf{p}) = \frac{C(\frac{\mathbf{q}+\mathbf{p}}{2})}{\sqrt{C(\mathbf{q})C(\mathbf{p})}} - 1 \tag{3}$$

where the complexity is computed as $C(\mathbf{p}) = b^{-\sum_{i=1}^N p_i \log_b p_i}$. The formulation in Eq. (3) gives an outcome in the range $[0,1]$, where 0 implies the two input vectors are identical, and 1 implies that they are orthogonal.

Triangular Divergence. The Triangular Divergence [5], also known as Triangular Discrimination [24] is defined as: $TD(\mathbf{q},\mathbf{p}) = \sum_{i=1}^N \frac{(q_i-p_i)^2}{q_i+p_i}$. Since the range of this function is $[0,2]$, in our work we use its scaled form:

$$TD(\mathbf{q},\mathbf{p}) = \tfrac{1}{2}\sum_{i=1}^N \frac{(q_i-p_i)^2}{q_i+p_i} = 1 - \sum_{i=1}^N \frac{2q_ip_i}{q_i+p_i} \tag{4}$$

where the formulation in the right part of Eq. (4) is an optimized version obtained observing that $(q_i-p_i)^2 = (q_i+p_i)^2 - 4q_ip_i$ and $\sum_{i=1}^N p_i = \sum_{i=1}^N q_i = 1$.

Cross Entropy. CE is a divergence measure widely used in machine learning to compare two probability distributions. It is defined as:

$$CE(\mathbf{q} : \mathbf{p}) = -\sum_{i=1}^n q_i \log p_i \tag{5}$$

It is worth noting that in the context of machine learning, as shown in [6], for spaces with certain properties, CE, KL, JS, TD shows very tight correlation. Specifically, if q is fixed, the perfect correlation between cross-entropy and Kullback-Leibler divergence is well-known and derives from simple algebra ($KL(\mathbf{q} : \mathbf{p}) = CE(\mathbf{q} : \mathbf{p}) - H(\mathbf{q})$) [6].

[5] Note that its square root is a metric, referred to as *Triangular Distance*, *Vincze-Le Cam* distance and the symmetric *chi-squared* distance [17]

Moreover, Jensen-Shannon correlates almost perfectly with triangular divergence in almost all high-dimensional spaces [24], while cross-entropy and triangular divergence are strongly correlated when the probabilities are obtained within the softmax function (Eq. (6)) with high temperature. Note that triangular is much cheaper calculation than cross-entropy and if the correlation is very strong the latter may be used instead.

2.2 Knowledge Distillation as the Teacher-Student Approach

Knowledge distillation was initially introduced to transfer knowledge from pre-trained teacher (large) networks to student (small) networks. This involves approximating the soft output or intermediate representation of teacher networks, aiming to derive a compact and faster model [16].

Concretely, for any input data x, the teacher network generates a vector of logits $\mathbf{z}(x) = [z_1(x), \ldots, z_N(x)]$ that are turned into a probability vector $\mathbf{p}(x) = [p_1(x), \ldots, p_N(x)]$ using the softmax function: $p_i(x) = \frac{e^{z_i(x)}}{\sum_j^N e^{z_i(x)}}$. Typically, neural networks produce probability distributions with sharp peaks, which might lack informativeness. To address this, Hinton et al. [10] proposed temperature scaling in the softmax to soften these probabilities:

$$p_i(x,T) = \frac{e^{z_i(x)/T}}{\sum_j^N e^{z_j(x)/T}}, \forall i \in \{1,\ldots,N\} \tag{6}$$

where T is a hyperparameter called temperature.

In KD, both the student and the teacher generate softened probability distributions, denoted as $\mathbf{p}_{\mathcal{S}}(x,T)$ and $\mathbf{p}_{\mathcal{T}}(x,T)$, respectively. The student's total loss is then defined as a linear combination of a supervised student loss $\mathcal{L}_{stu}$ and a knowledge distillation loss $\mathcal{L}_{KD}$:

$$\mathcal{L} = \alpha \mathcal{L}_{stu} + (1-\alpha)\mathcal{L}_{KD} \tag{7}$$

where $\alpha \in [0,1]$ is a hyperparemeter. Typically, $\mathcal{L}_{stu} = CE\left(\mathbf{y} : \mathbf{p}_{\mathcal{S}}(x, T=1)\right)$ and $\mathcal{L}_{KD} = CE\left(\mathbf{p}_{\mathcal{T}}(x, T=t) : \mathbf{p}_{\mathcal{S}}(x, T=t)\right)$, with $\mathbf{y}$ being the hard labels (ground-truth). Note that the distillation loss is expressed as minimizing the gap between the output representation of the teacher and the output representation of the student.

KD-based Distributed Learning. Recent research has explored KD for decentralized learning [28]. While much of this work focuses on a central teacher supervising student model training, there is a growing interest in fully decentralized settings where multiple clients collaborate to share knowledge without relying on a central authority.

Kim et al. [13] explored the role of the temperature hyperparameter in KD, showing higher temperature results in logit matching, which generally offers better generalization than label matching obtained with lower temperatures. They proposed employing MSE loss for direct logit matching. They showed that KL divergence loss stretches the second-to-last layer representations more than MSE loss and that KL divergence, especially with low temperature, is more resilient to noisy labels. Mishra et al. [18] developed EarlyLight, a method for training lightweight deep neural networks (DNNs) on edge devices using knowledge distillation from larger DNNs, considering also factors

Table 1: Summary of notation used

Notation	Description
T	Temperature in the softmax
N	Number of classes
$(\mathcal{G},\varepsilon)$	Network of clients. $\mathcal{G}$ is the set of nodes, ε is the set of edges
K, k	Number of clients, Index of current client
C^k	Current client
$D^k = (X^k, y^k)$	Local annotated dataset on client k. X^k is the data, y^k are the labels
$(x,y) \in D^k$	Data sample x and the corresponding label y
Φ_k, ϕ	Set of indices of remote clients with respect to C^k, Index of a remote client
$\mathcal{M}^k = [\mathcal{M}^k_{h_1}, \mathcal{M}^k_{h_2}]$	Multi-head model held by client k
$\mathbf{w}_k = [\mathbf{w}^k_1, \mathbf{w}^k_2]$	Weight parameters of the local model of client k
$\mathcal{L}_{k,CE}$	Fully supervised Loss computed on client k
$\mathcal{L}_{k,KD}$	Distillation loss used for client k
α	loss weight parameter

like storage, processing speed, and execution time. Molo et al. [19] proposed a knowledge distillation approach for vehicle detection using smart cameras in parking lots, where a large detector (teacher) guides smaller edge-based models (students) without additional labeled data. Their experimental results showed that students improve performance and can even surpass models trained with annotations.

Other approaches used a KD-based learning without a single teacher. Zhmoginov et al. [28] introduced Multi-Headed Distillation for distributed learning on the ImageNet dataset. This approach uses multiple model heads distilling to each other and simultaneous distillation of client model predictions and network embeddings, resulting in significantly higher accuracy than naive distillation methods. Jin et al. [12] introduced a personalized Federated Learning (FL) framework using self-KD to transfer historical personalized knowledge, balancing personalization and generalization. Similarly, Jeong et al. [11] addressed personalization challenges in FL for clients with diverse data and behaviors by proposing a KD-based algorithm to compare local models, enhancing client performance without data sharing and showing improved test accuracy, especially under non-iid data distributions.

Most works in the literature use KL divergence or CE as dissimilarity measures for distillation. However, there remains significant potential to investigate and utilize alternative dissimilarity measures, which could offer new insights into efficient knowledge transfer and performance across various learning tasks and scenarios.

3 Fully Decentralized Learning Model

In this section, we outline the decentralized learning environment used to evaluate the effectiveness of various information dissimilarity measures, introduced in Section 2.1, whose results are discussed in Section 5. The notation used is summarized in Table 1.

We consider a full network of K clients represented by a directed graph $(\mathcal{G},\varepsilon)$, where $\mathcal{G} = \{G^k \mid k \in K\}$ is a set of nodes and ε is the set of edges between the nodes.

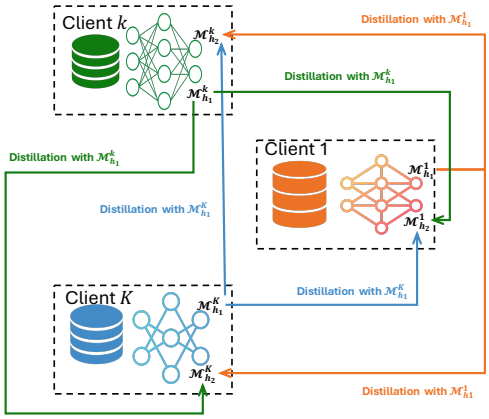


Fig. 2: KD-based decentralized network consisting of K clients, where distillation is performed using soft labels for effective knowledge transfer. In this setting, the first head of each client is communicated to the neighboring clients.

Each node G^k represents a client C^k holding a local dataset D^k composed of a pair (X^k, y^k), with $X^k = \{x_i^k\}_{i=1}^I$ representing the set of input data and $y^k = \{y_i^k\}_{i=1}^I$ the corresponding ground-truth labels. Each client C^k holds a model $\mathcal{M}^k$, which we assume to be a multi-head neural network. Specifically, the model has a backbone, which is the main body of the neural network that processes input data into a feature representation, and two heads, which take the features extracted by the backbone and perform final task-specific operations. The heads consist of a set of fully connected layers added on top of the backbone. We denote the models consisting of the backbone and the first head as $\mathcal{M}_{h_1}^k$, and the backbone and the second head as $\mathcal{M}_{h_2}^k$. The model with the first head, $\mathcal{M}_{h_1}^k$, is trained on the local distribution D^k, while the second, $\mathcal{M}_{h_2}^k$, is trained using knowledge distillation from connected clients.

The considered KD-based training procedure for this decentralized network involves training multiple clients concurrently, allowing them to share knowledge through distillation to improve overall model performance. Initially, each client's first model $\mathcal{M}_{h_1}^k$ is trained in a supervised manner until convergence with local data D^k. Then, as shown in Fig. 2, for each $k \in \{1, \ldots, K\}$, the first model from client C^k is shared with all outgoing connected clients in $\mathcal{G}$. Concurrently, client C^k receives the first head from all other incoming connected clients. This exchange enables each client to integrate knowledge from others while preserving their local data and model specialization, facilitating collaborative learning across the decentralized network. For the purposes of this study, we assume that all clients are interconnected. However, the proposed approach can be easily adapted to accommodate networks with different topologies and size.

For a fixed k, we used the notation Φ_k to indicate all the indices except k. We refer C^k as the current client and $\{C^\phi \,|\, \phi \in \Phi_k\}$ as the remote clients. So, once the first head of the models are trained, C^k communicates $\mathcal{M}_{h_1}^k$ to all remote clients and receives the models

$\{\mathcal{M}_{h_1}^{\phi}\}_{\phi\in\Phi_k}$ from them. The client C^k performs distillation using the available models from remote clients to train its second head $\mathcal{M}_{h_2}^k$. Specifically the parameters $\mathbf{w}_2^k$ of $\mathcal{M}_{h_2}^k$ are trained by optimizing a local total loss $\mathcal{L}_k$, which is obtained as a combination of a cross-entropy loss $\mathcal{L}_{k,CE}$ and a distillation loss $\mathcal{L}_{k,KD}$:

$$\mathcal{L}_k = \alpha\mathcal{L}_{k,CE} + (1-\alpha)\mathcal{L}_{k,KD}, \tag{8}$$

where $\alpha \in [0,1]$ is a parameter that weights the contribution of the losses with respect to the total loss. This dual-phase training approach allows each client to effectively train its local model while leveraging shared knowledge from other clients, improving generalization and performance across the network. The cross-entropy loss

$$\mathcal{L}_{k,CE} = \mathbb{E}_{(x,y)\sim D^k}\mathcal{L}_{CE}(\mathbf{w}_2^k, x, y) \tag{9}$$

is used to minimize local prediction with respect to the ground-truth labels of local data.[6]

For defining the distillation loss $\mathcal{L}_{k,KD}$ we considered two alternatives:

- **Case 1**: The *sum* of pairwise dissimilarities between the current client's soft-prediction and remote client's soft-predictions.
- **Case 2**: A distillation loss based on the dissimilarity between the current client's soft-predictions and the *average* of soft-predictions from remote clients.

Formally, let $\mathbf{p}^k(\mathbf{w}_2^k, x) = [p_1^k, p_2^k, \ldots, p_N^k]$ denote the softmax output obtained using the $\mathcal{M}_{h_2}^k$ model for the input data x, and $\mathbf{p}^{\phi}(x) = [p_1^{\phi}, p_2^{\phi}, \ldots, p_N^{\phi}]$ the softmax outputs of a remote client ϕ for the input data x (obtained using the pre-trained $\mathcal{M}_{h_1}^{\phi}$ model). For Case 1, we used

$$\mathcal{L}_{k,KD}(\mathbf{w}_2^k, x) = \sum_{\phi\in\Phi_k} \mathbb{E}_{x\sim X^k} f\left(\mathbf{p}^k(\mathbf{w}_2^k, x), \mathbf{p}^{\phi}(x)\right) \tag{10}$$

where f can be any divergence measure (e.g., CE,KL, TD, SED, JS). Since the sum of pairwise dissimilarities is used, we refer to this case as "sum" in the experiments. For Case 2, referred to as "average" in the experiments, we used

$$\mathcal{L}_{k,KD}(\mathbf{w}_2^k, x) = \mathbb{E}_{x\sim X^k} f\left(\mathbf{p}^k(\mathbf{w}_2^k, x), \frac{\sum_{\phi\in\Phi_k}\mathbf{p}^{\phi}(x)}{|\Phi|}\right) \tag{11}$$

Algorithm 1 summarizes the considered distillation training procedures.

4 Experimental Setup

Our analysis was conducted on a decentralized network consisting of three interconnected clients. This topology serves as a baseline evaluation, with plans for future work

[6] Please note that $\mathcal{L}_{CE}(\mathbf{w}_2^k, x, y)$ is simply the CE dissimilarity (Eq. (5)) between the output of $\mathcal{M}_{h_2}^k$ model for the input x and the true labels y

Algorithm 1: Decentralized Training with Knowledge Distillation

Data: $(\mathcal{G},\varepsilon)$ graph representing a network of K client $\{C_1,\dots C_K\}$
Local datasets $D^k=(X^k,y^k)$, where $X^k=\{x_i^k\}_{i=1}^I$ is set of input data and $y^k=\{y_i^k\}_{i=1}^I$ is the set of labels associated with each input, for all $k\in\{1,\dots,K\}$.
Result: Trained model parameters for each client.

```
// Initialization
```
foreach *client* $k\in\{1,\dots,K\}$ ***in parallel*** **do**
  `/* Train the model with the first head,` $\mathcal{M}_{h_1}^k$`, until convergence. `$\mathbf{w}_1^k$` are the model parameters to be updated */`
  $\mathcal{M}_{h_1}^k \leftarrow \text{LocalModelTraining}\left(\mathcal{L}_{CE}(\mathbf{w}_1^k, D^k)\right)$
  `// Inizialize the model with the second head,` $\mathcal{M}_{h_2}^k$.
  $backbone(\mathcal{M}_{h_2}^k) \leftarrow backbone(\mathcal{M}_{h_1}^k)$
  $head(\mathcal{M}_{h_2}^k)$ randomly inizialized
end
```
// Communication
```
for *each client* $k\in\{1,\dots,K\}$ ***in parallel*** **do**
  $\Phi_k \leftarrow$ indices of incoming connected clients in $\mathcal{G}$ `// remote client indices`
  Share $\mathcal{M}_{h_1}^k$ with all outgoing connected clients in $\mathcal{G}$
  Receive $\mathcal{M}_{h_1}^{\phi}$ from all remote clients $\phi\in\Phi_k$
end
```
// Knowledge Distillation
```
foreach *client* $k\in\{1,\dots,k\}$ ***in parallel*** **do**
  `/* Train the model with the second head` $\mathcal{M}_{h_2}^k$ `using KD until convergence. Use the loss` $\mathcal{L}_k=\alpha\mathcal{L}_{k,CE}+(1-\alpha)\mathcal{L}_{k,KD}$`, where` $\mathbf{w}_2^k$ `are the model parameters to be updated,` $\mathcal{L}_{k,KD}$ `is calculated either using Eq. Eq.10 or 11 */`
  $\mathcal{M}_{h_2}^k \leftarrow \text{LocalModelTraining}\left(\mathcal{L}_k(\mathbf{w}_2^k, D^k)\right)$
end

to extend the analysis to networks with more clients and various connectivity topologies.

We studied the effectiveness of different information dissimilarity measures (namely, CE, KL, SED, TD, JS) on distributed learning systems with different levels of data heterogeneity, ranging from scenarios where the data distribution is uniform across all clients (iid) to more extreme situations where each client focuses on its own specific tasks (non-iid). For this purpose, we used the CIFAR-10 [14] dataset and the SUN397 [29] dataset. We split the datasets into three subsets, corresponding to three clients in total.

For the CIFAR-10, the iid distribution is obtained by shuffling and evenly splitting the entire dataset, ensuring each client has different samples. For the non-iid distribution across the clients, we followed the configuration in [28]. Each client C^k receives a subset $\{\ell_i\}$ of the labels, which are designated as primary labels for C^k. Labels not included in $\{\ell_i\}$ are considered secondary for C^k. Samples for each label ℓ are distributed randomly among clients, with a higher probability ($1+\gamma$ times greater) of being assigned to

clients that have ℓ as a primary label. The parameter γ, referred to as dataset skewness, determines this distribution. When $\gamma = 0$, the data is distributed uniformly (iid), but as γ approaches infinity, samples for label ℓ are assigned exclusively to clients where ℓ is primary (non-iid). In the experiments, we used $\gamma = 15$ for CIFAR-10 and $\gamma = 10$ for SUN397.

Performance evaluation was conducted using 10% of the entire data distribution for both iid and non-idd datasets. For each client, we computed the accuracy of its model. In the next section, we present aggregated results, specifically the mean accuracy across the three clients.

In our implementation, we trained the three clients using independent Docker containers, each saving the model checkpoints to a shared folder. To train the second head of one client, the first heads from other remote clients are loaded from this shared folder for distillation. This choice was made to simplify the implementation and does not affect the analysis of the models' accuracy and the performance of the various losses. The study of training efficiency, including communication costs of model parameters, is left for future work.

All models are based on ResNet18 [9] and are initialized with weights pre-trained on ImageNet, as provided by PyTorch. We also employ standard data augmentation techniques as recommended in the PyTorch documentation[7] for ResNet18.

For the second head, we modified the classifier of ResNet18, using two dense hidden layers with 512 and 256 neurons for CIFAR-10 and 1024 and 512 neurons for SUN397, respectively. We set the skewness parameter to 15 for CIFAR-10, and 10 for SUN397. In all cases, the batch size is set to 128. The optimizer used is SGD with an initial learning rate of 0.001, momentum of 0.9, and a weight decay of 5×10^{-4}.

We performed the distillation using various temperature T values (1, 10, and 100)[8], depending on the dataset distribution. We report the optimal temperatures for each dissimilarity measure and dataset. For CIFAR-10 in the non-iid. context, $T = 10$ provided the best results for all dissimilarity measures for both the sum and average of remote predictions. In the iid context, $T = 10$ was optimal for the sum of distillation losses and only for CE, KL, and TD in the case of the average of remote predictions. For JS and SED, $T = 1$ is used. Moreover, for the SUN397 dataset, $T = 10$ was best for CE and KL, and $T = 1$ for all other cases, both for the sum and average of remote predictions.

The code to reproduce the experiments is available at https://github.com/joaquimbasa/Distributed_KD_Information_Dissimilarity.git.

5 Results and Discussion

Consistent with the correlation findings in [6], our experiments with an iid distribution of data among three clients revealed that the various dissimilarity measures tested in the KD-loss yielded comparable results. Fig. 3a and Fig. 3b present the average accuracy of secondary-head model $\mathcal{M}_{h_2}^k$ of clients belonging to $\mathcal{G}$ under iid data conditions on CIFAR-10 dataset, while varying the hyperparameter α. Here, $\alpha = 0$ indicates that

[7] https://pytorch.org/vision/main/models/generated/torchvision.models.resnet101.html

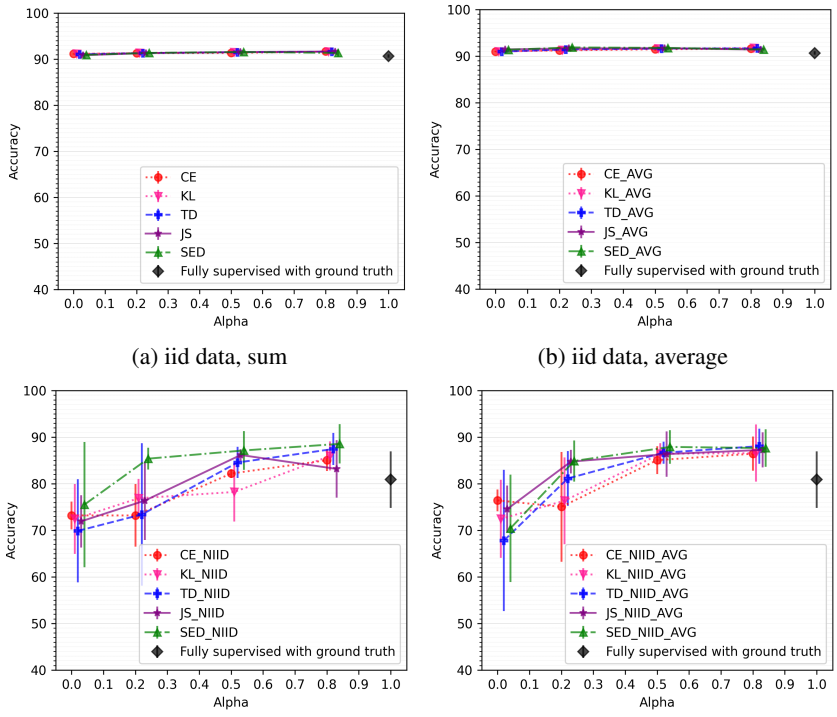
[8] As noted in [1], the best temperature is highly context-dependent, but a wide range of temperatures can be useful. They suggest using temperatures in the range of 0.1 up to 100

the total loss comprises only the distillation loss, while $\alpha = 1$ indicates that no distillation from remote clients is performed, and each model is trained solely in a supervised manner using its local annotated dataset. Overall, our results indicate that KD does not significantly enhance overall accuracy when the input data is sufficient and balanced. Furthermore, all tested dissimilarity measures exhibited performance similar to CE. This observation is consistent across both cases for computing the distillation loss: using the sum of pairwise distillation losses between the current client's predictions and those of each remote client (Eq. (10)), as shown in Fig. 3a, and using the distillation loss between the current client's prediction and the average of predictions from remote clients (Eq. (11)) as shown in Fig. 3b. Based on this observation, in iid settings, the choice of a dissimilarity measure may depend on implementation requirements, with a preference for computationally efficient measures such as TD. Fig. 3b also demonstrates that using distillation with the average predictions of remote clients C^{Φ} results in similar, and in some cases slightly better, performance than the sum of pairwise losses. This approach has the added advantage of allowing the computation of a single loss instead of multiple pairwise losses, thereby reducing computational complexity.

In the case of non-iid distribution (Fig. 3c and Fig. 3d), the distillation process led to an increase in the average accuracy of the clients' models compared to the fully-supervised approach. This improvement is particularly noticeable for the value $\alpha = 0.5$. For this value, all measures show minimal variance among the three clients (as indicated by the vertical bars) except in 3c, where the KL provides a high variance compared to others. For $\alpha > 0$ values, minimal differences are observed between JS and SED when computing the distillation loss with the average of predictions generated by the remote clients, whereas CE and KL perform worse in case $\alpha = 0.2$. Furthermore, the average of the predictions obtained from remote clients, in Fig. 3d shows that for $\alpha = 0.2$, SED and JS already exhibit good performance. However, for $\alpha = 0.8$, all measures perform similarly, with KL having higher variance across clients. On the other hand, SED appears to be superior to other measures from $\alpha = 0.2$, providing minimal variance when considering the sum of distillation losses.

In addition to CIFAR-10, we also performed experiments in the non-iid scenario using the SUN397 dataset. In these experiments, adding more layers to the second head caused the model to overfit, showcasing an average accuracy of 48.33% compared to the first head, showcasing an average accuracy of 57.74% over all clients. This confirms the argument made in [28] that when the client's training data is scarce, leading to model overfitting, communication between clients can enhance generalization and improve client's performance on their private tasks. Additionally, communication between clients improves their learned representations, making them better suited for adapting to tasks from other clients.

Regarding the performance of the different dissimilarity measures, Fig. 4a and Fig. 4b show that CE and KL are outperformed by SED, TD, and JS distances for $\alpha = 0$ and $\alpha = 0.8$ when using the sum of distillation losses from each remote client. However, when using the average of remote predictions, the CE and KL perform worse for the values of $\alpha = 0$ and $\alpha = 0.2$. In other cases, all measures perform equally well. Notably, all measures exhibit very low variance among the three clients.



(a) iid data, sum

(b) iid data, average

(c) non-iid data, sum

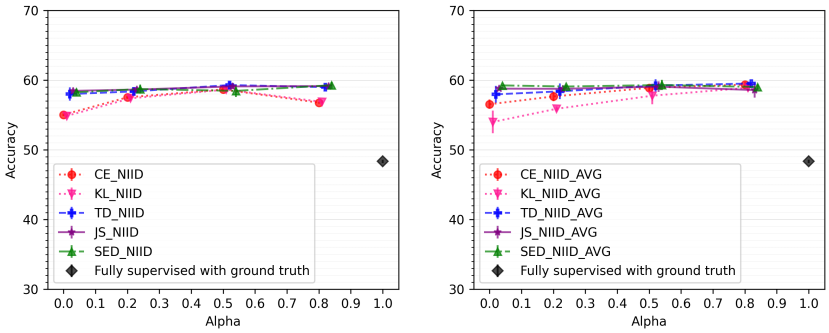
(d) non-iid data, average

Fig. 3: CIFAR-10: Mean accuracy over three clients considering the *sum* of the distillation losses (Eq 10) in the left-hand plots, and the the *average* of remote predictions to compute the distillation loss (Eq 11) in the right-hand plots. Results for iid data are shown in the top row, and results for non-iid data are shown in the bottom row. The bars indicate the standard deviation of accuracy across the three clients (not visible in the iid case, where the standard deviation is less than 0.8).

6 Conclusions

This paper empirically evaluated different information dissimilarity measures in a distributed KD setting. The core of our study was to understand the effectiveness of these measures using various data distributions. Furthermore, we used a multi-head neural network to facilitate knowledge transfer among clients, demonstrating that distance measures can significantly impact the training of distributed models using KD on non-iid data. Notably, the commonly used cross-entropy and Kullback-Leibler divergences are not always the most effective.

In future work, we plan to examine the stability of gradients (e.g, exploding or vanishing gradients) associated with the analyzed information dissimilarity measures, and evaluate the performance of the proposed distributed KD framework with a larger number of nodes and various graph topologies.



(a) sum (b) average

Fig. 4: SUN397 (non-iid data): Mean accuracy over three clients considering (a) the *sum* of the distillation losses (Eq 10); (b) the *average* of C^{Φ} prediction to compute the distillation loss (Eq 11). The standard deviation of accuracy across the three clients is less than 0.9 for all the plotted cases.

Acknowledgment

This work was partially funded by National Centre for HPC, Big Data and Quantum Computing project (EU NextGenerationEU PNRR, CUP B93C22000620006), and SUN – Social and hUman ceNtered XR (EC, Horizon Europe n. 101092612).

References

1. Agarwala, A., Pennington, J., Dauphin, Y., Schoenholz, S.: Temperature check: theory and practice for training models with softmax-cross-entropy losses. arXiv preprint arXiv:2010.07344 (2020)
2. Aguilar, G., Ling, Y., Zhang, Y., Yao, B., Fan, X., Guo, C.: Knowledge distillation from internal representations. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 7350–7357 (2020)
3. Bistritz, I., Mann, A., Bambos, N.: Distributed distillation for on-device learning. Advances in Neural Information Processing Systems **33**, 22593–22604 (2020)
4. Carta, A., Cossu, A., Lomonaco, V., Bacciu, D., van de Weijer, J.: Projected latent distillation for data-agnostic consolidation in distributed continual learning. Neurocomputing p. 127935 (2024)
5. Connor, R.: A tale of four metrics. In: 9th International Conference on Similarity Search and Applications, SISAP 2016. pp. 210–217. Springer (2016)
6. Connor, R., Dearle, A., Claydon, B., Vadicamo, L.: Correlations of cross-entropy loss in machine learning. Entropy **26**(6) (2024)
7. Gou, J., Xiong, X., Yu, B., Du, L., Zhan, Y., Tao, D.: Multi-target knowledge distillation via student self-reflection. International Journal of Computer Vision **131**(7), 1857–1874 (2023)
8. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**(6), 1789–1819 (2021)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (3 2015)
11. Jeong, E., Kountouris, M.: Personalized decentralized federated learning with knowledge distillation. In: ICC 2023-IEEE International Conference on Communications. pp. 1982–1987. IEEE (2023)
12. Jin, H., Bai, D., Yao, D., Dai, Y., Gu, L., Yu, C., Sun, L.: Personalized edge intelligence via federated self-knowledge distillation. IEEE Transactions on Parallel and Distributed Systems **34**(2), 567–580 (2023)
13. Kim, T., Oh, J., Kim, N., Cho, S., Yun, S.Y.: Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. arXiv preprint arXiv:2105.08919 (2021)
14. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research) (2009)
15. Liu, X., Yu, J., Liu, Y., Gao, Y., Mahmoodi, T., Lambotharan, S., Tsang, D.H.K.: Distributed intelligence in wireless networks. IEEE Open Journal of the Communications Society **4**, 1001–1039 (2023)
16. Luo, Y., Huang, Q., Ling, J., Lin, K., Zhou, T.: Local and global knowledge distillation with direction-enhanced contrastive learning for single-image deraining. Knowledge-Based Systems **268**, 110480 (2023)
17. Markatou, M., Chen, Y., Afendras, G., Lindsay, B.G.: Statistical distances and their role in robustness. New advances in statistics and data science pp. 3–26 (2017)
18. Mishra, R., Gupta, H.P.: Designing and training of lightweight neural networks on edge devices using early halting in knowledge distillation. IEEE Transactions on Mobile Computing (2023)
19. Molo, M.J., Carlini, E., Ciampi, L., Gennaro, C., Vadicamo, L.: Teacher-student models for ai vision at the edge: A car parking case study. Proceedings Copyright **508**, 515 (2024)
20. Moss, R., Connor, R.: A multi-way divergence metric for vector spaces. In: Similarity Search and Applications: 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings 6. pp. 169–174. Springer (2013)
21. Nielsen, F.: On a generalization of the jensen–shannon divergence and the jensen–shannon centroid. Entropy **22**(2), 221 (2020)
22. Parker, L.E.: Distributed intelligence: Overview of the field and its application in multi-robot systems. In: AAAI fall symposium: regarding the intelligence in distributed intelligent systems. pp. 1–6 (2007)
23. Sahni, Y., Cao, J., Zhang, S., Yang, L.: Edge mesh: A new paradigm to enable distributed intelligence in internet of things. IEEE access **5**, 16441–16458 (2017)
24. Topsoe, F.: Some inequalities for information divergence and related measures of discrimination. IEEE Transactions on Information Theory **46**(4), 1602–1609 (2000)
25. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1365–1374 (2019)
26. Wong, A.K., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE transactions on pattern analysis and machine intelligence (5), 599–609 (1985)
27. Yang, Z., Zeng, A., Li, Z., Zhang, T., Yuan, C., Li, Y.: From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17185–17194 (2023)
28. Zhmoginov, A., Sandler, M., Miller, N., Kristiansen, G., Vladymyrov, M.: Decentralized learning with multi-headed distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8053–8063 (2023)
29. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. Advances in neural information processing systems **27** (2014)